Master Thesis

Conformal Multistep-Ahead Multivariate Time-Series Forecasting

Filip Schlembach

Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science of Data Science for Decision Making
at the Department of Advanced Computing Sciences
of the Maastricht University

Thesis Committee:

Dr. E. Smirnov
Dr. I. Koprinska[1]
Dr. C. Seiler

Maastricht University
Faculty of Science and Engineering
Department of Advanced Computing Sciences

December 11, 2022

[1]School of Computer Science, The University of Sydney, Sydney

Contents

Abstract

Time-series forecasts underpin the decision-making process in a wide range of application domains. Energy demand, production, and price, stock prices, service demand, and medical prognoses are just a few of them. Recently, methods based on conformal prediction, a framework to improve the decision-making process by adding a prediction interval to point forecasts, have been proposed. They quantify the uncertainty of a predictive model, giving the user a range of scenarios to consider. However, these methods are limited in their application to time-series tasks, either because of the exchangeability condition they place on the data, or because they only allow for a single-step-ahead univariate setting. In this thesis, I combine two existing approaches, one built for multi-target regression and one designed to handle non-exchangeable data. The resulting method is computationally efficient, easy to implement and produces valid prediction regions for multistep-ahead multivariate time-series forecasts. A theoretical analysis proves the method's validity while experiments on real-world data sets give insights into its practical behavior.

Figure 1: I have attempted science [14].

Chapter 1

Introduction

1.1 Problem Statement

Time-series forecasting is an important part of the decision-making process in a wide range of application domains. Rolnick et al. [15] have, for instance, identified time-series analysis and uncertainty quantification as areas of machine learning relevant to enabling low-carbon electricity. Other application domains include stock price predictions, service demand forecasting, and medical prognoses [18]. While modern time-series forecasting techniques have become increasingly accurate [7, 8, 9] they usually produce point forecasts, providing little to no information about the model's uncertainty. Uncertainty quantification is however critical to improve decision-making, especially in high-stakes situations.

Recently, *conformal prediction* has been proposed for this task. Conformal prediction is a modern, model agnostic technique that provides uncertainty quantification in the form of valid prediction intervals [2] without making assumptions about the distribution of the data. Conformal prediction does however rely on two assumptions that are generally not met in time-series forecasting tasks. It assumes, that the examples in the data set are exchangeable and that the model fitting algorithm for the underlying model treats the examples in the data set symmetrically[1]. These requirements render the technique unsuitable for many time-series applications, among them energy production and consumption forecasting [25].

Multiple attempts have been made to lift these requirements. While some techniques derived from conformal prediction allow for non-exchangeable data and model-fitting algorithms that do not treat the examples in the data set symmetrically, they are limited to univariate, single-step-ahead time-series forecasting tasks.

1.2 Research Questions

Motivated by Rolnick et al. [15], I set out to research if and how conformal prediction can be extended to the more complex multistep-ahead multivariate

[1] These terms are defined in Chapter 2

time-series forecasting tasks. The work of Stankevičiūtė et al. [18] provides an approach that allows for the construction of valid prediction regions for multi-target regression by combining multiple conformal predictors. However, its application to time-series is limited by the exchangeability requirement. Barber et al. [2] on the other hand, propose a method that is specifically designed to overcome the requirements of the original conformal predictor and render it applicable to time-series forecasting tasks. While powerful, their method is bound to single-target regression. This leads me to the following research questions.

- Can the assumptions made by current multivariate time-series prediction interval estimation methods be relaxed?
- Can the coverage of current multivariate time-series prediction interval estimation methods be improved?

1.3 Contributions

To answer the research questions presented in Chapter 1.2, I develop a new method, *nmtCP*, that is capable of producing valid prediction regions for multistep-ahead multivariate time-series forecasting tasks. Chapter 3.2 describes the general concept of nmtCP, while Chapter 3.3 shows how it is implemented in practice. I offer a theoretical analysis of the validity of the method in Chapter 3.4 and compare it to existing methods in Chapter 3.5.

The development of nmtCP is preceded by a literature review that spans form the original conformal prediction method in Chapter 2.1 to the most recent adaptations to the time-series forecasting domain in Chapter 2.2, describing their respective contributions. Chapter 2.3 discusses and compares these methods.

After the theoretical analysis of nmtCP, I conduct a series of experiments to validate the theoretical properties on two real world data sets. This also serves to compare nmtCP to another state-of-the-art method. After a description of the data sets and the experimental setup, Chapter 4 displays the results of these experiments and contains a discussion of the method's behavior. A subset of these results has already been published in Schlembach et al. [17]. To conduct the practical experiments, I implemented the method in a modular pipeline that can easily accommodate different underlying models.

Chapter 2

Related Work

In this chapter, I first introduce the original conformal prediction method by Vovk et al. [24] and its subsequent adaptations for an efficient application in a regression setting in Chapter 2.1. This is followed by a description of the extensions of the original method for time-series forecasting in Chapter 2.2. Chapter 2.3 further analyses and discusses these extensions.

2.1 Conformal Prediction

The upper bounds on the probability of error of machine learning algorithms' predictions provided by statistical learning theory are generally too loose to be useful in practice [24, p.4]. This is why Vovk et al. [24] set out to quantify the confidence of machine learning algorithms' outputs derived from statistical learning theory. Their idea is to use a model's performance on previous samples as well as information about the current sample to estimate the confidence with which the label of a new object is predicted. Therefore, in the context of *conformal prediction*, the confidence for a new sample is based on its (dis)similarity to known samples [24, p.8], assuming that the underlying model will perform similarly on conforming examples.

A benefit of this method is that it can be used on top of almost any machine-learning algorithm while retaining its theoretical guarantees [24, p.11] under minimal assumptions [19], usually only requiring that the examples in the training set are exchangeable and that the underlying model treats them symmetrically [2]. Conformal prediction also requires minimal to no modifications of the underlying model [18].

2.1.1 Conformal Predictors

Formally, real world examples are drawn independently of each other from a distribution Q over a fixed space $\mathcal{Z}$. Each example $z_i \in \mathcal{Z}$ consists of an object $x_i \in \mathcal{X}$ and a label $y_i \in \mathcal{Y}$. $\mathcal{X}$ and $\mathcal{Y}$ are measurable spaces and represent the object space and the label space respectively. The example space is defined as the Cartesian product $\mathcal{Z} \coloneqq \mathcal{X} \times \mathcal{Y}$ [24, p.18]. No assumptions are made about the distribution Q.

Let

$$h : \mathcal{Z}^* \times \mathcal{X} \to \mathcal{Y} \tag{2.1}$$

be a *simple predictor* that, given a number n of samples $z_{1:n} \in \mathcal{Z}$ and an object $x_{n+1} \in \mathcal{X}$, predicts the associated label $\hat{y}_{n+1} \in \mathcal{Y}$ [24, p.18]. Here, $z_{1:n}$ stands for $z_1, \ldots, z_n$.

In contrast, let

$$\Gamma : \mathcal{Z}^* \times \mathcal{X} \times (0,1) \to 2^{\mathcal{Y}} \tag{2.2}$$

be a *confidence predictor* that, compared to the simple predictor, takes an additional input, the *significance level* $\alpha \in (0,1)$ [24, p.19]. It outputs $\hat{\mathbf{y}}_{n+1}$, a subset of $\mathcal{Y}$ associated with x_{n+1}.

The confidence predictor is said to be *exactly valid* or *exact* if the probability of an error, i.e. that $y_{n+1} \notin \hat{\mathbf{y}}_{n+1}$, is α and if errors for different objects are independent [24, p.20]. If $\mathsf{P}(y_{n+1} \notin \hat{\mathbf{y}}_{n+1}) \leq \alpha$ it is *conservatively valid* [24, p.20]. Vovk et al. [24, p.21] show that no confidence predictor achieves exact validity.

To go from confidence predictors to conformal predictors, I need to introduce nonconformity measures. They indicate how different a new example is from a bag of old examples. Any measurable function

$$A : \mathcal{Z}^{(*)} \times \mathcal{Z} \to \bar{\mathbb{R}} \tag{2.3}$$

is called a *nonconformity measure* (NCM). In this context, a bag is an unordered set that allows for repetition and $\lbag z_{1:n+1} \rbag$ denotes a bag of $n+1$ examples. $\mathcal{Z}^{(*)}$ is the set of all bags of elements of $\mathcal{Z}$ [24, p.23]. For a given NCM A as defined in Equation (2.3),

$$A_{n+1} : \mathcal{Z}^{(n)} \times \mathcal{Z} \to \bar{\mathbb{R}} \tag{2.4}$$

is also called nonconformity measure for bags of size $n+1$ [24, p.25]. Such a NCM A_{n+1} allows for the computation of the nonconformity scores

$$r_i \coloneqq A_{n+1}(\lbag z_{1:i-1,i+1:n+1} \rbag, z_i), \qquad \forall i = 1, \ldots, n+1 \tag{2.5}$$

for every example in a given a bag of $n+1$ examples [24, p.25]. The associated p-values

$$p_{z_i} \coloneqq \frac{|\{j = 1, \ldots, n+1 : r_j \geq r_i\}|}{n+1} \tag{2.6}$$

indicate what fraction of examples from the bag are at least as nonconforming as the ith one [24, p.25].

Assuming the label y_{n+1} to the $n+1$st element in the data set is unknown, $z_{n+1} = (x_{n+1}, \bar{y}_{n+1})$ is the *completion* for the $n+1$st element and the hypothetical label $\bar{y}_{n+1}$ [19]. The *conformal predictor* (CP) *determined by a nonconformity measure* (A_{n+1}) is

$$\Gamma^{\alpha}(\lbag z_{1:n} \rbag, x_{n+1}) \coloneqq \{y \mid p_{z_{n+1}} > \alpha\} \tag{2.7}$$

meaning, that for a given bag of n examples and an object x_{n+1} it predicts the set of all hypothetical labels $\bar{y}_{n+1}$, such that the completions z_{n+1} are less or as nonconforming as the fraction α of all $n+1$ examples [19].

CPs constructed in this way have conservative asymptotic validity guarantee, meaning that the frequency of errors converges to α [19]. Given the error indicator function

$$\mathrm{err}_i := \begin{cases} 1 & \text{if } y_i \notin \Gamma^\alpha(\lbag z_{1:i-1} \rbag, x_i) \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

this property can be formalized as

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} \mathrm{err}_i}{n} \leq \alpha \tag{2.9}$$

[19]. When the definition of the p-value in Equation (2.6) changes to a smoothed p-value

$$p_{z_i} := \frac{|\{j = 1, \ldots, n+1 : r_j \geq r_i\}| + \tau|\{j = 1, \ldots, n+1 : r_j = r_i\}|}{n+1} \tag{2.10}$$

CPs achieve exact asymptotic validity and are called *smoothed conformal predictors* [24, p.27]. Here, $\tau \sim U[0,1]$ is a random variable that acts as a tiebreaker for multiple elements with equal nonconformity score r_i [19].

In addition to these asymptotic guarantees, Vovk et al. [24, p.27] have proven

$$\forall n > 0,\ \forall \delta > 0 \qquad \mathsf{P}\left(\frac{\sum_{i=1}^{n} \mathrm{err}_i}{n} \geq \alpha + \delta\right) \leq e^{-2n\delta^2} \tag{2.11}$$

for a finite number of samples [19].

Note that for both versions of the conformal predictor, the validity guarantees hold independent of the chosen conformity measure A_{n+1} [19]. If the framework of conformal predictors is used to add a prediction region to a simple predictor as defined in Equation (2.1), a common class of NCMs is

$$r_i = \Delta(y_i, h_{\lbag z_{1:i-1, i+1:n+1} \rbag}(x_i)), \qquad \forall i = 1, \ldots, n+1 \tag{2.12}$$

measuring the discrepancy between the value predicted by the simple predictor for a given object x_i and the associated label y_i.

Next to validity, *(predictive) efficiency* is a desired property for CPs, meaning that the generated prediction sets should be non-empty and as small as possible [24, p.9].

2.1.2 Computational Efficiency

Vapnik [20, p.30] introduces the distinction between induction and transduction based on the principle that "when solving a given problem, try to avoid solving a more general problem as an intermediate step". In the context of inference, induction refers to the estimation of a function given a data set, and deduction to the use of the estimated function to derive a prediction, whereas transduction designates estimating the value the function would have given directly, based on the available data [20, p.293] [21, p.460]. This interpretation equates the estimation of a function, i.e. training a model, with solving a more general problem.

Vovk et al. [24, p.98] apply this distinctions to the underlying predictor appearing in the class of NCMs shown in Equation (2.12). An underlying model

h is said to be transductive if no preprocessing, i.e. training, can be done to speed up estimation of a label given a new sample.[1] Inductive models such as neural networks, on the other hand require a training or fitting step. A CP as defined in Equation (2.7) determined by a NCM that relies on an inductive underlying model is computationally expensive. This is because, as seen in Equation (2.12), the underlying model needs to be fitted for every example in the bag $\wr z_{1:n+1} \int$ for every new example z_{n+1} that is presented to the CP. *Inductive conformal prediction* overcomes the computational complexity of the transductive variant by estimating a generic function that does not depend on the current example [24, p.98]. This is achieved by splitting the known examples into two disjoint sets, a proper training set $D_{train} = \wr z_{1:m_0} \int$ of size m_0 and a calibration set $D_{cal} = \wr z_{m_0+1:n} \int$ of size $m_c = n - m_0$ [19][2]. The *inductive nonconformity measure* based on Equation (2.12)

$$r_i = \Delta(y_i, \hat{\mu}(x_i)), \qquad \forall i = m_0 + 1, \ldots, n + 1 \tag{2.13}$$

where $\hat{\mu} = h_{\wr z_{1:m_0} \int}$ fits the underlying model only once to the proper training set D_{train}. To retain the asymptotic validity guarantee, the definition of the p-value and smoothed p-value is changed to

$$p_{z_i} \coloneqq \frac{|\{j = m_0 + 1, \ldots, n + 1 : r_j \geq r_i\}|}{m_c + 1} \tag{2.14}$$

[24, p.98] and to

$$p_{z_i} \coloneqq \frac{|\{j = m_0 + 1, \ldots, n + 1 : r_j \geq r_i\}| + \tau|\{j = m_0 + 1, \ldots, n + 1 : r_j = r_i\}|}{m_c + 1} \tag{2.15}$$

respectively [24, p.99]. These changes result in the *inductive conformal predictor* (ICP) [24, p.98], more recently referred to as *split conformal predictor* [19]. The reduction of computational complexity ICPs offer compared to CPs comes at the cost of splitting the labeled examples, therefore decreasing the size of the (proper) training set for the underlying model as well as the number of examples for which a p-value is computed, which may reduce the accuracy of the method [2].

Equation (2.7) indicates that a nonconformity score needs to be computed for all possible labels of the current object x_{n+1} to verify whether they should be included in the prediction region $\hat{\mathbf{y}}_{n+1}$. This might be possible in the context of classification, where the number of labels is small [22]. It is however unfeasible for regression models, as the labels can generally take on an infinite number of values[3] [19]. In the context of ICPs, the computational efficiency can be increased by using nonconformity measures, for which the labels associated with a given p-value can be computed in a finite number of steps [19]. Residuals

$$r_i = \begin{cases} |y_i - \hat{\mu}(x_i)| & \forall i = m_0 + 1, \ldots, n \\ |\bar{y}_{n+1} - \hat{\mu}(x_{n+1})| & i = n + 1 \end{cases} \tag{2.16}$$

[1] Nearest neighbor classifiers are an example of transductive models [24, p.6].

[2] Due to the exchangeability of the examples z_i, this split is done randomly, even if the index might suggest otherwise.

[3] Exceptions to this rule are presented by [24, p.29-33] for Ridge regression and [2] for linear regression. They isolate the influence the hypothetical label $\bar{y}$ has on the p-value for these particular regression models and provide an explicit way to calculate the values of $\bar{y}$, for which the p-value changes.

are an example of such an NCM. Once the residuals have been computed for the validation set, an ICP determined by the NCM (2.16) produces the prediction region

$$\begin{aligned}\hat{\mathbf{y}}_{n+1} &= \Gamma^{\alpha}(\wr z_{1:n} \int, x_{n+1}) \\ &= \hat{\mu}(x_{n+1}) \pm \mathsf{Q}_{1-\alpha}\left(\sum_{i=m_0+1}^{n} \tfrac{1}{m_c+1} \cdot \delta_{r_i} + \tfrac{1}{m_c+1} \cdot \delta_{r_{+\infty}}\right)\end{aligned} \tag{2.17}$$

[2] where $\mathsf{Q}_{1-\alpha}$ denotes the quantile function, returning the $\lceil (1-\alpha)(m_c+1) \rceil$th smallest value of the residuals $r_{m_0+1:n}$ [18]. The Kronecker delta function δ_x represents the point mass at x [2].

2.1.3 On-Line, Semi-On-Line and Off-Line Setting

ICPs can be used in an on-line, semi-on-line, semi-off-line and off-line setting. In this section, I summarize how these situations are handled in practice.

In an *on-line setting* nature provides objects x_i sequentially and their corresponding labels y_i delayed by one step [24, p.5]. This means that at the $n_0 + 1$st step, a CP as defined in Equation (2.7) can make a prediction $\hat{\mathbf{y}}_{n_0+1} = \Gamma^{\alpha}(\wr z_{1:n_0} \int, x_{n_0+1})$ [24, p.5]. Then, the true label y_{n_0+1} and a new object x_{n_0+2} are revealed and the CP can make the prediction $\hat{\mathbf{y}}_{n_0+2}$, and so on [24, p.6]. As a consequence the set of known examples grows at every time step. If if an ICP according to Equation (2.17) is used, the underlying model is re-trained at every time step for a new split of the updated set of known examples as shown in Algorithm 1.

Although the method presented in Algorithm 1 benefits from the second improvement described in Section 2.1.2 by using the residuals as NCM, it still requires the underlying model to be fitted for every new object that is encountered. To fully utilize the computational improvements provided by ICPs, Vovk et al. [24, p.98] introduce the sequence $m_1 < m_2 < \ldots$ with $m_i \in \mathbb{N}$. These m_i act as thresholds that determine after how many new observations the underlying model is fitted to an updated training set. Once the number of known examples n surpasses a previously unsurpassed m_k, such that $m_k < n \leq m_{k+1}$, the underlying model is fitted to the $z_{1:m_k}$. The $z_{m_k+1:n}$ are used as calibration set to compute the residuals [24, p.99]. Moving forward I will call this situation the *semi-on-line setting*.

Finally, in the *off-line setting*, nature does not provide the objects and labels sequentially. Instead, it provides a bag of known examples $\wr z_{1:n} \int$ and a set of objects $x_{n+1:n+m_u}$ with unknown labels. In this situation the procedure described in Section 2.1.2 can be applied directly. After the underlying model is fitted to the proper training set and the ICP is calibrated on the test set, the ICP generates prediction regions for all $x_{n+1:n+m_u}$. This setup weakens the validity guarantee. While $\mathsf{P}(y_i \notin \hat{\mathbf{y}}_i) \leq \alpha, \forall i \in n+1, \ldots, n+m_u$ still holds, the errors are no longer independent and $\sum_{i=1}^{n+m_u} \mathrm{err}_i / m_u \leq \alpha$ may no longer be true [24, p.112].

Note that I make the distinction between the on-line setting, the semi-on-line setting and the off-line setting based on the on the number of times the underlying model is fitted to a set of examples and the points in time when that happens. This classification is different from the one used by Toccaceli [19] who

Input: $\lbag z_{1:n_0} \rbag \in \mathcal{Z}^{n_0}$ `// bag of known examples`
$\alpha \in [0,1]$ `// significance level`
$c \in [0,1]$ `// relative calibration set size`
Output: $\hat{\mathbf{y}}_{n_0+1:*} \subset \mathcal{Y}^*$

$n \leftarrow n_0$
while *true* **do**
  $x_{n+1}, y_n \leftarrow$ observe reality
  $\lbag z_{1:n} \rbag \leftarrow \lbag z_{1:n-1} \rbag \cup \lbag (x_n, y_n) \rbag$
  $m_c = \lceil c \cdot (n) \rceil$ `//` $|D_{cal}|$
  $m_0 = n - m_c$ `//` $|D_{train}|$
  $D_{train}, D_{cal} \leftarrow$ split $\lbag z_{1:n} \rbag$ into bags of size m_0 and m_c
  $\hat{\mu} \leftarrow h_{\lbag D_{train} \rbag}$ `// fit the model`
  for $j \in 1, \ldots, m_c$ **do**
    $x_j, y_j \leftarrow D_{cal,j}$
    $r_j \leftarrow |y_j - \hat{\mu}(x_j)|$
  end
  $\hat{y}_{n+1} \leftarrow \hat{\mu}(x_{n+1})$
  $\hat{\mathbf{y}}_{n+1} \leftarrow \hat{y}_{n+1} \pm \mathsf{Q}_{1-\alpha}\left(\sum_{j=1}^{m_c} \frac{1}{m_c+1} \cdot \delta_{r_j} + \frac{1}{m_c+1} \cdot \delta_{r_{+\infty}}\right)$
  $n \leftarrow n + 1$
end

Algorithm 1: On-line inductive conformal prediction. For this method to have theoretical guarantees, the examples z_i need to be exchangeable and the training algorithm $h_{\lbag \ldots \rbag}$ needs to treat them symmetrically [2].

uses *batch mode of operation* to denote what I call the off-line setting. Vovk et al. [24, 112] call the semi-on-line setting with only a single m_1 *semi-off-line ICP*. Even though the underlying model is only trained once, this differs from the off-line setting in that nature reveals new objects sequentially and their true labels with a delay. Once a label becomes known, it, together with its associated object, is added to the calibration set. This procedure ensures that the method retains the general validity guarantees [24, p.111]. In [23] the use of inductive conformal predictors is equated to the semi-on-line setting.

To summarize, I describe the different settings using the following terms:

- on-line setting: the underlying model is fitted for every new example that nature provides;
- semi-on-line: the underlying the model is fitted periodically, after m_1, m_2, ... examples and, for ICP, the calibration set is updated with every new observed label;
- semi-off-line: underlying model is only trained once and, for ICP, the calibration set is updated with every new observed label;
- off-line: the underlying model is only trained once and, for ICP, the calibration set is not updated.

2.1.4 ICP Variants

Vovk [23] attempts to overcome the reduction in accuracy that ICPs experi-

ence by merging them with K-fold cross validation. Instead of splitting the known examples in two disjoint sets, he splits them into K folds of equal size. The underlying model is fitted K times, leaving a different fold as calibration set at every iteration. This ensures that nonconformity measures and p-values are available for all known examples. He calls the resulting predictors *cross-conformal predictors* (CCPs). While the experimental results Vovk [23] presents for classification tasks appear promising, he provides no theoretical guarantees for CCPs. Linusson et al. [10] show that the validity of CCPs depends on the nonconformity measure.

Leave-one-out conformal predictors (LOOCPs) are a special case of CCPs that Vovk [23] examines. As the name suggests, they use a number of folds K equal to the number of known examples n, each of size one. The *leave-one-out* (LOO) nonconformity scores are used in combination with an underlying model trained on the entire data set to generate the prediction regions. Barber et al. [1] use the term *jackknife* to refer to leave-one-out conformal cross validation. Let $\hat{\mu}_{-i} = h_{\lfloor z_{1:i-1,i+1:n} \rfloor}$ be the underlying model fitted to all but the ith example and $r_i = |y_i - \hat{\mu}_{-i}(x_i)|$ the associated nonconformity score using the residuals as NCM. In this case, LOOCPs produce prediction intervals

$$\hat{\mathbf{y}}_{n+1} = \hat{\mu}(x_{n+1}) \pm \mathsf{Q}^{+}_{n,1-\alpha}(r_{1:n}) \tag{2.18}$$

where $\hat{\mu} = h_{\lfloor z_{1:n} \rfloor}$ and $\mathsf{Q}^{+}_{n,1-\alpha}$ returns the $\lceil (1-\alpha)(n+1) \rceil$ smallest value of the provided arguments. In a similar manner, $\mathsf{Q}^{-}_{n,\alpha}$ returns the $\lfloor \alpha(n+1) \rfloor$ smallest value of the provided arguments. Barber et al. [1] extend LOOCPs and call their new method *jackknife+*. Instead of building the prediction intervals around $\hat{\mu}(x_{n+1})$ using the quantiles of the LOO residuals, jackknife+ produces prediction intervals

$$\hat{\mathbf{y}}^{\text{jackknife+}}_{n+1} = [\mathsf{Q}^{-}_{n,\alpha}(\{\hat{\mu}_{-i}(x_{n+1}) - r_i\}^{n}_{i=1}), \mathsf{Q}^{+}_{n,1-\alpha}(\{\hat{\mu}_{-i}(x_{n+1}) + r_i\}^{n}_{i=1})] \tag{2.19}$$

using the LOO predictions for the new example $\hat{\mu}_{-i}(x_{n+1})$. Barber et al. [1] show that that

$$\mathsf{P}(y_{n+1} \notin \hat{\mathbf{y}}^{\text{jackknife+}}_{n+1}) \leq 2\alpha \tag{2.20}$$

giving the jackknife+ method a theoretic coverage guarantee, something that is not available for LOOCPs. Despite the theoretical upper bound of the error rate being 2α, their experiments show an empirical error rate that is closer to the targeted error rate α. Barber et al. [1] also provide a similar extension to CCPs they call *CV+*. In addition, they present a rigorous theoretical and practical comparison of CPs, ICPs, the Jackknife method, the Jackknife+ method, the CV+ method and the K-fold cross conformal method.

2.2 Conformal Time-Series Forecasting

The definition of inductive conformal predictors in an on-line setting provided in Equation (2.17) and the subsequent variations can directly be applied to time-series. However, the validity of the predicted regions is no longer guaranteed as the observations in real world time-series are generally not exchangeable. In this section, I present the work of some researchers that have sought to overcome this limitation.

Before doing that, let me introduce the notation that I will use going forward. $o_{t,j} \in \mathbb{R}$ denotes the value of feature j at time step t. $o_t = o_{t,1:F} \in \mathbb{R}^F$ represents the vector of observations for all F features at time step t and $o_{1:T} \in \mathbb{R}^{T \times F}$ stands for the entire multivariate time-series with T time steps and F features. In the case where only a single feature is observed, the second index is omitted and $o_{1:T} \in \mathbb{R}^T$ represents a univariate time-series with T time steps.
A *single-step ahead time-series forecast* predicts $y = o_{T+1}$ given a univariate time-series $x = o_{1:T}$. If $x = o_{1:T}$ is a multivariate time-series, the single-step ahead forecast predicts some or all features of $y = o_{T+1,1:F}$. In contrast, a *multi-step ahead time-series forecast* (called *multi-horizon time-series forecast* by some authors [18]) predicts the values $y = o_{T+1:T+T_h}$ for multiple future time steps in the univariate setting. In the multivariate setting, it predicts the value for some or all features of these future time steps $y = o_{T+1:T+T_h}$. The number of predicted future time steps T_h is called the *prediction horizon*.

Chernozhukov et al. [3] assume a series of ordered examples $z_{1:m_0}$ with $z_i \in \mathbb{R}^{F-1} \times \mathbb{R}$. This is followed by a series of ordered objects $x_{m_0+1:m_0+m_1}$ with $x_i \in \mathbb{R}^{F-1}$ for which they want to generate a prediction region. For a set of hypothetical labels $\bar{y}_{m_0+1:m_0+m_1}$, they split the data into blocks of consecutive examples. Next they generate permutations $\pi_j \in \Pi$ of the original sequence by updating the index according to $\pi_j(i) \to (i+(j-1)b) mod (m_0+m_1)$, where b is the number of examples per block. Every permutation shifts the original data by j blocks. They define the p-value associated with the set of hypothetical labels as

$$p(\bar{y}_{m_0+1:m_0+m_1}) = \frac{\sum_{j=1}^{|\Pi|} |\{\pi_j : A(z_{\pi_j(1):\pi_j(m_1)}) \geq A(z_{1:m_1})\}|}{|\Pi|} \tag{2.21}$$

where A is the chosen NCM. Finally, their method returns all sets of hypothetical labels such that $\{\bar{y}_{m_0+1:m_0+m_1} : p(\bar{y}_{m_0+1:m_0+m_1}) > \alpha\}$. While the authors did not name their method, I will refer to it by *Conformal Inference by Permutations* (CIbP). Chernozhukov et al. [3] prove that their method remains approximately valid under weak conditions on the nonconformity score, relaxing the exchangeability condition of the data. However, the method is impractical because it requires an infinite number of hypothetical labels $\bar{y}$ to be tested in general, and cannot directly benefit from the improvements in computational efficiency presented in Section 2.1.2.[4] For exact validity, their method requires the examples to be exchangeable under the permutations π, a condition that is still difficult to meet for time-series with long term dependencies surpassing the block size.

Instead of constraining the observations, Xu and Xie [25] make assumptions about the time-series' stochastic errors and the estimation quality of the underlying models to build distribution-free prediction intervals with approximate coverage guarantee. In practice, their method resembles LOOCP, but they reduce the number of underlying models that need to be fitted. First, they create B bags D_b of size n by sampling with replacement from the known examples $z_{1:n}$. Next, they fit B underlying models $\hat{\mu}_b = h_{D_b}, \forall b \in B$. The residuals

$$r_i = |y_i - \phi(\{\hat{\mu}_b(x_i) \ \forall b : z_i \notin D_b\})| \tag{2.22}$$

[4]The reason for this is that these improvements only work for single target regression. For a more detailed analysis see Section 3.1.

are the absolute difference between the real label y_i and the aggregated predictions of all underlying models that were not trained on z_i. ϕ is the chosen aggregation function. Finally, they produce the prediction interval

$$\hat{\mathbf{y}}_{n+1} = \mathsf{Q}_{1-\alpha}(\{\hat{\mu}_b(x_{n+1}) \; \forall b : z_i \notin D_b\}_{i=1}^{n}) \pm \mathsf{Q}_{1-\alpha}(r_{1:n}) \tag{2.23}$$

for a new object x_{n+1}. In addition, they implement a mechanism to add the residual r_{n+1} when the label for a new object becomes available and to remove the oldest residual. Xu and Xie [25] prove that EnbPI produces valid prediction intervals under two assumptions. The first assumption is that the $\{r_{1:n}\}$ are stationary and strongly mixing. The second assumption states that the average difference between the residuals and the stochastic part of the data generating process decreases over time. Xu and Xie [25] test their EnbPI method on single-step ahead and multi-step ahead solar power predictions tasks. They compare it to the ARIMA method, which does not retain valid coverage in the experiments.

Gibbs and Candès [5] construct their *adaptive conformal inference* (ACI) method explicitly with a data distribution that shifts over time in mind, moving away from the exchangeability assumption. They do this by adapting the significance level α used for the quantile function in ICPs over time. Although Gibbs and Candès [5] describe ACI in more general terms, I only show the situation where an underlying model is used to produce point forecasts in the semi-off-line setting to keep the notation consistent. In this situation initially n examples $z_{1:n}$ are known. I split them into two disjoint sets, a proper training set $D_{train} = \langle z_{1:m_0} \rangle$ of size m_0 and a calibration set $D_{cal} = \langle z_{m_0+1:n} \rangle$ of size $m_c = n - m_0$. After fitting the underlying model $\hat{\mu} = h_{D_{train}}$ to the training set, I use it to compute the nonconformity scores $r_{m_0+1:n}$ on the calibration set. For future objects x_{n+i} that nature provides sequentially, ACI computes the prediction interval

$$\hat{\mathbf{y}}_{n+i} = \hat{\mu}(x_{n+i}) \pm \mathsf{Q}_{1-\alpha_{n+i}} \left(\sum_{j=m_0+1}^{n+i-1} \tfrac{1}{m_c+i} \cdot \delta_{r_j} + \tfrac{1}{m_c+i} \cdot \delta_{r_{+\infty}} \right) \tag{2.24}$$

with

$$\alpha_{n+i} = \alpha_{n+i-1} + \gamma(\alpha - \mathrm{err}_{n+i-1}) \tag{2.25}$$

starting at $\alpha_{n+1} = \alpha$. An alternative version of the method

$$\hat{\mathbf{y}}_{n+i} = \hat{\mu}(x_{n+i}) \pm \mathsf{Q}_{1-\alpha_{n+i}} \left(\sum_{j=m_0+1}^{n+i-1} \tfrac{\tilde{w}_j}{m_c+i} \cdot \delta_{r_j} + \tfrac{\tilde{w}_{n+i}}{m_c+i} \cdot \delta_{r_{+\infty}} \right) \tag{2.26}$$

Gibbs and Candès [5] present under the same name introduces a series of increasing weights $\tilde{w}_j$ with $\sum \tilde{w}_j = 1$. It uses a weighted quantile function, giving more recent nonconformity scores more weight compared to older ones. In both cases, the *step size parameter* γ determines how quickly the method adapts to shifts in distribution. Gibbs and Candès [5] define the

$$\text{coverage gap} = (1 - \alpha) - \mathsf{P}\{y_{n+i} \in \hat{\mathbf{y}}_{n+i}\} \tag{2.27}$$

as the loss in coverage compared to what would be achieved if the the examples were exchangeable. They show that by correctly calibrating α_{n+i}, the coverage

gap can be made arbitrarily small. In addition, they prove that using the on-line update scheme described in Equation (2.25), ACI is guaranteed to be asymptotically valid according to Equation (2.9). This stems from the observation that $\forall m_1 \in \mathbb{N}$

$$\left| \frac{1}{m_1} \sum_{i=n+1}^{n+m_1} \text{err}_i - \alpha \right| \leq \frac{\max\{\alpha_{n+1}, 1 - \alpha_{n+1}\} + \gamma}{m_1 \gamma} \tag{2.28}$$

by taking $\lim_{m_1 \to +\infty}$ [5]. To prove these theoretical guarantees the quantile functions in Equations (2.24) and (2.26) use Dirac delta instead of the Kronecker delta introduced in Equation (2.17) to avoid discontinuity.

Zaffran et al. [26] provide an extensive analysis of the validity, the efficiency and the general behavior of ACI in various situations. They argue that forecasting of dependent examples, even in the absence of distribution shifts, can benefit from the theoretical framework created by Gibbs and Candès [5]. In addition, they propose two methods to automatically select a value for the learning rate γ. The first one consists in computing the quantiles for $K \in \mathbb{N}$ different values of γ and, for every new object, choosing the one that has produced the smallest intervals in the past while retaining validity. The second one is called *Online Expert Aggregation on ACI* (AgACI). Instead of just choosing the learning rate γ that performed best in the past, the intervals for different values of γ are aggregated using an on-line aggregation rule. Zaffran et al. [26] compare their method to ACI, EnbPI and semi-off-line ICP. They show that simply choosing the value for γ that was most efficient in the past does not lead to valid results. In their experiments EnbPI and semi-off-line ICP loose validity, while AgACI is more efficient than ACI and (almost) retains validity.

Feldman et al. [4] present a methods they call *rolling conformal inference* (Rolling CI) that aims to avoid splitting the known examples into a proper training set and a calibration set as is done for ICP. It builds directly upon ACI [5] by removing the calibration set and and adapting the interval width in a fully on-line manner. Assuming n known examples $z_{1:n}$, Rolling CI uses all of them to fit the underlying model $\hat{\mu} = h_{\lfloor z_{1:n} \rceil}$. It then constructs the prediction region

$$\hat{\mathbf{y}}_{n+i} = f(x_{n+i}, \theta_{n+i}, \hat{\mu}_{n+i}) \tag{2.29}$$

for every new object x_{n+i} that is observed sequentially using an interval construction function f that takes the new object x_{n+i}, a calibration parameter θ_{n+i} and the current underlying model $\hat{\mu}_{n+i}$ as inputs. The calibration parameter

$$\theta_{n+i+1} = \theta_{n+i} + \gamma(\text{err}_{n+i} - \alpha) \tag{2.30}$$

is updated after the label of y_{n+i} is revealed. This calibration parameter θ_{n+i} controls the size of the predicted region, similar to α_{n+i} in the ACI [5] method. Finally, Rolling CI updates the underlying model to $\hat{\mu}_{n+i+1}$ using a single gradient step if the model supports it. Feldman et al. [4] show, that for Rolling CI to produce valid intervals according to

$$\lim_{m_1 \to +\infty} \frac{1}{m_1} \sum_{i=1}^{m_1} \text{err}_{n+i} = \alpha \tag{2.31}$$

the only condition is the existence of two constants c_1, c_2 such that $f(x, \theta, \hat{\mu})$ returns an empty interval for $\theta < c_1$ and $\mathcal{Y}$ for $c_2 < \theta$. These requirements are

fulfilled by the quantile functions in Equations (2.24) and (2.26) of ACI. They also show that

$$\left| \frac{1}{m_1} \sum_{i=1}^{m_1} (\text{err}_{n+i} - \alpha) \right| \leq \frac{\max\{\theta_{n+1} - c_1, c_2 - \theta_{n+1}\} + \gamma}{m_1 \gamma} \tag{2.32}$$

for a finite number of examples m_1. In contrast to the other methods, Rolling CI can directly be applied to a multi-output regression setting and therefore also to a multi-step ahead time-series prediction setting. In this case, the parameter θ inflates or deflates the predicted region in all dimensions.

Barber et al. [2] directly address the two exchangeability conditions of the original conformal prediction method. Their *non-exchangeable conformal prediction* (nexCP) method does not require the examples to be exchangeable and does not require that they are treated symmetrically by the algorithm fitting the underlying model. A symmetric model fitting algorithm produces the same model[5] independent of the order of the examples in the training set. Barber et al. [2] apply their developments to CPs, ICPs, and the jackknife+ method. Due to its computational efficiency, I will focus on the adaptation of ICPs. In the semi-off-line setting, the method splits the $n = m_0 + m_c$ known examples into a proper training set D_{train} of size m_0 and a calibration set D_{cal} of size m_c. The underlying model $\hat{\mu} = h_{D_{train}}$ is fitted to the proper training set and the calibration set is used to compute the residuals $r_{m_0+1:n}$. By using the weighted quantile function introduced in Equation (2.26), the method adapts the generated prediction intervals to shifts in distribution without changing the value of α that is passed to the quantile function. This is achieved by assigning residuals of more recent examples higher weights than residuals of older ones. Assuming a test object x_{n+1} with the associated label y_{n+1}, Barber et al. [2] show that the coverage gap as defined in Equation (2.27) is bound by

$$\text{coverage gap} \leq \frac{\sum_{i=m_0+1}^{n} w_i \cdot \mathsf{d_{TV}}(D, D^{[i]})}{1 + \sum_{i=m_0+1}^{n} w_i} \tag{2.33}$$

where $D = D_{cal} \cup z_{n+1}$ and $D^{[i]}$ swaps elements i and $n+1$ in D. $\mathsf{d_{TV}}$ is the total variation distance. This means more generally, that by reducing the weights w_i associated with residuals of examples that contribute to an increase in the coverage gap, the validity of the predicted intervals can be improved. While a careful selection of the weights w_i leads to a reduction in the coverage gap and thereby addresses the exchangeability of the examples it does not lift the requirement, that the model fitting algorithm for the underlying model treats the examples symmetrically. This can be ignored for ICPs as the solution to lift this restriction proposed by Barber et al. [2] does not change the method beyond what has already been discussed.[6] Barber et al. [2] show further that

$$\mathsf{P}(y_{n+1} \in \hat{\mathbf{y}}_{n+1}) \geq 1 - \alpha - \sum_{i=m_0+1}^{n} \tilde{w}_i \cdot \mathsf{d_{TV}}(r_{1:n}, r_{1:n}^{[i]}) \tag{2.34}$$

where $\tilde{w}_i = w_i / \left(\sum_{i=m_0+1}^{n} w_i\right)$ are normalized weights, $r_{1:n}$ are the residuals computed on D and $r_{1:n}^{[i]}$ the residuals computed on $D^{[i]}$. By setting $w_i = 1$, $\forall i \in$

[5]The model fitting algorithm needs to produce models with the same distribution for randomized algorithms.

[6]Please refer to the original paper for the changes necessary for CP and jackknife+.

$\{m_0 + 1, \ldots, n\}$, these results also give an upper limit for the miscoverage rate of the original ICP in case the exchangeability condition is violated. Barber et al. [2] test their method applied to a CP in three settings, comparing it to the standard CP. In the first setting, using i.i.d. examples, they show that their method performs similarly to the standard CP. In the second and third setting, the examples present change points and distribution shift respectively, and nexCP retains approximately valid coverage while the standard CP does not.

Stankevičiūtė et al. [18] extend ICP to univariate multi-horizon time-series forecasts. In this situation, the objects

$$x_i = o^{(i)}_{1:T_w} \in \mathbb{R}^{T_w} \tag{2.35}$$

are the observations made over T_w time steps and the labels

$$y_i = o^{(i)}_{T_w+1:T_w+T_h} \in \mathbb{R}^{T_h} \tag{2.36}$$

are the observations made during the following T_h time steps. The superscript (i) indicates that the observation $o^{(i)}$ belongs to the ith example. Given a set of exchangeable examples $z_{1:n}$, they follow the ICP procedure and split them into a proper training set D_{train} of size m_0 and a calibration set D_{cal} of size m_c. Then they fit an underlying model $\hat{\mu} = h_{D_{train}} : \mathbb{R}^{T_w} \to \mathbb{R}^{T_h}$ to the training set and compute the residuals $r_i = |y_i - \hat{\mu}(x_i)| \in \mathbb{R}^{T_h}$ fore the calibration set. For a new object x_{n+1} the predicted region

$$\hat{\mathbf{y}}_{n+1} = \left[\hat{\mu}(x_i)_j \pm \mathsf{Q}_{1-\alpha/T_h}(r_{m_0+1:n,j}),\ \forall j \in \{1, \ldots, T_h\}\right] \in \mathbb{R}^{T_w \times 2} \tag{2.37}$$

is a T_w dimensional hyper-rectangle, where every dimension corresponds to one time step in the prediction horizon. T_h conformal predictors construct this hyper rectangle, one for each dimension. These conformal predictors use only the residuals associated with that particular time step in the prediction horizon from the examples in the calibration set in their quantile function. Stankevičiūtė et al. [18] apply the Bonferroni Correction, setting the confidence level of the quantile function to $1-\alpha/T_h$ instead of $1-\alpha$. This guarantees, that the predicted region is valid in general, meaning that

$$\mathsf{P}(y_{n+1} \in \hat{\mathbf{y}}_{n+1}) \geq 1 - \alpha. \tag{2.38}$$

This approach allows for dependence within the $o^{(i)}_{1:T_w+T_h}$ that form an example z_i, as only the z_i have to be exchangeable. Designed with recurrent neural networks (RNNs) as underlying models in mind, Stankevičiūtė et al. [18] call their method CF-RNN. They test their method for different RNN based underlying models on synthetic and real world data sets and show that the produced intervals are valid.

2.3 Discussion

Prediction intervals for time-series should, as is the case for conformal predictors in general, be valid and efficient. Due to the nature of time-series, a method generating these intervals needs to support non-exchangeable data and produce intervals for more than one feature and more than one future time step. Table 2.1

offers a comparison of the methods discussed in Chapter 2 according to these criteria. Ideally it would also be computationally efficient (or at least feasible in practice) and user-friendly by providing guidance on how to set the method's parameters.

Method	Coverage Guarantee	Multi-Target	NE Data
CP [24]	$1-\alpha$	✗	✗
ICP [24]	$1-\alpha$	✗	✗
jackknife (LOO) [1]	–	✗	✗
jackknife+ [1]	$1-2\alpha$	✗	✗
CIbP [3]	$1-\alpha$	✓	✻
EnbPI [25]	$1-\alpha$	✗	✗
ACI [5]	$1-\alpha$	✗	✓
AgACI [26]	–	✗	✓
Rolling CI [4]	$1-\alpha$	✓	✓
nexCP [2]	$1-\alpha$	✗	✓
CF-RNN [18]	$1-\alpha$	✓	✻

Table 2.1: Comparison between the different methods discussed in Chapter 2. Columns: *Coverage Guarantee* displays the long-term theoretical coverage guarantee the original authors provide for their methods. *Multi-Target* displays if the method supports multi-target regression, i.e. multiple time steps or multiple features in a time-series setting. *NE Data* displays if the method retains the coverage guarantee in the presence of non-exchangeable data. ✻ These methods allow for short term dependencies, but lose their validity guarantee if applied to time-series with long term dependencies.

While other methods to convey uncertainty in the form of prediction intervals exist [8], they do not come with the validity guarantees that some methods derived from CPs offer and that are desired in high stakes situations [4]. Gibbs and Candès [5] have shown that these other methods can be used as underlying models for CP based methods. Because of this, I have chosen to focus on CP based methods in my work.

As mentioned at the beginning of Section 2.2, the lack of exchangeability in time-series data means that the conditions for the validity guaranteed by CP, ICP and the jackknife+ method are not met. CIbP [3] relaxes this condition by splitting the data into blocks consisting of values of consecutive time steps. Chernozhukov et al. [3] assume that these blocks are exchangeable. By regarding each block as a point in a multidimensional space instead of a sequence of consecutive scalar values, with each value corresponding to a time step, the values within the block no longer need to be exchangeable. While some time-series may present such a block wise exchangeability, it is not given in general, especially if the time-series displays long term dependencies. Another drawback of the proposed method is that by evaluating all possible values for all dimensions of the label space, the method considers an infinite number of label candidates and is therefore not applicable in practice. CF-RNN [18] overcomes the computational complexity of CIbP [3] by estimating the intervals for every dimension in the label space individually and correcting for the family-wise error rate. CF-RNN [18] relies on a block wise exchangeable structure in the same way as CIbP [3] making it unsuitable for time-series with long term dependencies or a

change in distribution over time.

EnbPI [25] takes a different approach to relaxing the exchangeability condition for the examples by requiring the residuals to be stationary and strongly mixing. This assumption is only met if the underlying model maintains the quality of its predictions when the distribution of the data changes over time. Because this is difficult to prove in general, EnbPI [25] is listed as not retaining coverage in presence of non-exchangeable data in Table 2.1. Xu and Xie [25] apply EnbPI to the multi-step ahead setting by predicting the future time steps in the prediction horizon individually. If the method's conditions are met, this approach guarantees validity for every individual dimension of the label space, but it does not guarantee validity in general according to the definition given in Equation (2.9). This is why Table 2.1 lists EnbPI [25] as not supporting multi-target regression.

By growing or shrinking the generated intervals according to the coverage errors it has made in the past, ACI [5] achieves the desired coverage frequency in the long run without making any assumptions about the data or the errors the underlying model makes. This achievement comes at the cost of a quantile function that can return infinite values resulting in a predicted region that covers the entire label space which is uninformative. The version of the method using the unweighted quantile function requires only a single additional parameter, the learning rate γ, adding to its ease of use. ACI [5] is also computationally efficient as it takes advantage of all advancements made for ICPs. These advancements limit the method to single-target forecasts, meaning it is only applicable in the univariate single-step ahead setting.

Multiple extensions for ACI [5] are proposed. AgACI [26] automatically estimates the learning rate γ thereby achieving better efficiency. This requires the choice of an on-line aggregation method and does not address any of the other shortcomings of ACI [5]. Rolling CI [4] is the second extension of ACI [5]. It does not require a split of the known examples, allowing the the underlying model to be fitted to all of them. Rolling CI [4] also allows for multi-target forecasts by growing or shrinking the predicted region in all dimensions. If a shift in the distribution only occurs for a subset of features, this might lead to overly conservative intervals for the others.

nexCP [2] could be seen as a third extension of ACI [5]. It departs from the idea of adapting the α used in the quantile function based on the observed error rate of the method, and focuses instead on the weighted quantile function. This approach comes with strong theoretical guarantees. However, just as ACI [5], nexCP [2] may produce infinite prediction intervals and is limited to single-target forecasts. In addition, more research is required to determine the optimal weights for the weighted quantile function.

Applying conformal prediction to multivariate multistep-head time-series forecasts poses two challenges. The first being that time-series are generally non-exchangeable, and the second, that it is not sufficient to produce a prediction interval but instead a predictive region needs to cover all dimensions of the label space. Different methods have been proposed to address these challenges individually. And while Rolling CI [4] addresses both simultaneously, I believe that it can further be improved as it adapts the predicted region in all dimensions over time without considering which features are responsible for a change in coverage rate.

Chapter 3

nmtCP Method

This chapter aims to address some of the gaps identified in previous methods that are outlined in Chapter 2.3. To do this, I start by taking a deeper look at the difficulties that the conformal multi-target regression task presents in Chapter 3.1. Chapter 3.2 lays out my idea on how to address these difficulties. Chapter 3.3 takes this idea, turns it into a method and explains how to implement it in practice. This is followed by a theoretical analysis of the validity of the method in Chapter 3.4 and a discussion and comparison with related methods in Chapter 3.5.

3.1 Conformal Prediction and Multi-Target Regression

Most of the work presented in Sections 2.2 and 2.3 addresses the non-exchangeability of time-series data and how conformal prediction can be adapted to produce valid prediction intervals in the presence of non-exchangeable data. Before exposing my method, I will briefly lay out why it is difficult to apply conformal prediction to multi-target regression.

Equation (2.4) shows that the NCM returns a scalar value which allows for the computation of associated p-values in Equation (2.6). According to Equation (2.7) the predicted region for an object x_{n+1} contains all possible elements in the label space that satisfy $\{y|p_{z_{n+1}} > \alpha\}$. While this definition of CPs is not tied to any particular setting, applying it naively is impractical as it implies computing the nonconformity score for an infinite number of possible labels y [19]. For single-target regression, this is overcome in two steps, detailed in Section 2.1.2. The first step is realizing that the set of values the p-value can take is finite. This is true for both single-target and multi-target regression [19]. Fixing the p-value $p_{z_{n+1}}$ means selecting all possible labels y such that the nonconformity score

$$r_{n+1} \leq r_j \tag{3.1}$$

where r_j is chosen among known nonconformity scores such that $p_{z_{n+1}} > \alpha$. The second step is finding the values of the label y that produce nonconformity scores that satisfy the inequality in Equation (3.1) [19]. For single-target regression,

when using the absolute residuals as NCM, this is achieved by simply solving

$$|y - \hat{\mu}(x_{n+1})| \leq r_j. \tag{3.2}$$

Solving Equation (3.2) for y produces exactly one region, the prediction interval $[\hat{\mu}(x_{n+1}) - r_j, \hat{\mu}(x_{n+1}) + r_j]$, as solution. This shows that there exists a bijective relationship between the upper and lower bound of the prediction interval and the nonconformity score r_{n+1} associated with the p-value necessary to reach the desired coverage rate. This bijective relationship is important because it guarantees that there is exactly one solution to Equation (2.7), i.e. exactly one region that satisfies the desired coverage rate and that it is inexpensive to compute.

In the multi-target regression setting I have not been able to find such a bijective relationship. Replacing the absolute residuals by the Euclidean distance as NCM for a label space $\mathcal{Y} \subset \mathbb{R}^{F}$[1] is a natural choice. However, given the value of the Euclidean distance $r_{n+1} \in \mathbb{R}$ and a predicted vector $\hat{y}_{n+1}$ there are an infinite number of regions satisfying the condition

$$\{y || \hat{y}_{n+1} - y| \leq r_{n+1}\}, \tag{3.3}$$

exemplifying the issue.

Vovk [22] and later Stankevičiūtė et al. [18] circumvent this issue by calibrating one CP for every dimension in the label space, reverting to single target regression and a one-dimensional label space for each. Equation (2.37) shows how they combine the the individual prediction intervals generated by these CPs for every dimension into a multi-dimensional prediction region.

3.2 Idea

Applying the concepts of conformal prediction to multistep-ahead multivariate time-series forecasting faces two main obstacles. Sections 2.2 and 2.3 show the challenges that the nonexchangeability of the data in time-series poses and discuss methods to overcome them. Section 3.1 briefly explains the additional difficulty introduced when moving from single target to multi-target regression, necessary to achieve multistep-ahead multivariate forecasts.

In this chapter, I propose a new method that addresses both of these obstacles by combining two existing methods. Barber et al. [2] has provided a method that relaxes the exchangeability assumption while maintaining strong validity guarantees that is computationally efficient for single target regression. On the other hand, Stankevičiūtė et al. [18] shows how multiple conformal predictors can be combined to produce prediction regions for multi-target regression. My idea is to merge these two approaches, replacing the CPs in the method of Stankevičiūtė et al. [18] with nexCPs [2]. This leads to a method that produces prediction intervals for multi-step ahead multivariate time-series forecasts for any underlying model producing point forecasts for every dimension in the label space.

[1] F stands for an arbitrary number of dimensions here and not necessarily the number of considered time-series.

3.3 Implementation

This section explains how I put the idea outlined in Section 3.2 into practice, starting with the representation and preprocessing of the time-series data in Section 3.3.1. It also expands the notation introduced in Section 2.2. Then Section 3.3.2 provides a detailed look into the generation of the predictive region in the semi-off-line setting. Finally other settings are discussed in Section 3.3.3.

3.3.1 Preprocessing

When loading the data it is generally stored in a table, where each line contains the values of the observed features for a specific time step and each column contains all measurements for a specific feature. This table can be represented as $o_{1:T,1:F}$, where T is the total number of time steps and F the total number of features.

$$\begin{matrix} o_{1,1} & o_{1,2} & \dots & o_{1,F} \\ o_{2,1} & o_{2,2} & \dots & o_{2,F} \\ \vdots & \vdots & \ddots & \vdots \\ o_{T,1} & o_{T,2} & \dots & o_{T,F} \end{matrix} \tag{3.4}$$

I split this *raw data set* into examples $z_i = (x_i, y_i)$ consisting of objects x_i and their corresponding labels y_i. The objects x_i are created following a sliding window scheme. The ith object consists of

$$x_i = o_{i \cdot s : i \cdot s + T_w, 1:F} \tag{3.5}$$

starting at $i \cdot s$ where s is the stride, describing the number of time steps between the starting points of two consecutive objects, i.e. windows. T_w is the number of time steps each object contains and F is the number of input features for the model.[2] Consecutive objects overlap, unless $s \geq T_w$. Equation (3.7) visualizes the object x_i as a blue block. The label associated with the ith object is

$$y_i = o_{i \cdot s + T_w + 1 : i \cdot s + T_w + T_h, 1:F_h} \tag{3.6}$$

where T_h is the number of time steps in the prediction horizon and F_h is the number of features predicted by the model.[3] Assuming for visualization purposes that $F_h = 2$, y_i is represented by the green block in Equation (3.7). Note,

[2] I assume that all features, i.e. all time-series are used as input to the underlying model. If this is not the case F can be replaced with $F_w \subset 1 : F$.

[3] To avoid an overly complex notation I assume that the first F_h features are the ones being predicted. This can always be guaranteed by re-ordering the features. Although it is suggested by the representation in Equation (3.7), the predicted features do not need to be a subset of the features used for the labels x_i.

that the label starts at the first time step immediately after the object.

$$
\begin{array}{cccc}
o_{1,1} & o_{1,2} & \dots & o_{1,F} \\
o_{2,1} & o_{2,2} & \dots & o_{2,F} \\
\vdots & \vdots & \ddots & \vdots \\
o_{i\cdot s-1,1} & o_{i\cdot s-1,2} & \dots & o_{i\cdot s-1,F} \\
o_{i\cdot s,1} & o_{i\cdot s,2} & \dots & o_{i\cdot s,F} \\
o_{i\cdot s+1,1} & o_{i\cdot s+1,2} & \dots & o_{i\cdot s+1,F} \\
\vdots & \vdots & \ddots & \vdots \\
o_{i\cdot s+T_w,1} & o_{i\cdot s+T_w,2} & \dots & o_{i\cdot s+T_w,F} \\
o_{i\cdot s+T_w+1,1} & o_{i\cdot s+T_w+1,2} & \dots & o_{i\cdot s+T_w+1,F} \\
\vdots & \vdots & \ddots & \vdots \\
o_{i\cdot s+T_w+T_h,1} & o_{i\cdot s+T_w+T_h,2} & \dots & o_{i\cdot s+T_w+T_h,F} \\
o_{i\cdot s+T_w+T_h+1,1} & o_{i\cdot s+T_w+T_h+1,2} & \dots & o_{i\cdot s+T_w+T_h+1,F} \\
\vdots & \vdots & \ddots & \vdots \\
o_{T,1} & o_{T,2} & \dots & o_{T,F}
\end{array}
\tag{3.7}
$$

After processing the raw data set in this manner, $\mathcal{D} \coloneqq x_{1:n}$ designates the data set consisting of $n \coloneqq |\mathcal{D}|$ examples.

3.3.2 Semi-Off-Line Setting

Three steps are necessary to apply the idea in the semi off-line setting.

Step 1, Training After obtaining the data set $\mathcal{D}$, I partition it into a proper training set D_{train} and a calibration set D_{cal} along the time axis. $m_0 = |D_{train}|$ and $m_1 = |D_{cal}|$ designate the sizes of the proper training set and the test set respectively with $m_0 + m_1 = n$. The partition of the data set is drawn along the time axis and the elements within D_{train} and D_{cal} are not shuffled. Next, I fit the underlying model to the proper training set D_{train}. The underlying model is only fitted once in the semi-off-line setting.

Step 2, Initial Calibration I use the fitted model $\hat{\mu}$ to generate predictions $\hat{y}_i = \hat{\mu}(x_i)$ for all elements $x_i \in \mathcal{D}_{cal}$ in the calibration set. The chosen NCM Δ as defined in Equation (2.12) computes a nonconformity score $r_{i,t,f} = \Delta(y_{i,t,f}, \hat{y}_{i,t,f})$ for every dimension of these elements. For this thesis, I use the absolute residuals as NCM. These residuals are then used to calibrate $T_h \cdot F_h$ nexCPs [2] denoted by

$$
\Gamma^{\alpha'}_{t,f}, \quad \forall t = 1, \dots, T_h, \ \forall f = 1, \dots, F_h. \tag{3.8}
$$

For any combination of fixed t and f, the associated nexCP $\Gamma^{\alpha'}_{t,f}$ initially uses the nonconformity scores $r_{m_0:n,t,f}$. This means that the nexCP associated with one dimension of the labels space only uses the nonconformity scores computed on that particular dimension of the label space. Instead of using α as the targeted miscoverage rate, the conformal predictors use the Bonferroni Correction resulting in

$$
\alpha' = \frac{\alpha}{T_h \cdot F_h} \tag{3.9}
$$

to account for the family-wise error rate.

Step 3, Inference Presented with a new object $x_i, i > n$, the underlying model generates a prediction for the associated label $\hat{y}_i = \hat{\mu}(x_i)$. Then the $T_h \cdot F_h$ nexCPs [2] $\Gamma^{\alpha'}_{1:T_h,1:F_h}$ compute an interval for every dimension in the label space. They do this by applying the weighted quantile function to the absolute residuals $r_{m_0:i-1,t,f}$ of the associated dimension t, f. I combine the predicted intervals to form the predicted region $\hat{y}_i \in \mathbb{R}^{T_h \times F_h \times 2}$. In the semi-off-line setting nature provides the true label y_i, which I use to compute the residuals $r_{i,1:T_h,F_h}$. These new residuals will be used when the algorithm generated prediction interval for the next example n_{i+1}.

Input: $\mathcal{D} \in \mathcal{Z}^n$ // data set
$m_0 \in \mathbb{N}$ // proper training set size
$\alpha \in [0, 1]$ // significance level
$W \in \mathbb{N} \to \mathbb{R}^*$ // weight generating function
Output: $\hat{\mathbf{y}}_{n+1:n+*} \subset \mathcal{Y}^{2 \times *}$

$\mathcal{D}_{train} \leftarrow z_{1:m_0} \in \mathcal{Z}^{m_0}$ // proper training set
$\mathcal{D}_{cal} \leftarrow z_{m_0+1:n} \in \mathcal{Z}^{n-m_0}$ // calibration set
$\hat{\mu} \leftarrow h_{\mathcal{D}_{train}}$ // fit the underlying mode

for $i \leftarrow m_0$ **to** n // calibrate the nexCPs
do
  $r_i \leftarrow |y_i - \hat{\mu}(x_i)|$ // element-wise absolute difference
end

for $i \leftarrow n + 1$ **to** $*$ // loop for every new element x_i
do
  $w_{m_0:i} \leftarrow W(i - m_0)$ // generate weights
  $\tilde{w}_{m_0:i} \leftarrow w_{m_0:i} / \sum w_{m_0:i}$ // normalize weights

  $y_{i-1} \leftarrow$ observe_reality
  for $t \leftarrow 1$ **to** T_h **do**
    for $f \leftarrow 1$ **to** F_h **do**
      $r_{i-1,t,f} \leftarrow \Delta(y_{i-1,t,f} - \hat{\mu}(x_{i-1})_{t,f})$
    end
  end

  $\hat{y}_i \leftarrow \hat{\mu}(x_i)$
  for $t \leftarrow 1$ **to** T_h **do**
    for $f \leftarrow 1$ **to** F_h **do**
      $\hat{\mathbf{y}}_{i,t,f} \leftarrow \hat{y}_{i,t,f} \mp \mathsf{Q}_{1-\frac{\alpha}{T_h \cdot F_h}}(r_{m_0:i,t,f}, \tilde{w}_{m_0:i})$
    end
  end
  provide $\hat{\mathbf{y}}_i$
end

Algorithm 2: Semi-off-line adaptive multi-target conformal prediction. The underlying model is only trained once but the calibration set grows with every new label that is provided to the method.

Algorithm 2 describes these three steps in a way that can directly be imple-

mented.[4] In the presence of a pretrained model $\hat{\mu}$, the first step can be omitted entirely and the entire initial data set $\mathcal{D}$ can be used as calibration set. This holds as long as none of the examples from the data set $\mathcal{D}$ were used during training of the model $\mathcal{D}$.

3.3.3 On-Line, On-Line Batch and Off-Line Setting

The method also works in the other settings laid out in Section 2.1.3. Only minor changes to the steps shown in Section 3.3.2 are necessary.

In the on-line setting, the partition between proper training set $\mathcal{D}_{train}$ and the calibration set $\mathcal{D}_{cal}$ is redrawn every time a new label becomes available. The underlying model is then fitted to the new training set $\mathcal{D}_{train}$ and the nexCPs $\Gamma^{\alpha'}_{1:T_h,1:F_h}$ are calibrated on $\mathcal{D}_{cal}$. The prediction step omits the computation of the residuals when nature provides a new label as this action is already covered by the calibration step.

The on-line batch setting is a hybrid of the on-line and semi-off-line setting. As new examples become available, they are periodically split into a proper training set $\mathcal{D}_{train}$ and a calibration set $\mathcal{D}_{cal}$ along the time axis. When this happens, the underlying model is fitted to the new training set $\mathcal{D}_{train}$ and the nexCPs $\Gamma^{\alpha'}_{1:T_h,1:F_h}$ are calibrated on $\mathcal{D}_{cal}$. Then the method operates in the semi-off-line setting until the examples are partitioned again the the underlying model is retrained.

Although the method can operate in the off-line setting, it is not useful, as the information provided by new labels remains unused. Over time, it would not adapt to distribution shifts in the data.

3.4 Validity

Similar to the argument made by Stankevičiūtė et al. [18], I view each conformal predictor as having been calibrated on a dedicated calibration set, using the absolute residuals for the associated dimension as nonconformity scores. However, instead of using standard CPs, my method employs nexCPs. For nexCPs Barber et al. [2] state in Theorem 2a that in the inductive conformal prediction setting

$$\mathsf{P}\left(y_{n+1,t,f} \in \hat{\mathbf{y}}_{n+1,t,f}\right) \geq 1 - \frac{\alpha}{T_h \cdot F_h} - G_{n_0+1:n,t,f} \tag{3.10}$$

for a new object n_{n+1}, where the coverage gap $G_{n_0+1:n,t,f} = \sum_{i=n_0+1}^{n} \tilde{w}_i \cdot \mathsf{d}_{\mathsf{TV}}(r_{n_0+1:n,t,f}, r^{[i]}_{n_0+1:n,t,f})$ and $r^{[i]}$ swaps elements i and n in r. This is equivalent to

$$\mathsf{P}\left(y_{n+1,t,f} \notin \hat{\mathbf{y}}_{n+1,t,f}\right) \leq \frac{\alpha}{T_h \cdot F_h} + G_{n_0+1:n,t,f} \tag{3.11}$$

[4] For my implementation I vectorised operations where possible. In addition my implementation allows for the evaluation of multiple values for α in parallel, reducing the number of times a potentially complex underlying models needs to be run.

providing an upper bound for the miscoverage rate for every dimension in the label space. Using Boole's inequality,

$$\mathsf{P}\left(y_{n+1} \notin \hat{\mathbf{y}}_{n+1}\right) = \mathsf{P}\left(\cup_{t=1}^{T_h} \cup_{f=1}^{F_h} \left(y_{n+1,t,f} \notin \hat{\mathbf{y}}_{n+1,t,f}\right)\right) \tag{3.12}$$

$$\leq \sum_{t=1}^{T_h} \sum_{f=1}^{F_h} \mathsf{P}\left(y_{n+1,t,f} \notin \hat{\mathbf{y}}_{n+1,t,f}\right) \tag{3.13}$$

$$\leq \sum_{t=1}^{T_h} \sum_{f=1}^{F_h} \frac{\alpha}{T_h \cdot F_h} + G_{n_0+1:n,t,f} \tag{3.14}$$

$$= \alpha + \sum_{t=1}^{T_h} \sum_{f=1}^{F_h} G_{n_0+1:n,t,f} \tag{3.15}$$

bounds the miscoverage rate for the predicted region constructed from the individual nexCPs' intervals.

Equation (3.15) shows that the upper bound on the miscoverage rate of the proposed method depends on the targeted miscoverage rate α and the sum of the coverage gaps $G_{n_0+1:n,t,f}$ of the individual nexCPs. The coverage gaps $G_{n_0+1:n,t,f}$ of the individual nexCPs depend on the chosen weights $\tilde{w}_i$. This means that if the weights are chosen carefully, the coverage gaps $G_{n_0+1:n,t,f}$ decrease and the method becomes approximately valid [13] for slow shifts in distribution. Optimising the weighs directly $\tilde{w}_i$ is not possible, as the computation of the total variation distance $\mathsf{d_{TV}}$ requires knowledge of the true label y_{n+1}. However, assigning higher weights to past examples similar to the one for which the method is predicting the label and lower ones to past examples that are less similar leads to a decrease in coverage gap $G_{n_0+1:n,t,f}$ and give the weights their intuitive interpretation [2].

The theoretical upper bound on the method's miscoverage rate presented in Equation (3.15) does not require the same weights $\tilde{w}_i$ for every dimension. While this gives the user a lot of control over the method, I did not explore this possibility in the context of this thesis. Following the advice of Barber et al. [2], I opted for a few simple weight generating functions for the experiments presented in Chapter 4.

3.5 Comparison

This section compares my method to the original CP [24] and highlights some key differences with the methods discussed in Chapter 2. Henceforth, I will refer to my method as presented in Section 3.3.2 as non-exchangeable multi-target conformal prediction or nmtCP.

In contrast to the original CP [24], nmtCP does not require the underlying data to be exchangeable, and it can still benefit from the gains in computational efficiency presented in Section 2.1.2 even while being applied in a multi-target regression setting. This comes at the cost of a weakened validity guarantee.

CIbP [3] and CF-RNN [18] rely on the presence of exchangeable blocks in the time-series for their respective validity guarantees. nexCP [2] and therefor nmtCP makes no such assumption and can therefore be applied to time-series that exhibit long term dependencies or sudden distribution shifts.

EnbPI [25] is computationally more expensive than nmtCP because it trains an ensemble of underlying models. The benefit to this approach is, that it avoids splitting the known examples into a proper training set and a calibration set. In the semi-off-line setting EnbPI [25] behaves like nexCP [2] and therefore, like nmtCP with a weight function that assigns the same weight to a fixed number of past residuals. The weighted quantile function used in nexCP [2] and nmtCP can be interpreted as a generalization of the sliding window approach that Xu and Xie [25] chose for the integration of new residuals in EnbPI. Because Xu and Xie [25] built their validity guarantees on assumptions about the residuals, it is difficult to compare them to the one presented in Equation (3.15). However, nmtCP offers more flexibility to determine how the method reacts to distribution shifts through the weights w_i in the weighted quantile function, even if the performance of the underlying model varies over time.

ACI [5], being its direct precursor, is very similar to nexCP [2]. It is capable to adapt to distribution shifts, is computationally efficient, comes with strong validity guarantees and offers the flexibility of the weighted quantile function. Just as nexCP [2] and in contrast to nmtCP it lacks the support for multi-target regression.

In comparison to nmtCP, Rolling CI [4] does not need to split the known examples into a proper training set and a calibration set, allowing the underlying model to be fitted to the most recent data points. Rolling CI [4] also includes a training step for the underlying model every time a new label becomes available, making it highly efficient in an on-line setting. It behaves similar to ACI [5], scaling the size of the prediction region over time. However, for multi-target regression the prediction region is scaled equally in all dimensions. This might not be desirable as not all dimensions necessarily contribute equally to a change in coverage rate. nmtCP does not suffer from this behavior since the intervals for the individual dimensions in the label space are constructed independently and then combined into the prediction region.

Overall nmtCP offers a flexible solution that adapts to distribution shifts in the data. It can be used on top of any multi-target regression model with little to no modifications to the underlying model. The miscoverage rate of nmtCP is bound theoretically. This set of characteristics tailored to multistep-ahead multivariate time-series forecasting [17] is not found in the other methods examined in this thesis. It comes at the expense of weaker validity guarantees or splitting the data, reducing the effective training set size for the underlying model.

Chapter 4

Experiments

In this chapter, I examine the empirical performance of the nmtCP method presented in Chapter 3 by applying it to two real-world time-series forecasting tasks and comparing it to CF-RNN [18]. Chapter 4.1 introduces the two data sets, and explains the preprocessing steps taken to prepare them for the use with nmtCP. Chapter 4.2 describes how the experiments are conducted and what variants of the method I used, which is followed by the metrics used to evaluate the results in Chapter 4.3. The results of the experiments are reported in Chapter 4.4 which I discuss in Chapter 4.5.

4.1 Data Sets

Inspired by Rolnick et al. [15], I chose two data sets from the electricity domain to test the method on.

4.1.1 ELEC2

The ELEC2 data set [6] contains information about the electricity demand and price in the Australian states of New South Wales and Victoria. Measures have been taken every 30 minutes between May 1996 and December 1998. For the experiments, I chose to use the first 20 000 time steps in the data set and the features representing the electricity demand in New South Wales, the electricity demand in Victoria, and the amount of electricity transferred between the two states. Figure 4.1 shows this selection and the split between proper training set, calibration set and test set through the vertical dotted lines. The proper training set contains the first 8 000 time steps, the calibration set is made up of the following 8 000 time steps and the test set comprises the last 4 000 time steps.

Before feeding the data set to the underlying model and the nmtCP, I split it into consecutive windows as shown in Chapter 3.3.1 with $T_w = 192$, $T_h = 12$, and a stride of $s = 12$ between consecutive windows, resulting in 1650 examples and a prediction horizon of six hours. Both the object and the corresponding label are made up of all three features, hence $F = F_h = 3$. I split the examples into a proper training set, a calibration set, and a test set containing 660, 660 and 330 examples, respectively, along the time axis. Note that the features

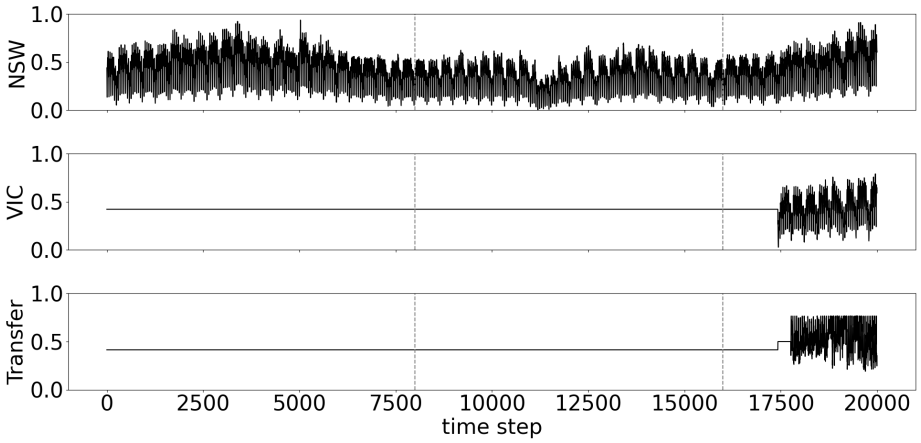

Figure 4.1: Features and time steps from the ELEC2 data set [6] used in the experiments.

VIC and Transfer have a constant value throughout the proper training and the calibration set and display a strong distribution shift in the test set, visible in Figure 4.1.

4.1.2 Tétouan City

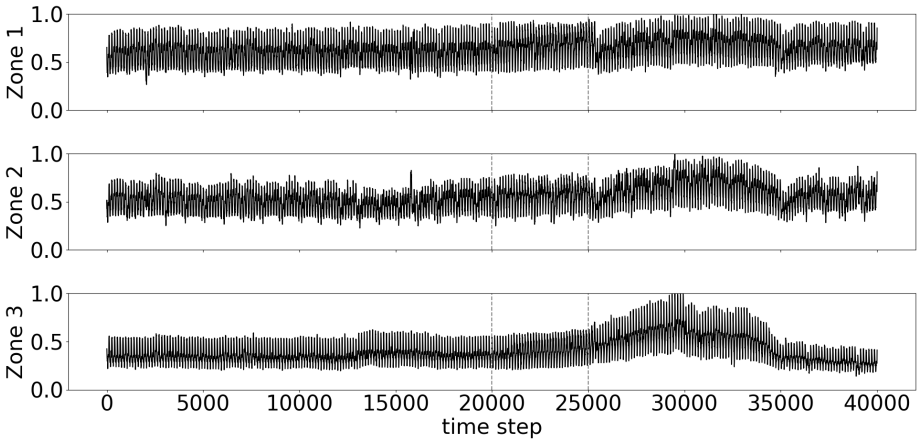

Figure 4.2: Features and time steps from the Tétouan City data set [16] used in the experiments.

The Tétouan City data set [16] contains weather information and the electricity consumption of three different distribution networks of the northern Moroccan city of Tétouan. The measures were taken every 10 minutes in 2017. For the experiments, I chose to use the first 40 000 time steps in the data set and the features representing the electricity consumption for each of the three distribution networks. Figure 4.2 shows this selection and the split between proper training set, calibration set, and test set through the vertical dotted lines. The

proper training set contains the first 20 000 time steps, the calibration set is made up of the following 5 000 time steps and the test set comprises the last 15 000 time steps. The three distribution networks are referred to as Zone 1, Zone 2, and Zone 3 in Figure 4.2.

Similar to the ELEC2 data set [6], I split the Tétouan City data set [16] into consecutive windows with $T_w = 144$, $T_h = 6$, and a stride of $s = 6$ between consecutive windows, resulting in 6642 examples and a prediction horizon of one hour. Again, both, the object and the corresponding label are made up of all three features, hence $F = F_h = 3$. The proper training set, the calibration set, and the test set contain 3321, 830 and 2491 examples, respectively, split along the time axis. Compared to the ELEC2 data set [6], the features of the Tétouan City data set [16] exhibit a more gradual distribution shift in the test set, shown in Figure 4.2.

4.2 Experimental Setup

The experiments follow the process outlined in Chapter 3.3.2.

Step 1, Training I chose linear regression for the underlying model, using PyTorch for the implementation and the Adam optimizer during training. While producing similar results in this task, compared to the recurrent neural networks used in Stankevičiūtė et al. [18] and Schlembach et al. [17], the linear regression model is much faster to train and allowed me to repeat every experiment multiple times.
For both data sets, I train the underlying models on the proper training set for 1 000 epochs, using a batch size of 100 and a learning rate of 0.01.

Step 2, Initial Calibration After the underlying model is fitted to the proper training set, I use the model to generate predictions on the test set. Then, the nexCPs [2] use these predictions to calibrate for every dimension in the label space. To increase efficiency, I calibrate multiple groups of nexCPs [2] with different weights for the quantile functions simultaneously. Chapter 4.2.1 provides a detailed description of the different functions generating these weights. In addition to increasing efficiency, sharing the same underlying model between multiple instances of nmtCP, with different weights for the quantile function, has the added benefit that, when comparing the validity and efficiency of the produced prediction regions, they are based on the same residuals.

Step 3, Inference Finally, I present the underlying model with the objects from the test set sequentially in their original order. The prediction is given to each group of calibrated nexCPs [2] for each to generate a prediction region in the form of a hyper-rectangle. Then I compute the corresponding residual for every dimension in the label space using the true label and add the resulting values to the calibration set of each nmtCP.

After computing all examples in the test, set I evaluate them using the metrics described in Chapter 4.3[1]. All experiments are repeated 20 times and results averaged to account for the random initialization of the underlying model.

[1]The software architecture supports on-line processing of new examples and the on-line evaluation of the results. This feature is not used for the experiments presented in this thesis.

4.2.1 Weight Functions

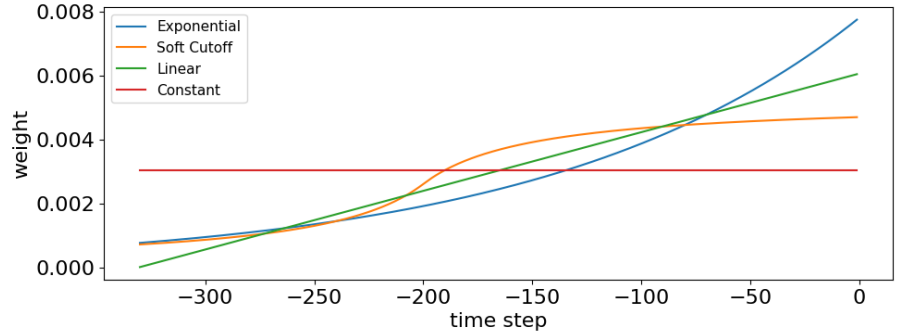

Figure 4.3: Normalized weights produced by different weight functions for the past 300 time steps.

In the semi-off-line setting, the number of residuals used by the weighted quantile function in every nexCP [2] grows with every new true label y_{i-1} that becomes available. Therefore, instead of assigning static weights $\tilde{w}_i$, I used functions to generate the appropriate number of weights every time a new object x_i is processed. In the experiments, I compared the following weight functions.

Exponential The weight of past residuals decreases exponentially the further they are in the past.

$$W(i-m_0) = [e^{\beta(j-i+m_0)}]_{j=0}^{i-m_0} \tag{4.1}$$

where β is a parameter controlling how quickly the value of past weights decreases. In the experiments, $\beta = 0.007$.

Soft Cutoff This weight function mimics a sliding window approach with a soft transition from the past residuals that are considered to the ones that are excluded.

$$W(i-m_0) = \left[\frac{j-i+m_0+\beta_c}{\beta_s + |j-i+m_0+\beta_c|} + 1\right]_{j=0}^{i-m_0} \tag{4.2}$$

where β_c is the cutoff point, indicating how many past residuals are considered and β_s controls the "softness" of the transition. In the experiments $\beta_c = 200$ and $\beta_s = 50$.

Linear The weight of past residuals decreases linearly the further they are in the past.

$$W(i-m_0) = \left[\frac{j}{i-m_0}\right]_{j=0}^{i-m_0} \tag{4.3}$$

Constant All past residuals are given the same constant weight.

Figure 4.3 illustrates the weights these functions generate for 300 past residuals.

4.3 Evaluation

To evaluate the method's validity in the experiments, I used two metrics, the (average) *coverage rate*[2] and the *rolling coverage rate*. Following the definition of validity given in Chapter 2.1.1, the coverage rate computes the fraction of prediction regions $\hat{\mathbf{y}}_i$ that contain the true label y_i for a given confidence level $1-\alpha$. If the coverage rate is greater than $1-\alpha$ the method is valid. The rolling coverage rate applies the same principle using a sliding window approach hence providing a local coverage rate.

The *mean interval width* provides information about the efficiency of the method for a given confidence level $1-\alpha$. In the multistep-ahead multivariate time-series setting, it computes the average difference between the upper and the lower bound of the prediction interval for every dimension in the label space and every example in the test set. A lower value therefor indicates greater efficiency.[3]

I did not apply additional metrics such as the *average miscoverage streak length* [4] or the Δ*coverage* [4].

4.4 Results

4.4.1 ELEC2

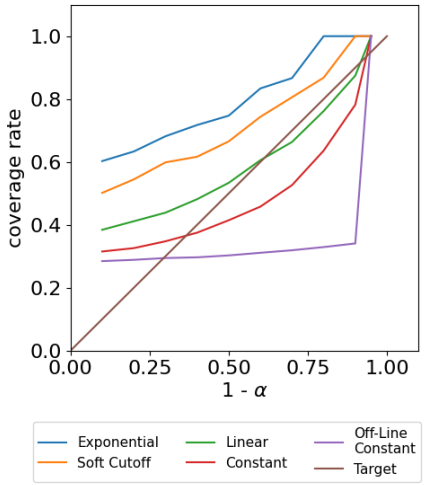

Figure 4.4: Empirical coverage rate measured on the ELEC2 data set [6]. Average taken over 20 trials.

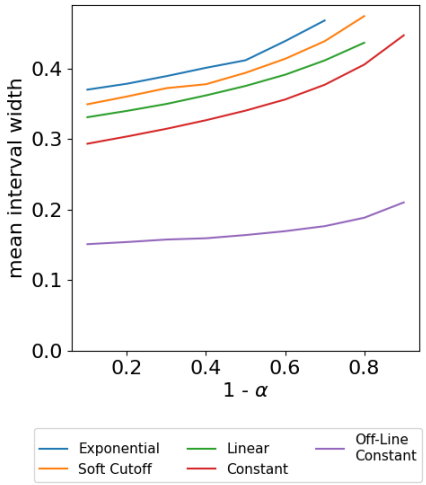

Figure 4.5: Mean interval width measured on the ELEC2 data set [6]. Average taken over 20 trials.

[2] Vovk [23] suggest the use of the calibration plot instead, representing the same information using α and the empirical miscoverage rate instead of the confidence level $1-\alpha$ and the empirical coverage rate. I opted for the latter do to it appearing more intuitive to me.

[3] In addition to applying these metrics globally the user of the nmtCP method can gain further insights into its performance and the behavior of the underlying model by applying them to a subset of the dimensions of the label space.

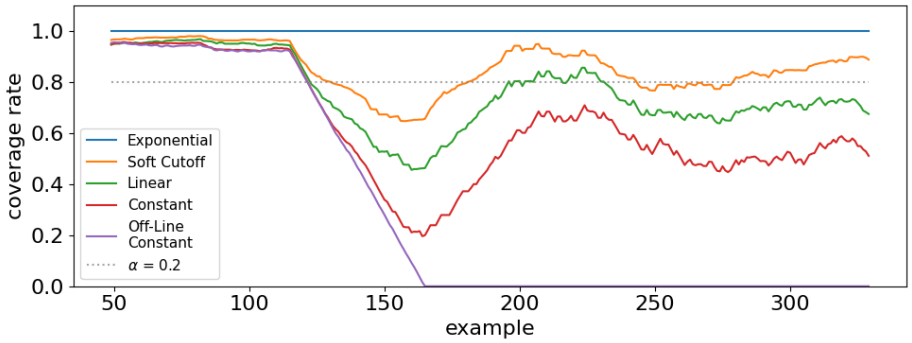

Figure 4.6: Rolling coverage rate with a window size of 50 for different weight functions. Target coverage rate $1 - \alpha = 0.8$. Average taken over 20 trials.

1-α	Exponential	Soft Cutoff	Linear	Constant	Off-Line Constant
0.10	0.603	0.502	0.384	0.315	0.285
0.20	0.633	0.545	0.411	0.326	0.289
0.30	0.682	0.599	0.439	0.348	0.294
0.40	0.717	0.617	0.482	0.375	0.297
0.50	0.747	0.665	0.534	0.415	0.303
0.60	0.834	0.743	0.605	0.458	0.311
0.70	0.867	0.806	0.663	0.526	0.319
0.80	1.000	0.868	0.762	0.636	0.329
0.90	1.000	1.000	0.875	0.782	0.341
0.95	1.000	1.000	1.000	1.000	1.000

Table 4.1: Empirical coverage rate measured on the ELEC2 data set [6]. Average taken over 20 trials.

1-α	Exponential	Soft Cutoff	Linear	Constant	Off-Line Constant
0.10	0.370	0.349	0.331	0.293	0.151
0.20	0.378	0.360	0.340	0.304	0.154
0.30	0.389	0.372	0.350	0.315	0.157
0.40	0.401	0.378	0.362	0.327	0.159
0.50	0.412	0.394	0.375	0.340	0.164
0.60	0.439	0.414	0.391	0.356	0.169
0.70	0.468	0.439	0.411	0.377	0.176
0.80	$+\infty$	0.474	0.437	0.406	0.188
0.90	$+\infty$	$+\infty$	$+\infty$	0.447	0.210
0.95	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

Table 4.2: Mean interval width measured on the ELEC2 data set [6]. Average taken over 20 trials.

This section presents the results obtained from applying nmtCP to the subset of the ELEC2 data set [6] from Chapter 4.1.1. Due to the nature of the data set, this set of experiments showcases how the method behaves in the presence of a strong distribution shift.

In these adverse conditions, nmtCP remains valid when using the exponential or the soft cutoff weight functions. nmtCP remains almost valid in combination with the linear weight function, but loses validity for most confidence levels $1 - \alpha$ when all residuals are given the same weight. Table 4.1 shows that in all instances, nmtCP outperforms CF-RNN [18] by realizing a greater coverage rate on average. This behavior can be observed in Figure 4.4. It shows the coverage rate of the prediction regions generated by nmtCP on the test set for different confidence levels $1 - \alpha$. In addition to the results of nmtCP with different weight functions W, *Off-Line Constant* represents the off-line setting with constant weights $\tilde{w}_i$. *Off-Line Constant* is equivalent to CF-RNN [18], which does not utilize new residuals when they become available.

Table 4.2 shows that nmtCP achieves this boost in validity by increasing interval widths of the predicted region compared to CF-RNN [18]. Figure 4.5 displays the mean prediction interval width for different confidence levels $1 - \alpha$ corresponding to the coverage rates in Figure 4.4.

Both, the global coverage rate and the mean interval width are large for small values of the confidence levels $1 - \alpha$ resulting in all methods being valid for $1 - \alpha \leq 0.2$. The mean interval width increases slowly with the confidence levels $1 - \alpha$.

Tables 4.1 and 4.2 also show that a coverage rate of 1 is only achieved through infinite prediction intervals. This behavior cannot be represented in Figure 4.5.

While Tables 4.1 and 4.2 and Figures 4.4 and 4.5 provide a global view of the behavior of nmtCP with different weight functions W, the rolling coverage rate shown in Figure 4.6 shows how the coverage rate changes locally. With a window size of 50 it computes the coverage rate over the past 50 time steps. Once the distribution shift occurs, only nmtCP using the exponential weight function remains valid for a targeted coverage rate of $1 - \alpha = 0.8$. It does so by continuously producing infinite intervals. In combination with the soft cutoff weight function the method looses validity briefly, remaining approximately valid for the remaining time steps of the test set. In combination with the linear weight function and the constant weight function, the nmtCP is not able to regain validity consistently after the change point. Figure 4.6 also shows that CF-RNN [18] none of the real labels y_i are within the predicted region after the distribution shift.

4.4.2 Tétouan City

This section presents the results obtained from applying nmtCP to the subset of the Tétouan City data set [16] from Chapter 4.1.2. In contrast to the results presented in Chapter 4.4.1, this set of experiments shows how the method behaves in the presence of a slow distribution shift.

Globally all model are valid for all confidence levels $1 - \alpha$. nmtCP with the exponential weight function, the soft cutoff weight function and the linear weight function are the models with the highest coverage rate closely followed by the *Off-Line Constant* variant that is equivalent to CF-RNN [18] and the nmtCP with constant weights. Table 4.3 and Figure 4.7 show these results. They also show that the difference in coverage rate between the variants of the method is small and that their coverage rate is above the targeted coverage rate, especially towards the lower end of the confidence level $1 - \alpha$.

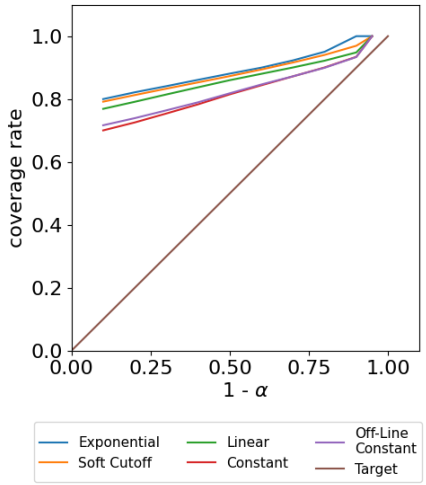

Figure 4.7: Empirical coverage rate. Average taken over 20 trials.

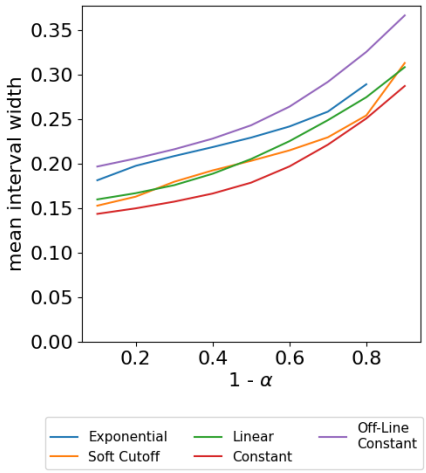

Figure 4.8: Mean interval width. Average taken over 20 trials.

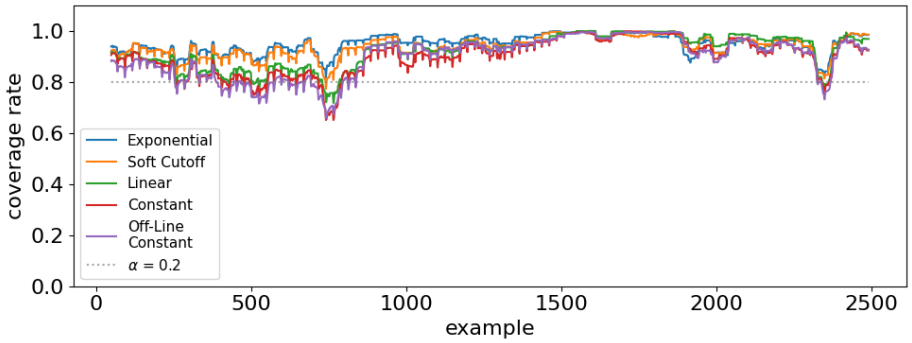

Figure 4.9: Rolling coverage rate with a window size of 50 for different weight functions. Target coverage rate $1 - \alpha = 0.8$. Average taken over 20 trials.

1-α	Exponential	Soft Cutoff	Linear	Constant	Off-Line Constant
0.10	0.800	0.792	0.769	0.700	0.717
0.20	0.822	0.813	0.791	0.726	0.739
0.30	0.841	0.833	0.815	0.754	0.764
0.40	0.861	0.853	0.837	0.783	0.790
0.50	0.881	0.873	0.860	0.815	0.818
0.60	0.900	0.894	0.880	0.844	0.846
0.70	0.923	0.916	0.901	0.872	0.872
0.80	0.950	0.940	0.922	0.901	0.900
0.90	1.000	0.969	0.948	0.934	0.934
0.95	1.000	1.000	1.000	1.000	1.000

Table 4.3: Empirical coverage rate measured on the Tétouan City data set [16]. Average taken over 20 trials.

1-α	Exponential	Soft Cutoff	Linear	Constant	Off-Line Constant
0.10	0.181	0.153	0.160	0.144	0.197
0.20	0.198	0.163	0.167	0.150	0.206
0.30	0.209	0.180	0.176	0.157	0.216
0.40	0.219	0.192	0.189	0.166	0.228
0.50	0.229	0.203	0.205	0.179	0.243
0.60	0.242	0.215	0.225	0.197	0.264
0.70	0.259	0.230	0.249	0.221	0.292
0.80	0.289	0.254	0.275	0.251	0.326
0.90	$+\infty$	0.313	0.309	0.287	0.367
0.95	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

Table 4.4: Mean interval width measured on the Tétouan City data set [16]. Average taken over 20 trials.

While in Chapter 4.4.1 models with larger coverage rate also produced larger prediction intervals, this is not the case for this set of experiments. nmtCP paired with the constant weight function has both the lowest coverage rate and produces the smallest prediction regions. However CF-RNN [18] has a similar coverage rate while producing the largest prediction regions. Table 4.4 and Figure 4.8 also indicate that for $1-\alpha = 0.95$ all of them produce infinitely large prediction regions while nmtCP with the exponential weight function already does so for $1 - \alpha = 0.9$.

The three models with the lowest coverage rate, namely nmtCP with the constant weight function, CF-RNN [18] and nmtCP with the linear weight function all loose their coverage rate temporarily as shown in Figure 4.9. For most of the time all models reach a coverage rate that is substantially above the targeted coverage rate of $1 - \alpha = 0.8$.

4.5 Discussion

The results presented in Chapter 4.4.1 show that nmtCP is able to quickly adapt to strong distribution shifts, where CF-RNN [18] loses its validity. In this situation, nmtCP retains its global validity when the right weight functions W are chosen. Chapter 4.4.2 shows that nmtCP can produce smaller, more efficient prediction regions when both methods are valid. Both experiments confirm that nmtCP is suited for the multistep-ahead multivariate time-series forecasting task and that when it does loose validity locally, it recovers quickly.

Both experiments also show that the global validity, the loss of local validity, the speed of recovery of local validity and the mean interval width all depend on the weight function W that is used. The use of the exponential weight function consistently leads to the highest coverage rates, followed by the soft cutoff weight function. With the chosen parameters for both weight functions, they place a larger weights on recent residuals while discounting older ones heavily. This allows nmtCP to quickly adapt to distribution shifts. The drawback of these weight functions is that by heavily discounting a large number of old residuals they reduce the effective sample size of past residuals [2] leading to larger steps in the weighted quantile function `Q`. This contributes to a larger number of

uninformative intervals of infinite size for high confidence levels $1 - \alpha$. The other two weight functions suffer less from this phenomenon. Discounting older residuals less quickly leads to a slower recovery after a local loss of validity which costs them their global validity in Experiment 4.4.1. Hence, the choice of the weight function and thereby the speed and severity with which to discount older residuals presents the method's user with a classic bias–variance tradeoff.

The theoretic proof presented in Chapter 3.4 guarantees the method's validity through an upper bound of the miscoverage rate by relying on the Bonferroni correction. It does, however, not provide information about the method's efficiency. Both experiments show that nmtCP is often too conservative, producing prediction regions exceeding the targeted coverage rate, especially for low confidence levels $1-\alpha$. While (almost) assumption free validity is the main appeal of the conformal prediction methods, inefficient intervals are undesirable as they underestimate the underlying model's confidence. This conservative behavior of nmtCP is, in the worst case, another factor contributing to more uninformative intervals of infinite size for high confidence levels $1 - \alpha$.

To investigate the influence of the correction method for the family-wise error rate on the method's efficiency and its conservative behavior I conducted an additional experiment that uses the subsection of the ELEC2 data set [6] described in Chapter 4.1.1. However, instead of comparing different weight functions W for the weighted quantile function Q, it compares different correction methods for the family-wise error rate. All trials use the soft cutoff weight function presented in Chapter 4.2.1. The correction methods are

- the Bonferroni correction shown in Equation (3.9), replicating the results from Chapter 4.4.1;
- the assumption that the errors occur independently in every dimension of the label space, resulting in $\alpha' = 1 - (1 - \alpha)^{T_h \cdot F_h}$, and;
- no correction at all with $\alpha' = \alpha$.

Chapter B.1 in the appendix contains the results for this experiment. Table B.1 shows that with no correction at all nmtCP is not valid, except in the extreme case where the produced interval is of infinite size and therefor uninformative. Using a correction based on the assumption that the errors occur independently in every dimension of the label space remains valid on average for all confidence levels $1 - \alpha$ and compares favorably to the Bonferroni correction by being more efficient. Figure B.1 shows that the coverage rate remains closer to the targeted coverage rate, especially for lower confidence levels $1-\alpha$ and Figure B.2 confirms that the produced intervals are smaller on average. For $1 - \alpha = 0.8$, the two methods are remarkably similar as shown in Figure B.3. Due to their similarity for high confidence levels $1 - \alpha$, the correction based on the assumption that the errors occur independently does not reduce the generation of uninformative intervals of infinite size.

In addition to being a valid method for generating prediction intervals for the multistep-ahead multivariate time-series forecasting, nmtCP also provides the user with tools to investigate the contributions of every dimension in the label space to the miscoverage rate and the average interval width. This was already hinted to in Chapter 4.3. Not only can individual dimensions be scrutinized but they can also be aggregated. Figure B.4 in the appendix is an example of this,

showing the rolling coverage rate for every feature of the results presented in Chapter 4.4.1. These detailed insights are easily obtained for nmtCP because the prediction intervals are generated and expressed independently of each other for every dimension in the label space and aggregated to a common prediction region after their generation.

Chapter 5

Conclusion

This thesis examines the task of conformal multistep-ahead multivariate time-series forecasting.

After presenting and discussing the original conformal prediction method and its subsequent evaluations, I investigate the difficulty associated with conformal multi-target regression. I then propose a method, nmtCP, that is computationally efficient and easy to implement, requiring no modifications to the underlying model. For this method, I show that it has desirable theoretical properties by proving that the miscoverage rate has an upper bound that depends on the chosen hyperparameters. Since this proof does not rely on the exchangeability of the examples in the data set or a symmetric model fitting algorithm for the underlying model, the method is suitable for time-series applications. I can therefore answer the first research question "*Can the assumptions made by current multivariate time-series prediction interval estimation methods be relaxed?*" affirmatively.

In addition, the method's architecture allows for an easy investigation of its behavior. This is possible because the metrics used to evaluate the method can be applied to each dimension in the label space individually.

The experimental results on two public data sets from the electricity domain, a subset of which has already been published in [17], serve to validate nmtCP's theoretical properties and provide insights into nmtCP's behavior. These results answer the second research question "*Can the coverage of current multivariate time-series prediction interval estimation methods be improved?*" as they show that even in adverse conditions the method retains its validity when paired with the right parameters. It recovers quickly after a strong distribution shift and is more efficient than a competing method when both are valid.

The experiments also show that the method is generally too conservative, especially for small confidence levels, producing prediction intervals that are larger than they need to be, even of infinite size in the worst case.

5.1 Future Work

Addressing the method's overly conservative behavior is the most pressing next step. Integrating the work of Messoudi et al. [11] offers a promising opportunity to do that. Messoudi et al. [11] leverage the correlations between the different di-

mensions in the label space using copulas. Instead of a hyper-rectangle, in their subsequent publication, Messoudi et al. [12] present a method that produces a multidimensional ellipsoidal uncertainty region.

While the weights w of the weighted quantile function Q offer the user a lot of flexibility to tune the presented method, this thesis gives no concrete advice on how to do that. Based on the suggestion made by Barber et al. [2] and the intuitive interpretation of the task, I assume that newer residuals should carry a higher weight. More experiments are required to offer more practical guidance to the user on how to set them.

In its current form, nmtCP only supports underlying models that produce point forecasts. I would like to explore if the advances made for quantile regression used in [2, 5] can be applied to nmtCP. This might add support for underlying quantile regression models.

Finally, to test and compare nmtCP and future methods, I would appreciate a set of synthetic benchmarks that test different scenarios such as change points in the distribution or slow distribution shifts in the data occurring for all or a subset of the features. These benchmarks should ideally come with a set of metrics that allow for an objective and comparable quantification of the method's properties, such as the time it takes for the method to recover after falling below the targeted coverage rate.

Bibliography

[1] R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1), Feb. 2021. ISSN 0090-5364. doi: 10.1214/20-AOS1965. URL https://projecteuclid.org/journals/annals-of-statistics/volume-49/issue-1/Predictive-inference-with-the-jackknife/10.1214/20-AOS1965.full.

[2] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani. Conformal prediction beyond exchangeability. *arXiv:2202.13415 [stat]*, Mar. 2022. URL http://arxiv.org/abs/2202.13415. arXiv: 2202.13415.

[3] V. Chernozhukov, K. Wüthrich, and Y. Zhu. Exact and robust conformal inference methods for predictive machine learning with dependent data. Technical report, The IFS, Mar. 2018. URL https://www.ifs.org.uk/uploads/CWP161818.pdf.

[4] S. Feldman, S. Bates, and Y. Romano. Conformalized Online Learning: Online Calibration Without a Holdout Set, May 2022. URL http://arxiv.org/abs/2205.09095. Number: arXiv:2205.09095 arXiv:2205.09095 [cs, stat].

[5] I. Gibbs and E. Candès. Adaptive Conformal Inference Under Distribution Shift. *arXiv:2106.00170 [cs, stat]*, Oct. 2021. URL http://arxiv.org/abs/2106.00170. arXiv: 2106.00170.

[6] M. B. Harries. SPLICE-2 Comparative Evaluation: Electricity Pricing, 1999.

[7] M. S. Hossain and H. Mahmood. Short-Term Photovoltaic Power Forecasting Using an LSTM Neural Network and Synthetic Weather Forecast. *IEEE Access*, 8:172524–172533, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.3024901. URL https://ieeexplore.ieee.org/document/9200614/.

[8] Y. Lin, I. Koprinska, and M. Rana. SSDNet: State Space Decomposition Neural Network for Time Series Forecasting. In *IEEE ICDM 2021 21st IEEE International Conference on Data Mining*. arXiv, Dec. 2021. URL http://arxiv.org/abs/2112.10251. Number: arXiv:2112.10251 arXiv:2112.10251 [cs].

[9] Y. Lin, I. Koprinska, and M. Rana. Temporal Convolutional Attention Neural Networks for Time Series Forecasting. In *2021 International Joint*

Conference on Neural Networks (IJCNN), pages 1–8, Shenzhen, China, July 2021. IEEE. ISBN 978-1-66543-900-8. doi: 10.1109/IJCNN52387.2021.9534351. URL https://ieeexplore.ieee.org/document/9534351/.

[10] H. Linusson, U. Norinder, H. Boström, U. Johansson, and T. Löfström. On the Calibration of Aggregated Conformal Predictors. In A. Gammerman, V. Vovk, Z. Luo, and H. Papadopoulos, editors, *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, volume 60 of *Proceedings of Machine Learning Research*, pages 154–173. PMLR, June 2017. URL https://proceedings.mlr.press/v60/linusson17a.html.

[11] S. Messoudi, S. Destercke, and S. Rousseau. Copula-based conformal prediction for multi-target regression. *Pattern Recognition*, 120:108101, Dec. 2021. ISSN 00313203. doi: 10.1016/j.patcog.2021.108101. URL https://linkinghub.elsevier.com/retrieve/pii/S0031320321002880.

[12] S. Messoudi, S. Destercke, and S. Rousseau. Ellipsoidal conformal inference for Multi-Target Regression. In U. Johansson, H. Boström, K. An Nguyen, Z. Luo, and L. Carlsson, editors, *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction with Applications*, volume 179 of *Proceedings of Machine Learning Research*, pages 294–306. PMLR, Aug. 2022. URL https://proceedings.mlr.press/v179/messoudi22a.html.

[13] R. I. Oliveira, P. Orenstein, T. Ramos, and J. V. Romano. Split Conformal Prediction for Dependent Data. *arXiv:2203.15885 [math, stat]*, Mar. 2022. URL http://arxiv.org/abs/2203.15885. arXiv: 2203.15885.

[14] N. W. Pyle. STRANGE PLANET: I HAVE ATTEMPTED SCIENCE, June 2022. URL https://twitter.com/nathanwpyle/status/1536691382332669959?s=20&t=2mCEQdg2pBZXCoVVaAuo2Q.

[15] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, A. Luccioni, T. Maharaj, E. D. Sherwin, S. K. Mukkavilli, K. P. Kording, C. Gomes, A. Y. Ng, D. Hassabis, J. C. Platt, F. Creutzig, J. Chayes, and Y. Bengio. Tackling Climate Change with Machine Learning. *arXiv:1906.05433 [cs, stat]*, Nov. 2019. URL http://arxiv.org/abs/1906.05433. arXiv: 1906.05433.

[16] A. R. Salam and A. E. Hibaoui. Comparison of Machine Learning Algorithms for the Power Consumption Prediction : - Case Study of Tetouan city –. *2018 6th International Renewable and Sustainable Energy Conference (IRSEC)*, pages 1–5, 2018.

[17] F. Schlembach, E. Smirnov, and I. Koprinska. Conformal Multistep-Ahead Multivariate Time-Series Forecasting. In U. Johansson, H. Boström, K. An Nguyen, Z. Luo, and L. Carlsson, editors, *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction with Applications*, volume 179 of *Proceedings of Machine Learning Research*, pages 316–318. PMLR, Aug. 2022. URL https://proceedings.mlr.press/v179/schlembach22a.html.

[18] K. Stankevičiūtė, A. M. Alaa, and M. van der Schaar. Conformal Time-Series Forecasting. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/312f1ba2a72318edaaa995a67835fad5-Abstract.html.

[19] P. Toccaceli. Introduction to conformal predictors. *Pattern Recognition*, 124:108507, Apr. 2022. ISSN 00313203. doi: 10.1016/j.patcog.2021.108507. URL https://linkinghub.elsevier.com/retrieve/pii/S003132032100683X.

[20] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer New York, New York, NY, 2000. ISBN 978-1-4419-3160-3 978-1-4757-3264-1. doi: 10.1007/978-1-4757-3264-1. URL http://link.springer.com/10.1007/978-1-4757-3264-1.

[21] V. N. Vapnik and S. Kotz. *Estimation of dependences based on empirical data*. Information science and statistics. Springer, New York, 2nd ed edition, 2006. ISBN 978-0-387-34239-9.

[22] V. Vovk. Transductive conformal predictors. In H. Papadopoulos, A. S. Andreou, L. Iliadis, and I. Maglogiannis, editors, *Artificial Intelligence Applications and Innovations*, volume 412, pages 348–360. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-41141-0 978-3-642-41142-7. doi: 10.1007/978-3-642-41142-7_36. URL http://link.springer.com/10.1007/978-3-642-41142-7_36. Series Title: IFIP Advances in Information and Communication Technology.

[23] V. Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74(1-2):9–28, June 2015. ISSN 1012-2443, 1573-7470. doi: 10.1007/s10472-013-9368-4. URL http://link.springer.com/10.1007/s10472-013-9368-4.

[24] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic learning in a random world*. Springer, New York, 2005. ISBN 978-0-387-00152-4 978-0-387-25061-8.

[25] C. Xu and Y. Xie. Conformal prediction interval for dynamic time-series. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11559–11569. PMLR, July 2021. URL https://proceedings.mlr.press/v139/xu21h.html.

[26] M. Zaffran, A. Dieuleveut, O. Féron, Y. Goude, and J. Josse. Adaptive Conformal Predictions for Time Series. *arXiv:2202.07282 [cs, stat]*, Feb. 2022. URL http://arxiv.org/abs/2202.07282. arXiv: 2202.07282.

Appendix A

Symbols

This chapter contains a list of the symbols used throughout this thesis with a brief description.

Data

- $z_i = (x_i, y_i) \in \mathcal{Z}$ example in example space
- Q distribution of the $z_i \in \mathcal{Z}$
- $x_i \in \mathcal{X}$ object in object space
- $y_i \in \mathcal{Y}$ label in label space
- $\hat{y}_i \in \mathcal{Y}$ predicted label in label space
- $\hat{\mathbf{y}}_i \subseteq \mathcal{Y}$ prediction interval or region for regression tasks OR prediction subset for classification tasks
- D_{train} proper training set consisting of z_i
- D_{cal} calibration set consisting of z_i
- r_i nonconformity score, Eq 2.5
- p_{z_i} p-value, Eq 2.6, Eq 2.10
- $\mathbb{R}$ real numbers
- $o_{i,j} \in \mathbb{R}$ observation at time step i of feature j

Functions

- h simple predictor, Eq 2.1
- Γ confidence predictor, Eq 2.2
- A, A_n nonconformity measure, Eq 2.3, Eq 2.4
- h_D model fitting function, fits the model to the data set D
- $\hat{\mu}$ fitted model

- err_i error indicator function function, Eq 2.8
- Q (weighted) quantile function
- P probability

Sizes and Indexes

- n Number of known examples
- n_0 Starting set size, number of known examples in the initial state
- $n+1$ Index of the object x_{n+1} for which the label is unknown
- m_u size of the test set
- m_k, m_0 size of the proper training set if the order matters or if it is split randomly, respectively, $k = 1, 2, \ldots$
- m_c size of the calibration set
- F number of features in the multivariate setting
- T number of total time steps in the time-series
- T_w number of time steps per window
- T_h number of time steps in the prediction horizon
- W weight generating function for the weighted quantile function Q

Parameters

- $\alpha \in [0, 1]$ significance level, usually 0,05
- $\epsilon \in [0, 1]$ confidence level $= 1 - \alpha$, usually 0,95
- w weights used in the weights quantile function Q in Gibbs and Candès [5] and Barber et al. [2]
- γ step size parameter in Gibbs and Candès [5]

Appendix B

Additional Experimental Results

B.1 Correction Method Comparison on ELEC2

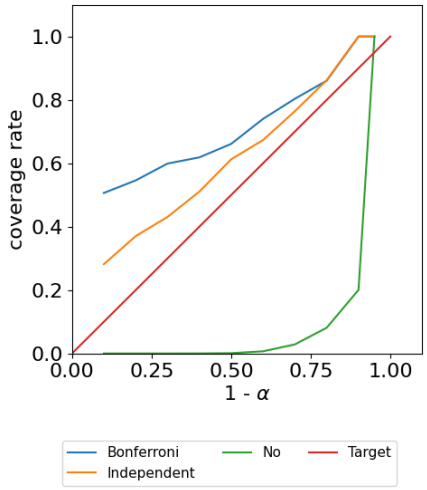

Figure B.1: Empirical coverage rate measured on the ELEC2 data set [6]. Average taken over 20 trials.

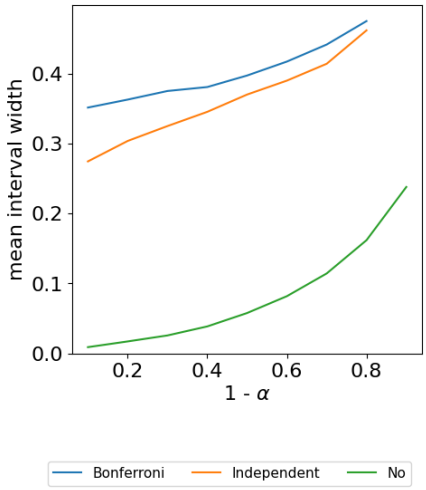

Figure B.2: Mean interval width measured on the ELEC2 data set [6]. Average taken over 20 trials.

This experiment uses the same setup as the experiment presented in Chapter 4.4.1, applying nmtCP to the subsection of the ELEC2 data set [6] presented in Chapter 4.1.1. However, instead of comparing different weight functions, the aim of this experiment is to compare different ways to correct for the family-wise error rate. Using the soft-cutoff weight function for all runs, the results presented here showcase the behavior of nmtCP using the Bonferroni correction, reproducing the results presented in Chapter 4.4.1, using a correction based on the assumption that errors occur independently of each other, and using no correction for the family-wise error rate at all. These three correction schemes

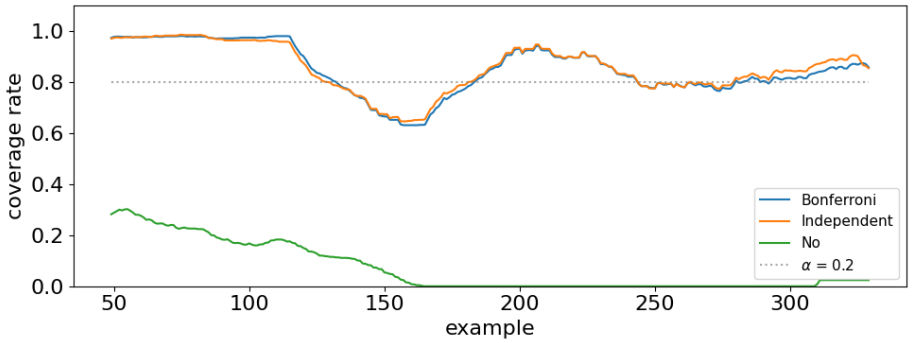

Figure B.3: Rolling coverage rate with a window size of 50 for different weight functions. Target coverage rate $1 - \alpha = 0.8$. Average taken over 20 trials.

	Coverage Rate			Mean Interval Width		
1-α	Bonferroni	Independent	No	Bonferroni	Independent	No
0.10	0.506	0.282	0.000	0.351	0.274	0.009
0.20	0.546	0.370	0.000	0.363	0.304	0.017
0.30	0.599	0.431	0.000	0.375	0.325	0.026
0.40	0.619	0.511	0.000	0.381	0.345	0.039
0.50	0.661	0.613	0.001	0.397	0.370	0.058
0.60	0.740	0.673	0.007	0.417	0.390	0.082
0.70	0.803	0.764	0.028	0.441	0.414	0.114
0.80	0.861	0.862	0.081	0.475	0.462	0.162
0.90	1.000	1.000	0.201	$+\infty$	$+\infty$	0.238
0.95	1.000	1.000	1.000	$+\infty$	$+\infty$	$+\infty$

Table B.1: Empirical coverage rate and mean interval width measured on the ELEC2 data set [6]. Average taken over 20 trials.

are labeled as *Bonferroni*, *Independent* and *No*, respectively, in Figures B.1, B.2 and B.3 as well as in Table B.1.

B.2 Weight Function Comparison on ELEC2

Additional results related to the ones presented in Chapter 4.4.1.

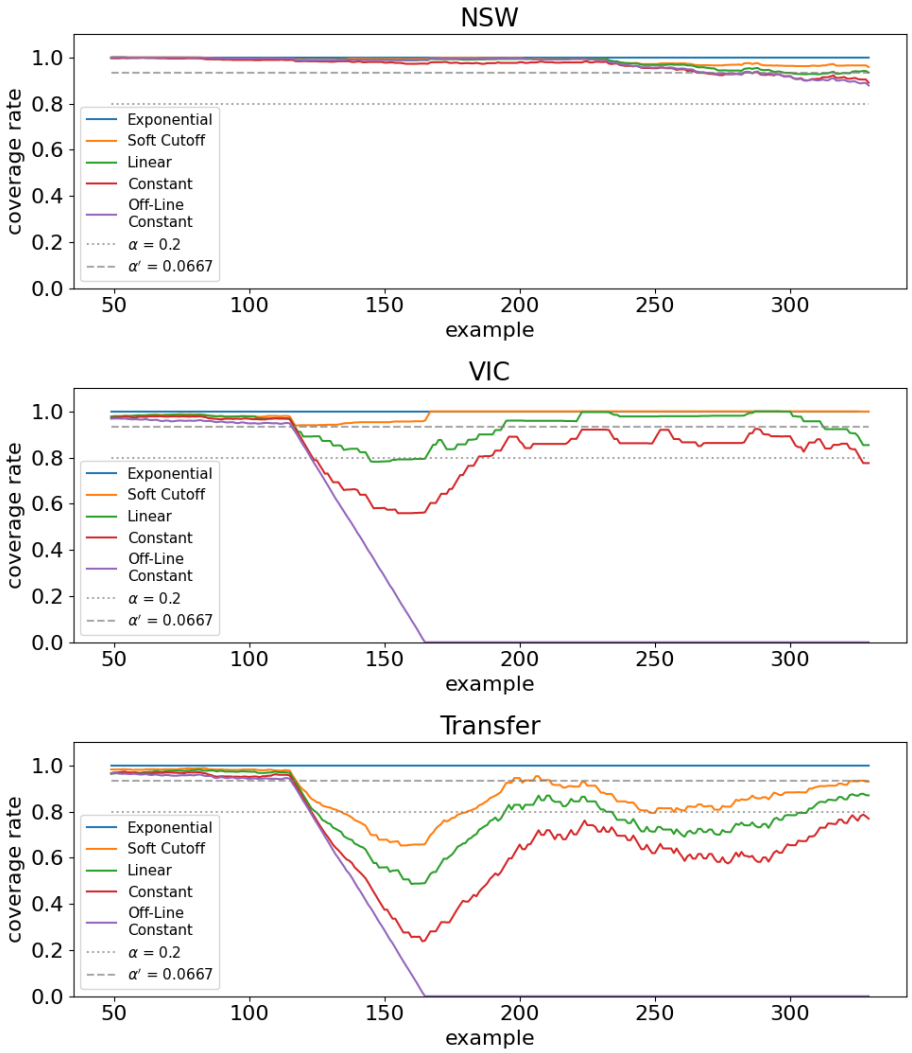

Figure B.4: Rolling coverage rate with a window size of 50 for all features in the data set and different weight functions. Target coverage rate $1 - \alpha = 0.8$. Average taken over 20 trials.