Proceedings of Machine Learning Research 266:1–26, 2025 Conformal and Probabilistic Prediction with Applications

Conformal Predictive Decision Making: A Comparative Study with Bayesian and Point-Predictive Methods

Simon Lanngren SIMLANN@CHALMERS.SE
Chalmers University of Technology and Algorithma AB, Sweden

Martin Toremark TOREMARK@CHALMERS.SE
Chalmers University of Technology and Algorithma AB, Sweden

Johan Hallberg Szabadváry JOHAN.HALLBERG.SZABADVARY@MATH.SU.SE
Jönköping University, Stockholm University, and Algorithma AB, Sweden

Editor: Khuong An Nguyen, Zhiyuan Luo, Harris Papadopoulos, Tuwe Löfström, Lars Carlsson and Henrik Boström

Abstract

In many real-world settings, machine learning predictions serve as intermediate outputs used to inform decision-making. However, quantifying and accounting for uncertainty in these decisions remains a fundamental challenge. Conformal Predictive Decision Making is a framework for decision-making under uncertainty that leverages Conformal Predictive Distributions to optimize outcomes over a specified utility function. In this work, we evaluate Conformal Predictive Decision Making on synthetic datasets in both online and inductive settings, and compare its performance to two alternative approaches: Bayesian Decision Theory and Point Predictive Decision Making.

Online Conformal Predictive Decision Making showed signs of greater robustness than Bayesian Decision Theory and Point Predictive Decision Making in scenarios involving noisy data and skewed utility functions, suggesting it may be a suitable option in more complex settings. However, it generally performed worse than the two alternative methods. In contrast, inductive Conformal Predictive Decision Making consistently outperformed the alternatives. This, combined with its computational advantages, makes it a promising approach for larger real-world decision-making applications where well-calibrated uncertainty quantification is needed for robustness.

Keywords: Conformal Prediction, Conformal Predictive Decision Making, Bayesian Decision Theory, Decision Making

1. Introduction

Machine learning (ML) has advanced rapidly in recent decades, resulting in the development of numerous highly effective algorithms. Although much of the field has focused on designing learning algorithms that improve predictive accuracy, predictions alone are often insufficient. In many real-world applications, decisions must ultimately be made and predictions often serve as input to the decision-making process (Agrawal et al., 2019). We refer to this integration of prediction and action as predictive decision-making. In this setting, the goal extends beyond generating accurate predictions: it involves using predictions to select actions that lead to the most favorable outcomes. The preferences of possible outcomes are typically formalized using utility functions, allowing decisions to be guided by the principle of expected utility maximization (von Neumann and Morgenstern, 1953).

A central challenge in predictive decision-making is handling uncertainty. Point estimates from many standard ML models may not be sufficiently reliable for critical decisions (Nemani et al., 2023). This motivates the need for uncertainty quantification to support more informed, robust, and risk-aware decision-making.

A powerful and widely used approach for quantifying uncertainty in ML is probabilistic prediction using Bayesian methods, which provide a principled way for refining probability estimates over time (Murphy, 2023). Although Bayesian probability estimates reflect uncertainty, the methods do not guarantee validity. In other words, the posterior distribution is not necessarily well calibrated in taking the true underlying uncertainty into account. In contrast, the conformal prediction (CP) framework provides valid prediction sets under the assumption of exchangeability in the data (Vovk et al., 2005). However, the set-valued outputs of conformal predictors lack the expressiveness of full probability distributions, making their integration into predictive decision-making less straightforward. To address this limitation, Vovk et al. (2017) extended the CP framework to probabilistic regression using conformal predictive systems (CPSs), which produce conformal predictive distributions (CPDs). Importantly, these predictive distributions are valid. However, they do not come with any efficiency guarantees, that is, how narrow the distribution is in relation to the true distribution.

Access to CPDs allowed Vovk and Bendtsen (2018) to apply the CP framework to decision-making and develop a coherent, systematic theory for conformal predictive decision-making (CPDM). While extensive research has been conducted in Bayesian decision theory (BDT), CPDM remains largely unexplored. Since its introduction, no further studies have examined it, particularly compared to alternative methods such as BDT. In their original paper, the authors applied their theory only to the simple Mushroom dataset from the UCI repository (UCI, 1981). Although their results were promising, they concluded that further research is needed to evaluate CPDM's practical usefulness.

This study aims to expand the existing literature on CPDM by evaluating its effectiveness relative to BDT and point predictive decision making (PPDM), where we simply rely on point estimates to make decisions, in a binary decision-making setting. The comparison seeks to clarify the strengths and limitations of CPDM compared to the other methods. The evaluation will be conducted on synthetic datasets for which we will vary the availability of data, noise levels, and utility functions. In addition, the approaches will be tested both in an online and inductive setting. This analysis will help identify their relative performance in different contexts.

2. Preliminaries

In this section, we present the theoretical background required to understand the methods employed in this study. We first provide an in-depth overview of CPSs and their inductive variant, followed by a detailed description of predictive decision-making and CPDM.

2.1. Conformal Predictive Systems

The CP framework operates in a supervised learning setting, where we assume access to a set of objects $\boldsymbol{x}_i \in X$ that are labeled as $y_i \in Y$, where X and Y are fixed, non-empty measurable spaces, referred to as the *object space* and *label space*, respectively. For

notational convenience, we also define the Cartesian product $Z := X \times Y$, which is called the *example space*, and let $z_i := (\boldsymbol{x}_i, y_i)$.

A CPS is an extension of CP to the probabilistic setting, allowing the derivation of CPDs that are valid under the assumption of exchangeability. Importantly, CPDs encode more information than standard CP intervals; in particular, a single CPD can produce multiple CP prediction sets. However, one limitation of CPDs is that they can only be created in the context of probabilistic regression, where $Y = \mathbb{R}$.

We begin our formal description of CPSs by introducing the notion of a *conformity measure*. Following Vovk et al. (2022), a conformity measure is a measurable function:

$$A : Z^{(*)} \times Z \to \overline{\mathbb{R}},$$

where $Z^{(*)}$ denotes the set of all finite *bags* of elements from Z, and $\overline{\mathbb{R}}$ represents the extended real numbers. A bag is an unordered collection that may contain repeated elements, unlike a set. For any bag of examples $\sigma := \{\!|z_1, \ldots, z_{n-1}|\!\} \in Z^{(*)}$, and any example $z_n \in Z$, the conformity measure A assigns a numerical score that indicates how well z_n conforms to σ. In our setting, we are only interested in values $A(\sigma, z_n)$ where $z_n \in \sigma$. Since bags are unordered, the conformity measure A is invariant under permutations of the elements in σ.

Given a conformity measure A, a *smoothed conformal transducer* is a function $Q : Z^n \times [0,1] \to [0,1]$ defined as:

$$\begin{aligned} Q(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), \tau) := \frac{1}{n} \left|\{i = 1, \ldots, n : \alpha_i^y < \alpha^y\}\right| \\ + \frac{\tau}{n} \left|\{i = 1, \ldots, n : \alpha_i^y = \alpha^y\}\right| \end{aligned}$$

where $z_1, \ldots, z_{n-1} \in Z^{n-1}$ is a training set, $\boldsymbol{x}_n \in X$ is a test object, $\tau \in [0,1]$ is a random number independent of everything else, and distributed uniformly on $[0,1]$. The numbers α_i^y and α^y are the *conformity scores* defined by:

$$\begin{aligned} \alpha_i^y &:= A(\sigma, z_i), \quad i = 1, \ldots, n-1, \\ \alpha^y &:= A(\sigma, (\boldsymbol{x}_n, y)), \\ \sigma &:= \{\!|z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y)|\!\}, \end{aligned}$$

for each $y \in \mathbb{R}$ (Vovk and Bendtsen, 2018; Vovk et al., 2022).

Vovk et al. (2022) defines a function $Q : Z^n \times [0,1] \to [0,1]$ as a *randomized predictive system* (RPS) if it satisfies the following two conditions:

C1 The function $Q(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), \tau)$ is monotonically increasing in both $y \in \mathbb{R}$ and $\tau \in [0,1]$, for all $(z_1, \ldots, z_{n-1}) \in Z^{n-1}$ and all $\boldsymbol{x}_n \in X$.

C2 The function $Q(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), \tau)$ satisfies the following limits:

$$\begin{aligned} \lim_{y \to -\infty} Q(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), 0) &= 0, \\ \lim_{y \to \infty} Q(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), 1) &= 1, \end{aligned}$$

for all $(z_1, \ldots, z_{n-1}) \in Z^{n-1}$ and all $\boldsymbol{x}_n \in X$.

A function that is both a smoothed conformal transducer, determined by some conformity measure, and a randomized predictive system is called a *conformal predictive system* (CPS) (Vovk et al., 2022). Given a training set $(z_1, \ldots, z_{n-1}) \in Z^{n-1}$ and test object $\boldsymbol{x}_n \in X$, a CPS Q outputs the function:

$$Q_n : (y, \tau) \in \mathbb{R} \times [0,1] \mapsto Q(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), \tau),$$

which is known as a *conformal predictive distribution* (CPD), a cumulative distribution function constructed from p-values. Under conditions **C1** and **C2**, Q_n is a monotonically increasing function in y that satisfies:

$$\lim_{y \to -\infty} Q_n(y) = 0 \quad \text{and} \quad \lim_{y \to \infty} Q_n(y) = 1,$$

ensuring that Q_n behaves like a proper cumulative distribution function, without requiring it to be right-continuous (Vovk and Bendtsen, 2018; Vovk et al., 2022). Vovk et al. (2022) defined the *thickness* of a CPD Q_n as the infimum of all $\epsilon \geq 0$ such that, for all but finitely many $y \in \mathbb{R}$, the set:

$$\{Q_n(y, \tau) : \tau \in [0,1]\}$$

has diameter at most ϵ; that is,

$$Q_n(y, 1) - Q_n(y, 0) \leq \epsilon.$$

The finitely many y-values where this condition fails are called *exception points*, and their total number is refereed to as the *exception size* of Q_n.

Vovk et al. (2022) define a smoothed conformal transducer Q to be *exactly valid* if the p-values $Q(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), \tau)$ are uniformly distributed over $[0,1]$, assuming that the examples $z_1, \ldots, z_{n-1}$ are generated by an exchangeable probability distribution on Z^{n-1}, and that $\tau \sim \mathcal{U}[0,1]$ is sampled independently. Since every CPS is a smoothed conformal transducer, it follows that all CPSs are exactly valid. Randomization is required to achieve exact validity; without randomization, only conservative validity is guaranteed (see Vovk et al. (2022) for details). The value of τ does not substantially affect the p-values.

2.1.1. Inductive Conformal Predictive Systems

In online settings, CPSs can be computationally intensive, particularly when dealing with large datasets. To address this, Vovk et al. (2022) introduced *inductive conformal predictive systems* (ICPSs). This enables the use of ICPSs within the CPDM framework. As is typical in the inductive setting, the training set $z_1, \ldots, z_{l-1}$ is split into a proper training set $z_1, \ldots, z_m$ and a calibration set $z_{m+1}, \ldots, z_{l-1}$.

An *inductive conformity measure* is defined as a measurable function:

$$A : Z^* \times Z \to \mathbb{R},$$

where Z^* is the set of all finite sequences of elements of Z, that is invariant to permutations in its first argument. Given an inductive conformity measure A, an *inductive smoothed conformal transducer* $Q : Z^l \times [0,1] \to [0,1]$ is defined as:

$$Q(z_1,\dots,z_{l-1},(\boldsymbol{x}_l,y),\tau) := \frac{1}{l-m}\left|\{i=m+1,\dots,l : \alpha_i < \alpha^y\}\right| + \frac{\tau}{l-m}\left|\{i=m+1,\dots,l : \alpha_i = \alpha^y\}\right| + \frac{\tau}{l-m}$$

where $z_{m+1},\dots,z_{n-1} \in Z^{l-m-1}$ is a calibration set, $\boldsymbol{x}_l \in X$ is a test object, $\tau \in [0,1]$ is a smoothing parameter, and α_i and α^y are the conformity scores defined by:

$$\alpha_i^y := A(z_1,\dots,z_m,z_i), \quad i = m+1,\dots,l-1,$$
$$\alpha^y := A(z_1,\dots,z_m,(\boldsymbol{x}_l,y)),$$

for each $y \in \mathbb{R}$.

Similar to a CPS, an ICPS is defined as a function that is an inductive smoothed conformal transducer, determined by some inductive conformity measure, and an RPS. Validity is automatically satisfied for an inductive smoothed conformal transducer, and means the p-values $Q(z_1,\dots,z_{l-1},(\boldsymbol{x}_l,y),\tau)$ are uniformly distributed over $[0,1]$, provided the examples $z_1,\dots,z_{l-1},(\boldsymbol{x}_l,y)$ are generated by an exchangeable probability distribution on Z^l and $\tau \sim \mathcal{U}[0,1]$ is generated independently of them.

2.2. Predictive Decision Making

Assume we are given a training set $z_1,\dots,z_{n-1}$ of examples, where $z_i = (\boldsymbol{x}_i, y_i) \in Z$, and a finite set of available decisions D. We consider the decision problem of selecting the best decision $d \in D$ for a new object $\boldsymbol{x}_n \in X$, based on the information contained in the training set. Moreover, to guide the decision-making process, we assume the existence of a measurable *utility function* $U : Y \times D \to \mathbb{R}$, where $U(y,d)$ represents the von Neumann-Morgenstern utility associated with the decision $d \in D$ when the true label is y (von Neumann and Morgenstern, 1953).

The decision-making problem under consideration can be formalized using the concept of a *predictive decision-making system* (PDMS). Following Vovk and Bendtsen (2018), a PDMS is defined as a measurable function:

$$F : Z^{n-1} \times X \to D,$$

which recommends a decision d for a new test object $\boldsymbol{x}_n$ based on the training set $z_1,\dots,z_{n-1}$. A PDMS may also be *randomized*, in which case it is defined as a measurable function $F : Z^{n-1} \times X \times [0,1] \to D$ that takes an additional random number $\tau \in [0,1]$ as input. The objective is to find a PDMS F that maximizes the expected utility; thus, the optimal PDMS F^* is given by:

$$F^* = \arg\max_F \mathbb{E}\big[U(y, F(z_1,\dots,z_{n-1},\boldsymbol{x}_n,\tau))\big],$$

where the expectation is taken with respect to the joint distribution (Murphy, 2023; Vovk and Bendtsen, 2018).

In Bayesian decision theory, upon observing a test object $\boldsymbol{x}_n \in X$, the training set $z_1,\dots,z_{n-1}$ and the test object $\boldsymbol{x}_n$ are treated as fixed while the corresponding label y of

$\boldsymbol{x}_n$ is modeled as a random variable distributed according to the conditional probability distribution $P(y \mid \boldsymbol{x}_n)$, which is assumed to exist (Murphy, 2023). The expected utility for selecting the decision $d \in D$ given $\boldsymbol{x}_n$ is defined as:

$$\mathbb{E}_{P(y|\boldsymbol{x}_n)}\big[U(y, F(z_1, \ldots, z_{n-1}, \boldsymbol{x}_n, \tau))\big] = \int U(y, F(z_1, \ldots, z_{n-1}, \boldsymbol{x}_n, \tau)) \times P(y \mid \boldsymbol{x}_n)\, dy$$

and the optimal decision $d^*(\boldsymbol{x}_n)$ for a test object $\boldsymbol{x}_n$ is given by:

$$d^*(\boldsymbol{x}_n) := \arg\max_{d \in D} \mathbb{E}_{P(y|\boldsymbol{x}_n)}\big[U(y, F(z_1, \ldots, z_{n-1}, \boldsymbol{x}_n, \tau))\big],$$

where $d = F(z_1, \ldots, z_{n-1}, \boldsymbol{x}_n, \tau)$.

Vovk and Bendtsen (2018) used *regret* to measure how much worse a PDMS performs compared to optimal decisions. Given a training set $z_1, \ldots, z_{n-1}$, and an object $\boldsymbol{x}_n$, the regret of a PDMS F under a probability distribution P on Z is defined as:

$$R_F(z_1, \ldots, z_{n-1}, \boldsymbol{x}_n, \tau) := \max_{d \in D} \int U(y, d)\, P(dy \mid \boldsymbol{x}_n) - \int U\big(y, F(z_1, \ldots, z_{n-1}, \boldsymbol{x}_n, \tau)\big)\, P(dy \mid \boldsymbol{x}_n).$$

Of particular interest are PDMSs that show small regret, as well as those that are *asymptotically efficient*, meaning that $R_F(z_1, \ldots, z_{n-1}, \boldsymbol{x}_n, \tau) \to 0$ as $n \to \infty$.

2.2.1. Conformal Predictive Decision Making

In developing CPDM, Vovk and Bendtsen (2018) adopted the Bayesian approach for statistical decision-making. However, instead of using Bayesian models to produce posterior distributions, they used CPSs to derive CPDs. Specifically, they focused on CPDs with thickness $1/n$ and exception size at most $n-1$, meaning Q_n changes by at most $1/n$ as τ varies over $[0, 1]$, apart from exception points.

Vovk and Bendtsen (2018) defined the integral:

$$\int U(y, d) Q_n(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), \tau)\, dy := \sum_{y : \Delta Q_n(y) \neq 0} U(y, d) \Delta Q_n(y), \tag{1}$$

for any utility function $U : \mathbb{R} \times D \to \mathbb{R}$, where $\Delta Q_n(y) := Q_n(y+) - Q_n(y-)$ denotes the increase in the value of Q_n at the point y. They showed that any CPD Q_n is a piecewise constant function with at most $2(n-1)$ discontinuities, ensuring the sum in (1) is finite. To ensure that different integrals of the type (1) are comparable, they also strengthened the condition **C2** by requiring that the function $Q(z_1, \ldots, z_{n-1}, (\boldsymbol{x}_n, y), \tau)$ also satisfies the following limits:

$$\lim_{y\to-\infty} Q(z_1,\ldots,z_{n-1},(\boldsymbol{x}_n,y),1) = \frac{1}{n},$$
$$\lim_{y\to\infty} Q(z_1,\ldots,z_{n-1},(\boldsymbol{x}_n,y),0) = \frac{n-1}{n},$$

for all $(z_1,\ldots,z_{n-1}) \in Z^{n-1}$ and all $\boldsymbol{x}_n \in X$. These limits make the total mass of Q_n equal to $(n-1)/n < 1$. Therefore, the resulting integrals are not expectations in the strict sense. However, because the loss in total mass is consistent across different integrals, they remain comparable. Moreover, these additional limits are often satisfied by many conformity measures, for instance, the standard regression conformity measure $A(z_1,\ldots,z_{n-1},(\boldsymbol{x}_n,y)) := y - \hat{y}$, where $\hat{y}$ is the point prediction for y.

3. Method

In this section, we describe the experimental setup, including the datasets, models, utility functions, and evaluation metrics. All the code for the experiments is available on the GitHub page[1] associated with this article.

3.1. Data and Preproccessing

We generated synthetic datasets using a linear model generator and a set of nonlinear benchmark functions originally proposed by Friedman (1991), all of which are available via `scikit-learn` (Pedregosa et al., 2011). This setup supports evaluation in both linear contexts and more complex scenarios involving nonlinearities and feature interactions, thus capturing a broad spectrum of real-world modeling challenges. A detailed description of the data-generating procedures is provided in the list below, where x_i represents the i:th feature, y the target variable, and ϵ the noise.

1. **Linear Dataset (`make_regression`)**

$$x_i \sim \mathcal{N}(0,1), \quad i = 1,\ldots,5$$
$$y = \boldsymbol{w}^\top \boldsymbol{x} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,\sigma^2)$$

2. **Friedman #1 Dataset (`make_friedman1`)**

$$x_i \sim \mathcal{U}(0,1), \quad i = 1,\ldots,5$$
$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,\sigma^2)$$

3. **Friedman #2 Dataset (`make_friedman2`)**

$$x_1 \sim \mathcal{U}(0,100), \quad x_2 \sim \mathcal{U}(40\pi, 560\pi), \quad x_3 \sim \mathcal{U}(0,1), \quad x_4 \sim \mathcal{U}(1,11)$$
$$y = \left(x_1^2 + \left(x_2 x_3 - \frac{1}{x_2 x_4}\right)^2\right)^{1/2} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,\sigma^2)$$

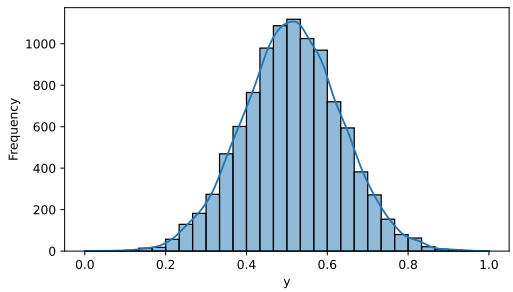
1. https://github.com/simonlanngren/Conformal-Predictive-Decision-Making

4. **Friedman #3 Dataset** (`make_friedman3`)

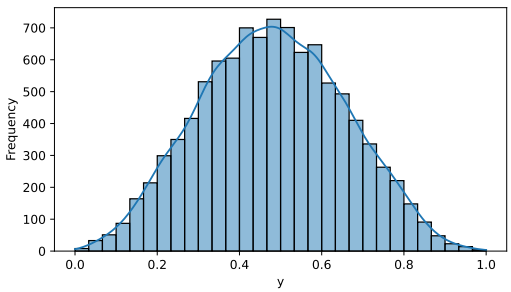
$$y = \arctan\left(\frac{x_2 x_3 - \frac{1}{x_2 x_4}}{x_1}\right) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

The input features follow the same distributions as in Friedman #2.

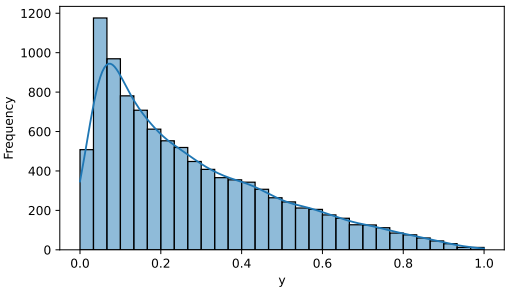
An initial pool of 10,000 data points was generated for each data-generating process. The target variable in each dataset was scaled to the $[0, 1]$ range using min-max normalization. The target distributions after scaling are shown in Figure 1. Bootstrap samples were then drawn from these pools to construct the datasets used in each experimental run. A total of 100 bootstrap runs were performed for each inductive experiment, while the online experiments were carried out using 50 runs to maintain computational feasibility.



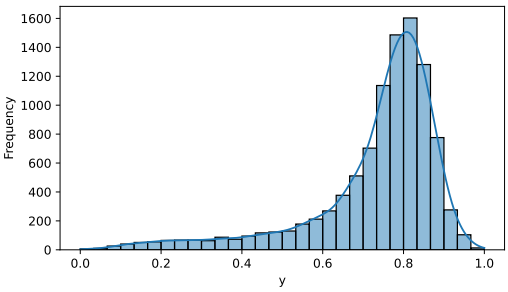
(a) Linear Dataset



(b) Friedman #1 Dataset



(c) Friedman #2 Dataset



(d) Friedman #3 Dataset

Figure 1: The target distributions of the four datasets used in this study.

Setting	Train	Calibration	Test
Standard Data Availability			
Online	25	—	75
Inductive	160	40	50
Limited Data Availability			
Online	7	—	43
Inductive	64	16	20

Table 1: An overview of data splits used in the online and inductive settings under different scenarios.

Setting	Linear	Friedman #1	Friedman #2	Friedman #3
Standard Noise	0.1	0.1	0.1	0.1
Noisy Data	10	5	100	0.5

Table 2: The noise levels σ^2 used across datasets under the standard and noisy setting.

The number of examples in each bootstrap sample varied between two different scenarios: a standard setting and a reduced-data setting. Each dataset, regardless of its size, was split in the same way into training and test sets. In the inductive setting, the training set was further divided into a proper training set and a calibration set, which was used to calibrate the ICPSs. The two data settings are summarized in Table 1. Additionally, to evaluate robustness to noise in the data, the noise level σ^2 was varied across two settings, as detailed in Table 2.

After data generation and bootstrap sampling for each run, an additional preprocessing step was performed by standardizing all the features. To avoid data leakage in the inductive setting, the scaler was fitted only on the proper training set, whereas in the online setting, it was fitted on the entire training set in each iteration.

3.2. Predictive Decision-Making Systems

This section outlines the PDMSs used for each of the approaches under consideration, in both the online and the inductive setting. The online PDMSs involving CPSs were implemented using the newly developed `online-cp` package (Szabadváry et al., 2025). The inductive PDMSs involving ICPSs, on the other hand, were implemented using the `crepes` package developed by Boström (2024), with the underlying ML-models implemented using `scikit-learn` (Pedregosa et al., 2011).

Vovk and Bendtsen (2018) provided a PDMS for CPDM, which they proved to be asymptotically efficient. Their approach is illustrated in Algorithm 1.

Algorithm 1: CPDM

Input: a training set $(\boldsymbol{x}_i, y_i) \in Z,\ i = 1, \ldots, n-1$, a test object $\boldsymbol{x}_n \in X$, a set of decisions D, and a utility function $U(y, d)$

for $d \in D$ **do**

  Create a new training set $(\boldsymbol{x}_i, U(y_i, d)),\ i = 1, \ldots, n-1$

  Find the CPD Q_n^d for this new training set

  Compute the expected utility of d as $\int uQ_n^d(du)$

end

Output: $d \in D$ with the highest expected utility

Transforming the labels in the training set using a utility function, as shown in Algorithm 1, is not standard practice in BDT. Instead, the probability distribution models the uncertainty in the label of the test object, while the utility function provides a deterministic mapping from each possible outcome to the decision-maker's subjective valuation of it. In other words, the randomness resides in the label, and the utility function does not introduce any additional randomness. Hence, in BDT, the predictive distribution is based on the original training set, as outlined in Algorithm 2. In this setting, the predictive distribution P_n is the posterior distribution obtained from a Bayesian model.

Algorithm 2: BDT

Input: a training set $(\boldsymbol{x}_i, y_i) \in Z,\ i = 1, \ldots, n-1$, a test object $\boldsymbol{x}_n \in X$, a set of decisions D, and a utility function $U(y, d)$

Find the posterior distribution P_n for the test object $\boldsymbol{x}_n$ given the training set

for $d \in D$ **do**

  Compute the expected utility of d as $\int U(y, d)P_n(dy)$

end

Output: $d \in D$ with the highest expected utility

An inductive version of Algorithms 1 and 2 can be easily formulated. These are presented in Algorithm 3 for CPDM and Algorithm 4 for BDT. It should be noted that, in contrast to Algorithm 1, the labels are not transformed into utilities in Algorithm 3 in order to remain consistent with the standard approach of BDT.

Algorithm 3: Inductive CPDM

Input: a proper training set $(\boldsymbol{x}_i, y_i) \in Z,\ i = 1, \ldots, m$, a calibration set $(\boldsymbol{x}_i, y_i) \in Z,\ i = m+1, \ldots, l-1$, a test object $\boldsymbol{x}_l \in X$, a set of decisions D, and a utility function $U(y, d)$

Train the ML model on the proper training set

Find the predictive distribution Q_l for the test object $\boldsymbol{x}_l$ given the calibration set

for $d \in D$ **do**

  Compute the expected utility of d as $\int U(y, d)Q_l(dy)$

end

Output: $d \in D$ with the highest expected utility

Algorithm 4: Inductive BDT

Input: a training set $(\boldsymbol{x}_i, y_i) \in Z,\ i = 1, \ldots, l-1$, a test object $\boldsymbol{x}_l \in X$, a set of decisions D, and a utility function $U(y, d)$

Find the posterior distribution P_l for the test object $\boldsymbol{x}_l$ given the training set

for $d \in D$ **do**

  Compute the expected utility of d as $\int U(y, d) P_l(dy)$

end

Output: $d \in D$ with the highest expected utility

In both the online and inductive settings, the PPDM methods are obtained by using the point estimates provided by the ML model, rather than a predictive distribution. Hence, we do not compute expected utility and instead use a threshold $t = 0.5$ to decide which decision to make: $d = 1$ if $t > 0.5$, $d = 0$ otherwise. We do not include explicit algorithms for brevity, since they are trivial.

3.3. Models and Model Selection

The models considered in this study, along with their tunable hyperparameters and their corresponding search spaces or optimization strategies, are summarized in Table 3. To maintain comparability and consistency for the kernel-based models, they all used the *radial basis function* (RBF) kernel:

$$\mathcal{K}(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right).$$

In online CPDM, conformalized variants of the corresponding traditional ML models were employed. Specifically, the *Least Squares Prediction Machine* (LSPM) with an added regularization parameter, the *Nearest Neighbors Prediction Machine* (NNPM), and the *Kernel Ridge Regression Prediction Machine* (KRRPM) were used for *Ridge Regression* (RR), *k-Nearest Neighbors Regression* (KNN), and *Kernel Ridge Regression* (KRR), respectively. The complete details of the conformalized models are provided by Vovk et al. (2022).

Model	Tunable Parameters	Search Space / Bounds
Non-bayesian models (for CPDM and PPDM)		
KNN/NNPM	k (number of neighbors)	{1, 3, 5, 7, 10, 15, 20}
RR/LSPM	γ (regularization)	{0.01, 0.1, 1.0, 10.0, 100.0}
KRR/KRRPM	γ (regularization)	{0.001, 0.01, 0.1, 1.0}
	ℓ (length scale)	{1, 10, 100}
Bayesian Models (for BDT)		
BRR	Gamma distribution priors	
	β_1, β_2 (shape, inverse scale)	Internally optimized
	λ_1, λ_2 (shape, inverse scale)	Internally optimized
GPR	ℓ (length scale)	$[10^{-10}, 10^{10}]$

Table 3: An overview of models, their hyperparameters, and search spaces/bounds.

Regarding the model selection, the non-Bayesian models were tuned using grid search and 5-fold cross-validation, while the Bayesian models were tuned automatically during fitting. In the inductive setting for CPDM, hyperparameter tuning was performed once on the proper training set before the calibration procedure. However, in the online setting, hyperparameters were re-tuned at each iteration to allow the models to adapt to new data dynamically over time.

On the Linear dataset, the parametric models RR (with LSPM for online CPDM) and *Bayesian Ridge Regression* (BRR) were used as underlying models for the different PDMSs. However, to better capture the inherent nonlinearities of the Friedman datasets (#1, #2, and #3), the non-parametric methods KNN (with NNPM for online CPDM), KRR (with KRRPM for online CPDM), and *Gaussian Process Regression* (GPR) were employed instead.

3.4. Utility Functions

Since the study investigates binary decision-making, it is most intuitive to formulate the utility functions by assigning values to the components of a confusion matrix. We customized the utility function for each dataset based on its target distribution. Moreover, we defined two different utility scenarios in each case to explore how varying the penalty for misclassifications affects model performance. The *standard scenario* served as the default across datasets, while the *skewed scenario* aimed to simulate a more risk-sensitive setting where the penalty for minority-class misclassifications was increased. All utility scenarios are detailed in Table 4, and for all experiments, a threshold of $t = 0.5$ was used to convert the values of the predictive distributions into class labels.

	Linear / Friedman #1		**Friedman #2**		**Friedman #3**	
Actual \Predicted	**True**	**False**	**True**	**False**	**True**	**False**
Standard						
True	1	-5	2	-10	1	-5
False	-5	1	-5	1	-10	2
Skewed						
True	1	-10	2	-10	1	-2
False	-2	1	-2	1	-10	2

Table 4: Utility matrices for all datasets under two different penalty scenarios.

3.5. Evaluation Metrics

In the online learning setting, we assessed the performance of each model using cumulative regret. The cumulative regret R_c is defined as:

$$R_c = \sum_{i=1}^{n} \left(U(y_i, d_i^*) - U(y_i, d_i) \right),$$

where n is the number of test examples, $U(y_i, d_i^*)$ is the utility obtained by the optimal decision d_i^* for the true value y_i, and $U(y_i, d_i)$ is the utility obtained by the decision d_i selected by the model. The metric quantifies the total utility lost over time due to suboptimal choices, effectively measuring how well the system improves its decision-making over time.

In the inductive learning setting, performance was instead assessed using average utility over test cases, since the systems do not improve their policy over time. The average utility $\bar{U}$ is defined as:

$$\bar{U} = \frac{1}{n} \sum_{i=1}^{n} U(y_i, d_i).$$

3.6. Experimental Setup and Configurations

This study adopted a comparative experimental design to systematically evaluate the performance of CPDM against BDT and PPDM under varying experimental conditions. A series of controlled experiments was conducted using various combinations of data availability, noise level, and utility function skewness scenarios outlined in the previous subsections. Table 5 summarizes these experimental setups. It should be noted that the different dimensions may vary between inductive and online learning settings, as described previously.

Configuration	Data	Noise Level	Utility Function
#1	Standard	Standard	Standard
#2	Standard	Standard	Skewed
#3	Standard	Noisy	Standard
#4	Standard	Noisy	Skewed
#5	Limited	Standard	Standard
#6	Limited	Standard	Skewed
#7	Limited	Noisy	Standard
#8	Limited	Noisy	Skewed

Table 5: Overview of the experimental configurations with varying data availability, noise levels, and utility functions.

3.7. Method discussion

Although our methodology relies on CPSs to obtain predictive distributions for CPDM, it is important to note that the underlying problem could, in principle, also be modeled using Venn predictors (Vovk et al., 2003). These predictors are designed for probabilistic classification within the CP framework. To apply them, one would need to discretize the regression targets into classes at the outset and then generate discrete predictive distributions accordingly.

As described by Vovk et al. (2022), Venn predictors do not produce a single definitive predictive distribution for each example, similar to CPSs. However, instead of generating an interval, they output one probability distribution per class, making them multi-probabilistic

predictors. Venn predictors also produce predictive distributions that are valid in an important sense: specifically, a prediction is considered valid if any of the individual class-wise distributions is valid.

Venn predictors are defined using Venn taxonomies, and for each taxonomy, one can define a corresponding Venn predictor (Vovk et al., 2022). However, these taxonomies are often difficult to specify in practice. Nonetheless, Venn-Abers predictors can mitigate this by leveraging isotonic regression to derive practical taxonomies for binary classification problems (Vovk and Petej, 2014). Therefore, Venn-Abers predictors could be used to produce discrete, valid predictive distributions instead of CPSs for our problem setting.

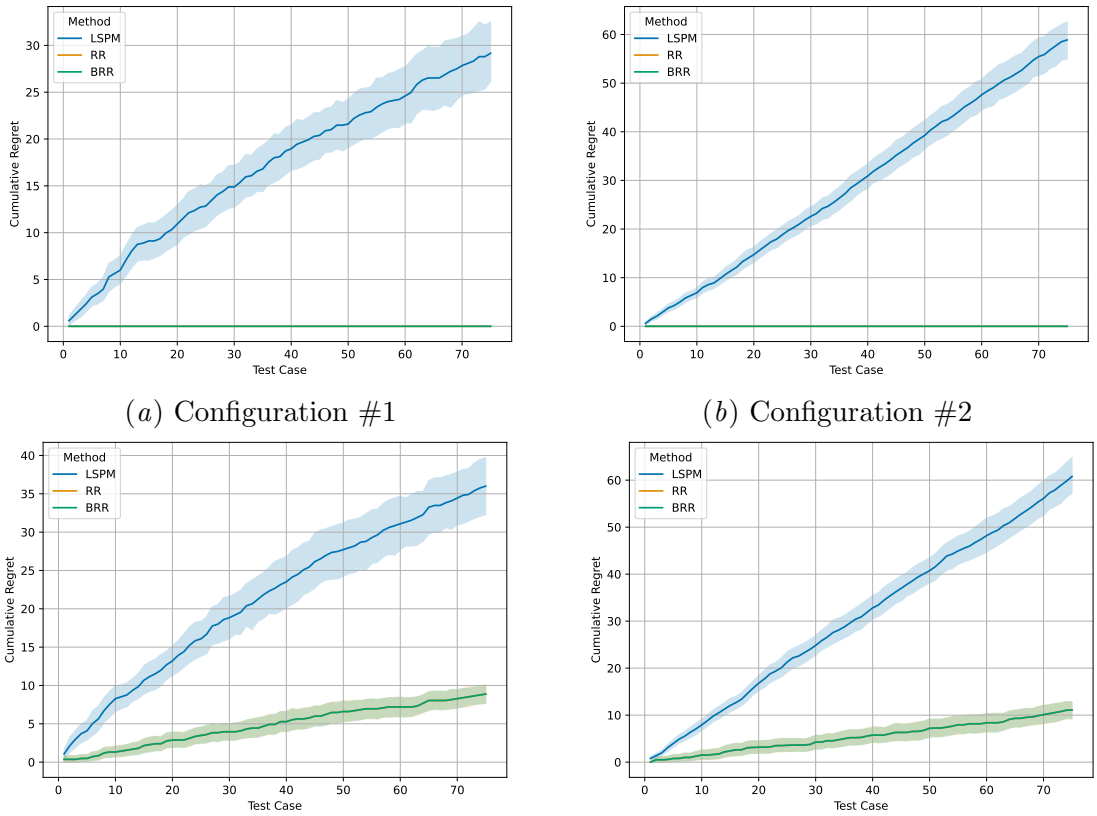
While it is possible to use Venn-Abers predictors instead of CPSs, it is more logical to begin with the version of CPDM proposed by Vovk and Bendtsen (2018), as no work has yet explored their theoretical findings. Incorporating Venn-Abers predictors would require additional adjustments and the development of new theory specifically tailored to them, which Vovk and Bendtsen (2018) did not cover. Using CPSs is a more general approach because they apply to both classification and regression decision-making settings and support continuous utility functions. Moreover, they are easier to extend to multi-class problems, as they do not require a one-vs-all approach for classification. Finally, converting a regression dataset into discrete classes will invariably lead to a loss of potentially valuable information.

4. Results

This section presents the empirical evaluation of all methods across the datasets and configurations described previously, focusing on predictive performance and robustness. Due to the large number of experiments conducted, only a selection of illustrative examples is shown. For brevity, we will refer to the PDMSs used only by their base model names. In the inductive setting, CPDM methods are denoted by the model name followed by "-CPDM" to distinguish them from PPDM. In the online setting, each graph shows cumulative regret over test cases with 95% confidence intervals based on 50 runs. The corresponding metric for the inductive setting is average utility over test cases, also reported with 95% confidence intervals based on 100 runs.

4.1. Online Setting Comparison

Figure 2 presents the results for the Linear dataset in the online learning setting across all configurations. The results showed that RR and BRR consistently outperformed LSPM, and in both Configurations #1 and #2, they demonstrated optimal performance. Although performance declined slightly when noise was introduced in Configurations #3 and #4, RR and BRR still maintained a clear advantage.

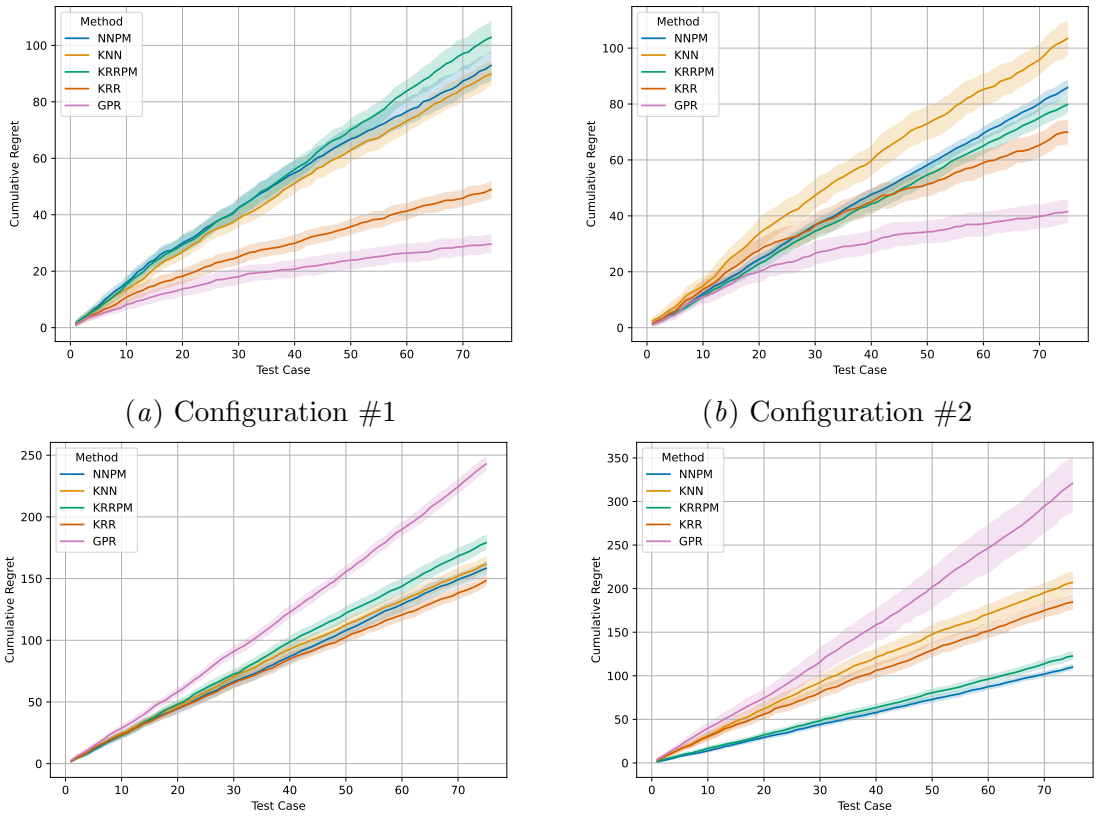


(a) Configuration #1 (b) Configuration #2

(c) Configuration #3 (d) Configuration #4

Figure 2: Online setting comparison of CPDM, PPDM, and BDT methods on the Linear dataset across configurations #1–#4.

Figure 3 shows the results for the Friedman #1 dataset. In Configuration #1, the PPDM methods outperformed their CPDM counterparts, while GPR achieved the best overall performance. All methods exhibited some ability to learn, as indicated by a reduction in the rate of increase of cumulative regret, with GPR showing the most pronounced improvement. A similar pattern was observed for Configuration #2. However, in this case, NNPM outperformed KNN, and KRRPM closed the gap significantly to KRR. In this configuration GPR still achieved the highest performance, though the performance gap to the other methods slightly decreased. Introducing noise into the data, as in Configurations #3 and #4, drastically reduced GPR's performance and its ability to learn. In Configuration #3, PPDM methods still outperformed CPDM but only marginally. However, with the skewed utility function in Configuration #4, CPDM demonstrated significantly better performance. Overall, CPDM methods appeared more robust to noise and skewness, consistently outperforming PPDM under these conditions. In contrast, GPR performed substantially worse, struggling to adapt to the more complex environment.



(*a*) Configuration #1

(*b*) Configuration #2

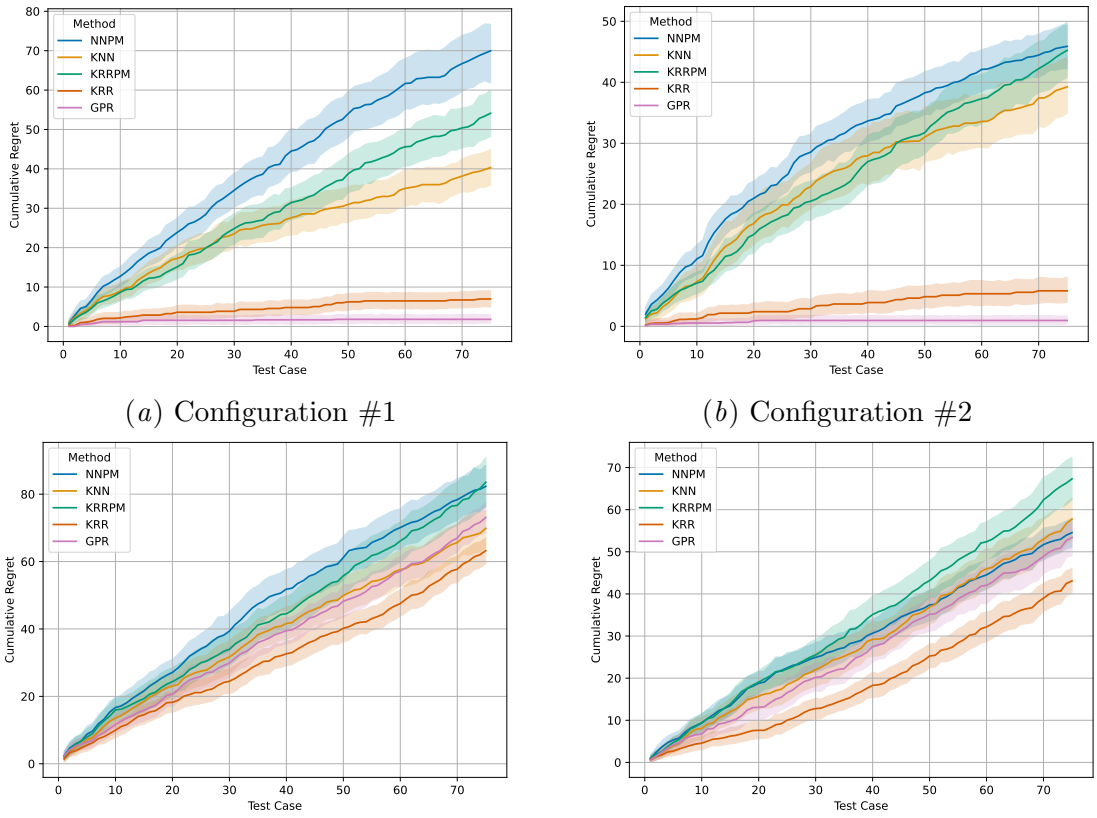
(*c*) Configuration #3

(*d*) Configuration #4

Figure 3: Online setting comparison of CPDM, PPDM, and BDT methods on the Friedman #1 dataset across configurations #1–#4.

Results for the Friedman #2 dataset are shown in Figure 4. On this dataset, GPR performed exceptionally well in Configurations #1 and #2, fully learning the decision problem. KRR also demonstrated good performance, closely following GPR. The other methods performed significantly worse but showed some signs of learning. When compared directly, PPDM consistently outperformed the CPDM methods. However, under Configuration #2, the CPDM approaches began to close the gap, though their performance remained lower overall.

In Configurations #3 and #4, when noise was introduced into the data, GPR struggled more and was now unable to fully learn the decision problem, resulting in slightly better performance compared to the CPDM methods. Similarly, PPDM methods also experienced reduced performance but still managed to perform better than CPDM. Notably, in both Configurations #3 and #4, KRR outperformed all other methods. As with the Friedman #1 dataset, CPDM demonstrated greater robustness to noise and changes in the utility function. However, for this dataset, that robustness did not translate into superior overall performance as it did for Friedman #1.



(*a*) Configuration #1

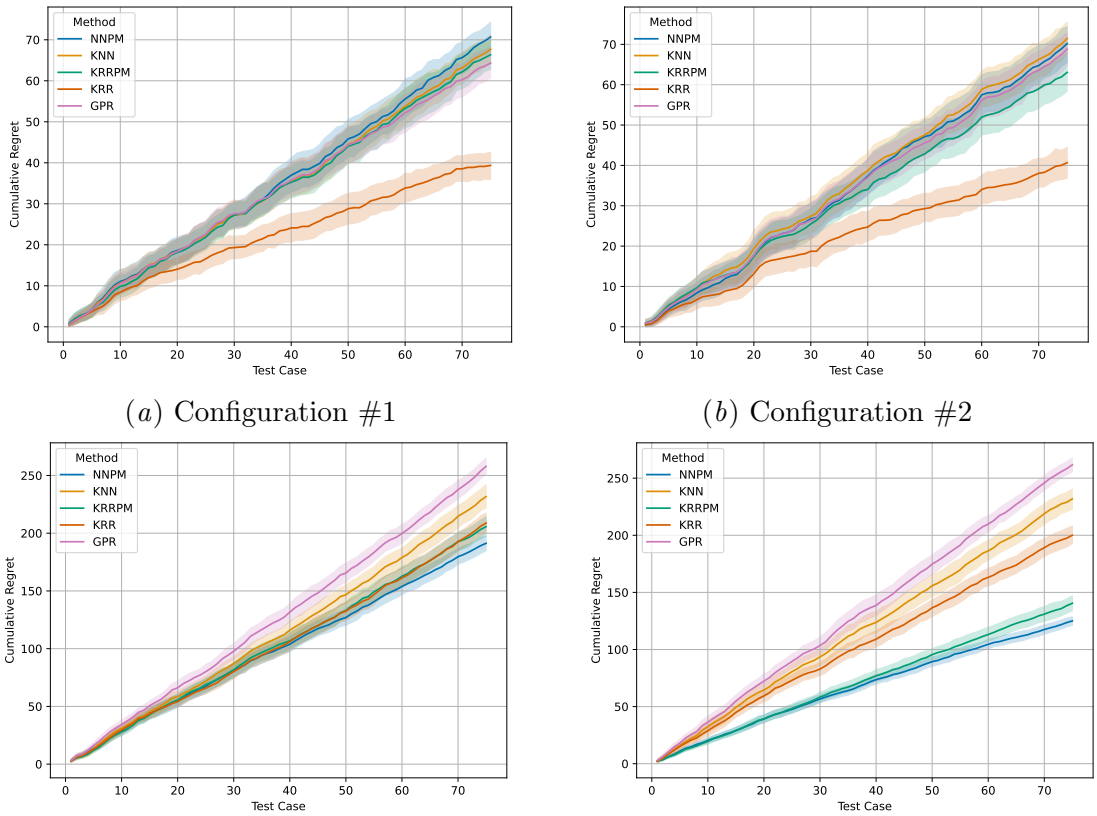
(*b*) Configuration #2

(*c*) Configuration #3

(*d*) Configuration #4

Figure 4: Online setting comparison of CPDM, PPDM, and BDT methods on the Friedman #2 dataset across configurations #1–#4.

Results for the Friedman #3 dataset are shown in Figure 5. For this dataset, all methods performed similarly in Configuration #1, except for KRR, which clearly outperformed the others and was the only method showing signs of learning the problem. The results in Configuration #2 look nearly identical to those of Configuration #1. In Configurations #3 and #4, GPR once again showed low robustness to noise, performing worst among all methods. In Configuration #3, CPDM methods slightly outperformed their PPDM counterparts, though none of the methods appeared to learn the problem. In Configuration #4, CPDM methods demonstrated the best performance overall and were the only ones to show some signs of learning the problem.



(a) Configuration #1

(b) Configuration #2

(c) Configuration #3

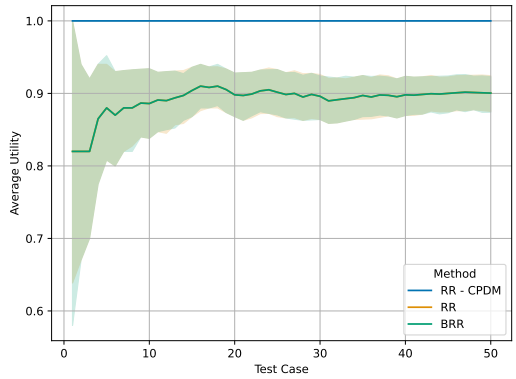
(d) Configuration #4

Figure 5: Online setting comparison of CPDM, PPDM, and BDT methods on the Friedman #3 dataset across configurations #1–#4.

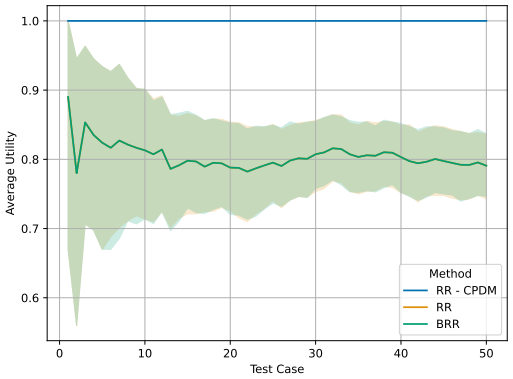
Compared to configurations using standard data availability, the results of Configurations #5, #6, #7, and #8, where less training data was used, showed that the relative performance of the methods remained largely unchanged across all datasets. However, there was some indication that CPDM could benefit from the lower amount of data, although the difference was not significant for the given quantity.

4.2. Inductive Setting Comparison

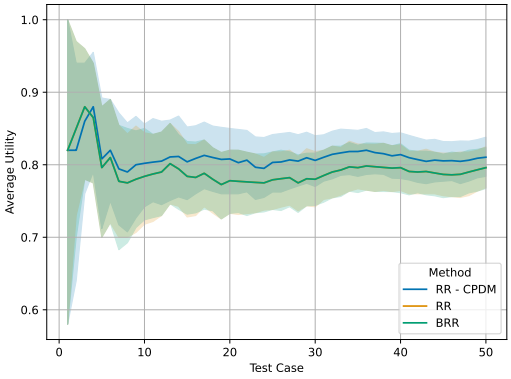
For the inductive setting on the Linear dataset, results are presented in Figure 6. In both Configurations #1 and #2, RR-CPDM achieved a perfect score on the test dataset, demonstrating strong generalization to simple problems when sufficient data is provided. BRR and RR showed worse performance compared to RR-CPDM but similar to each other, with a slight drop in average utility from Configuration #1 to #2. In Configuration #3, where noise was added to the data, RR-CPDM only marginally performed better than the other methods. However, in Configuration #4, which combined noise and a skewed utility function, RR-CPDM once again clearly outperformed all other methods. Overall, RR-CPDM showed greater robustness and handled increased complexity better than the alternative models on the Linear dataset.



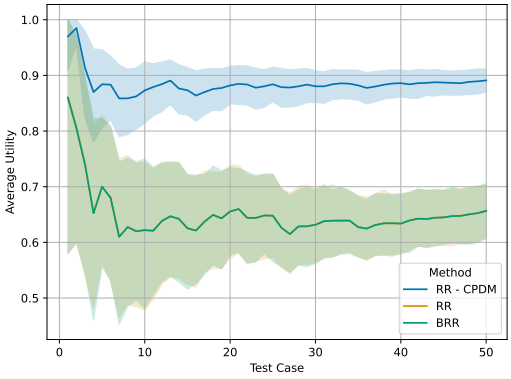
(*a*) Configuration #1



(*b*) Configuration #2



(*c*) Configuration #3

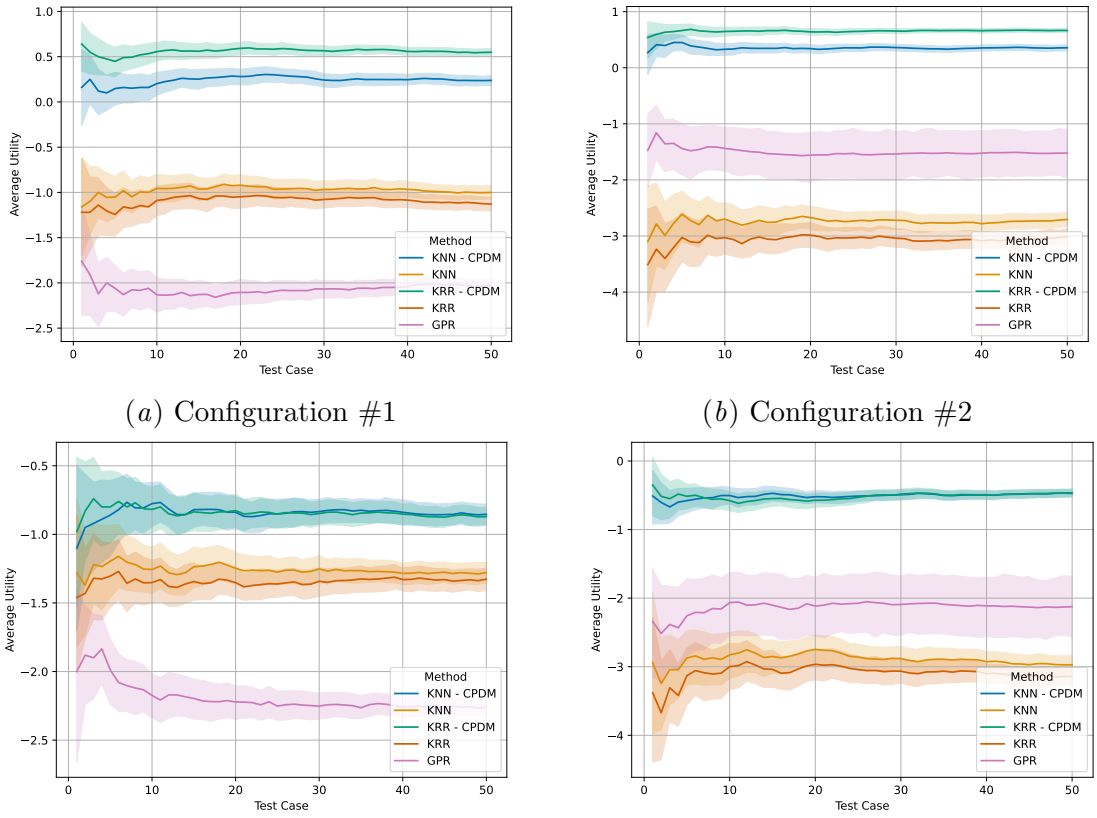


(*d*) Configuration #4

Figure 6: Inductive setting comparison of CPDM, PPDM, and BDT methods on the Linear dataset across configurations #1–#4.

Results for the Friedman #1 dataset in the inductive setting are shown in Figure 7. Here, CPDM methods clearly outperformed the others across all configurations. KRR-CPDM is the slightly better model in Configuration #1 and #2. However, KNN-CPDM performs

similarly to KRR-CPDM in Configuration #3 and #4, suggesting that KNN-CPDM is slightly more robust to noise.



(*a*) Configuration #1

(*b*) Configuration #2

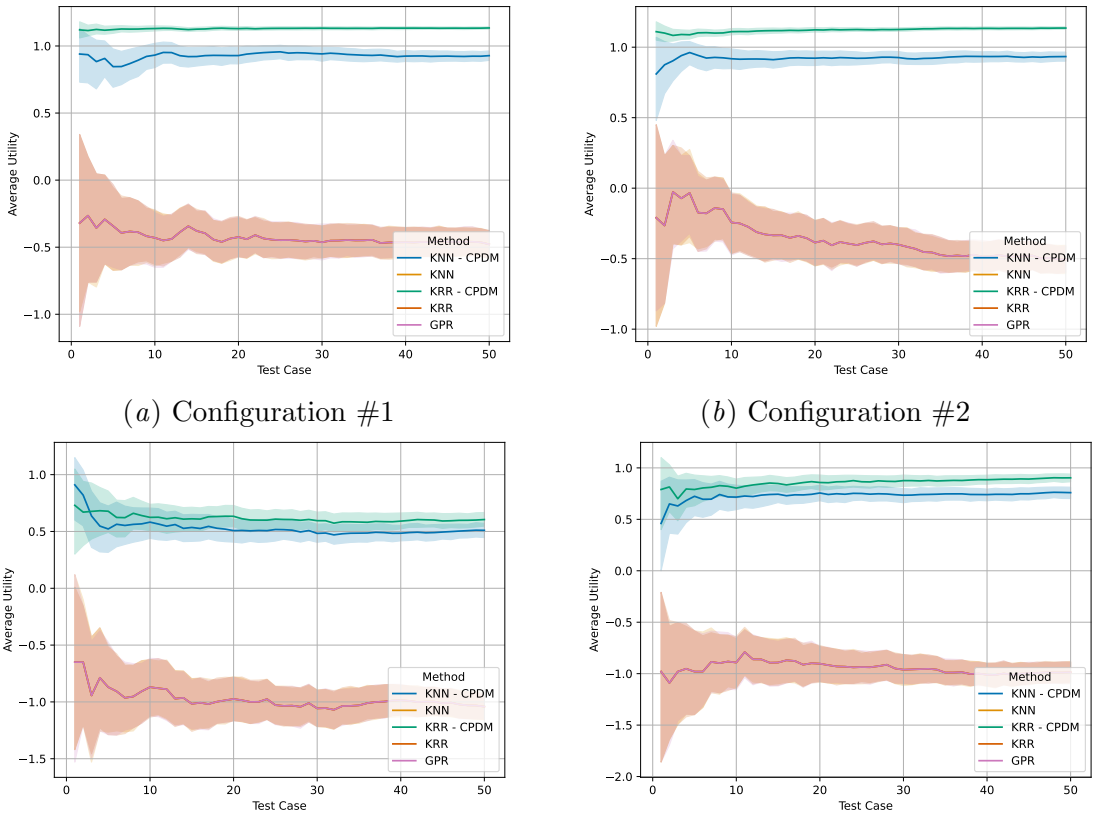
(*c*) Configuration #3

(*d*) Configuration #4

Figure 7: Inductive setting comparison of CPDM, PPDM, and BDT methods on the Friedman #1 dataset across configurations #1–#4.

A similar pattern was observed for the Friedman #2 dataset, with results shown in Figure 8. The CPDM methods clearly outperformed all others across all configurations, with KRR-CPDM achieving the best performance. Once again, KNN-CPDM showed better robustness to noise compared to KRR-CPDM, as the performance gap narrowed in Configurations #3 and #4.

Results for the Friedman #3 dataset in the inductive setting are shown in Figure 9. Once more the CPDM methods outperforms the other methods, and KNN-CPDM shows slightly better robustness to noise compared to KRR-CPDM. However, for this dataset GPR and KNN is much closer to the CPDM methods in performance for Configurations #1 and #2. Notably, the performance of KRR is extremely bad for these configurations.



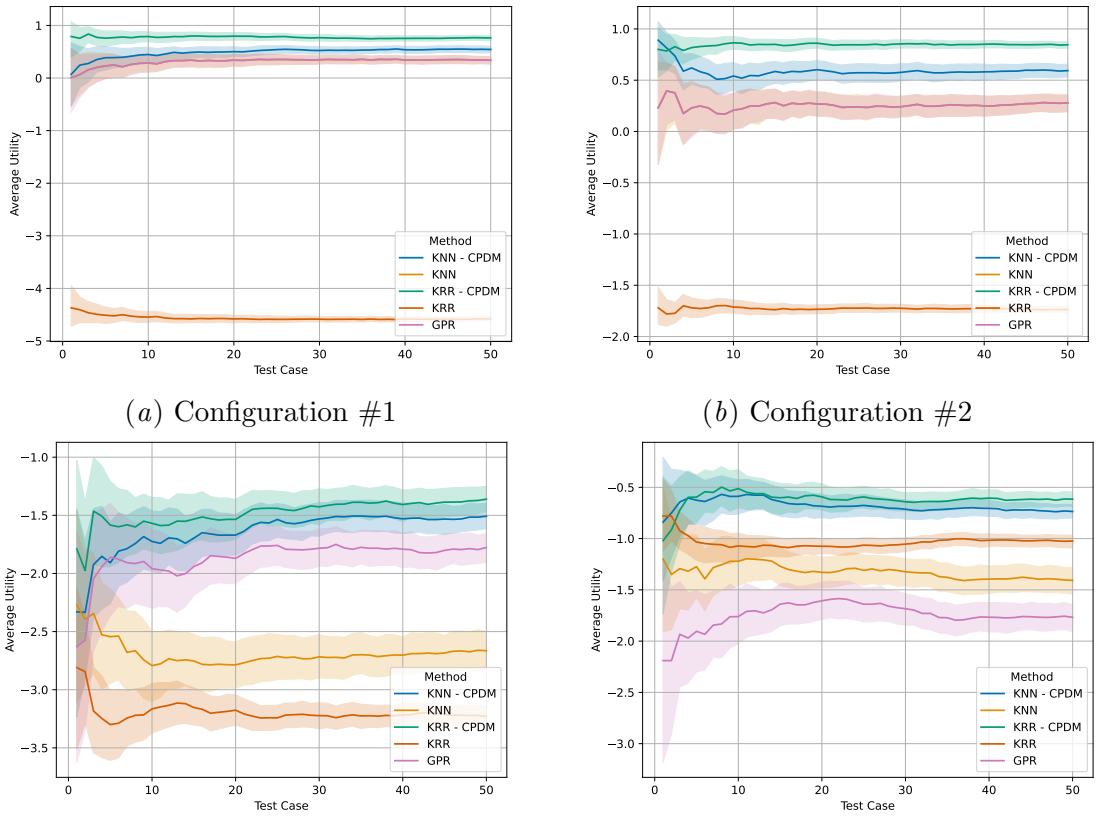
(*a*) Configuration #1

(*b*) Configuration #2

(*c*) Configuration #3

(*d*) Configuration #4

Figure 8: Inductive setting comparison of CPDM, PPDM, and BDT methods on the Friedman #2 dataset across configurations #1–#4.



(*a*) Configuration #1 (*b*) Configuration #2

(*c*) Configuration #3 (*d*) Configuration #4

Figure 9: Inductive setting comparison of CPDM, PPDM, and BDT methods on the Friedman #3 dataset across configurations #1–#4.

Compared to the configurations using standard data availability, the results of Configurations #5, #6, #7, and #8, with reduced training data, showed that the relative ranking of the methods remained largely unchanged. However, in some cases, the performance gaps were smaller, indicating that CPDM's advantage may diminish when training and calibration data are limited.

5. Discussion

In this section, we discuss the results following the same structure as before: first the online setting, then the inductive setting.

5.1. Online Setting Comparison

On the Linear dataset, BRR and RR produced nearly identical results in the online setting. This similarity is expected, as both models are based on the same underlying linear assumption, and since the predictive mean in BRR corresponds to the point prediction made by RR. This underscores how BRR can be viewed as a probabilistic generalization of RR. In our case, the only difference lies in how the regularization parameters are selected for the two methods, which might result in slightly different linear models.

When comparing CPDM against BDT and PPDM on the Linear dataset, we observed that it did not achieve comparable performance in the online setting. This highlights the cost of attaining validity at the expense of efficiency. Therefore, in situations where the assumptions of linear regression are largely satisfied, CPDM is unlikely to be particularly useful. The validity guarantees come with an efficiency cost that degrades performance. This situation is discussed in Vovk et al. (2022) under the name *Burnaev-Wasserman programme*, whose main concern is how much we have to "pay" in efficiency for the gain in validity provided by CP. If the price is low, conformalizing may be viewed as a cheap insurance policy; otherwise, it may be viewed as being over insured, as appears to be the case here.

On the nonlinear datasets in the online setting, we generally found that CPDM handled noise more effectively. In percentage terms, its performance drop was always less than or equal to that of the other methods, indicating that CPDM offers some robustness benefits. However, the other methods often outperformed CPDM overall. Among the models, KNN and NNPM consistently handled noise better than KRR and KRRPM. Notably, GPR was the worst-performing method in terms of noise robustness.

GPR's sensitivity to noise must be due to the model underestimating the noise level, causing it to fit the noise rather than capturing the true underlying trend, resulting in overfitting and poor generalization. The RBF kernel may not be expressive enough to account for the noise, and performance might have improved if the kernel had explicitly modeled it. Interestingly, KRR and KRRPM, which use the same kernel, did not exhibit the same level of degradation. This could be because hyperparameter tuning is simpler for these methods, constraining it to more robust configurations, whereas GPR has more flexibility in its hyperparameter search, which may lead to overfitting.

In terms of utility function skewness, the results were inconclusive, making it difficult to discern to what extent it affected the different frameworks in general. However, when a skewed utility function was combined with increased noise, CPDM generally outperformed the other methods on the nonlinear datasets, except for Friedman #2. This suggests that the CPDM framework is especially effective in complex scenarios.

5.2. Inductive Setting Comparison

In contrast to the online setting, the inductive CPDM methods outperformed the alternative methods on all datasets across all configurations in the inductive setting. One possible explanation for the improved performance of inductive CPDM relative to the online setting is the increased amount of data—it may require a certain volume of data to reach the same level of performance as the other two methods. However, inductive CPDM still performed comparably even when we reduced the training data, thus pointing towards something else.

Another possible explanation could be that the inductive distributions are more efficient than their online counterparts. This would mean that we would not need to "pay" an efficiency cost for the gain in validity provided by inductive CP. However, the different distributions are not comparable since they reside in different label spaces.

Data availability had little impact on the relative ranking of methods. However, there were indications that limited data availability slightly reduced the relative performance of inductive CPDM, even though it remained the best-performing approach overall. This suggests that even in the limited data configurations, the calibration sets for the ICPSs were sufficiently large to provide informative predictive distributions for reliable decision-making. Nevertheless, the performance decline could potentially be mitigated by employing more advanced inductive conformal methods, such as *cross-conformal prediction* described in Vovk et al. (2022), which do not require a separate calibration set. However, this was outside the scope of this study and is left for future research.

Overall, the consistently superior performance of inductive CPDM compared to BDT and PPDM makes it a promising alternative for practical decision-making applications. Compared to online CPDM, inductive CPDM is much more computationally efficient, which opens up the possibility of leveraging more complex underlying ML models, such as Random Forests, XGBoost, and Neural Networks, while still attaining the inductive validity guarantee for better-calibrated UQ. This, in turn, allows the framework to be applied to more complex problems with larger datasets.

6. Conclusion

In conclusion, online CPDM generally demonstrates inferior performance compared to BDT and PPDM, particularly in simpler scenarios where narrow parametric or Bayesian assumptions are likely to hold. In such cases, CPDM may introduce unnecessary overhead, making its validity guarantee excessively costly in terms of efficiency, and thus limiting its practical benefit. However, CPDM showed greater robustness than BDT and PPDM in scenarios involving noisy data and skewed utility functions for two of the four datasets, suggesting that online CPDM can be a suitable option in more complex settings.

Inductive CPDM, on the other hand, consistently outperformed the alternative methods. This, combined with its computational advantages, makes it a promising approach for larger real-world decision-making applications where well-calibrated UQ is needed for robustness.

7. Future work

Based on the results presented in this paper, several natural directions for future work emerge. First, it would be valuable to investigate whether transforming the training set using the utility function before fitting the CPS offers any benefit, or if the standard BDT approach is preferable for CPDM. Second, a direct comparison of online and inductive CPDM could provide insights into the trade-off between validity guarantees and computational efficiency. Third, it is important to evaluate CPDM's performance in real-world applications to better understand its practical usefulness.

References

Mushroom. UCI Machine Learning Repository, 1981. URL https://doi.org/10.24432/C5959T.

Ajay Agrawal, Joshua S. Gans, and Avi Goldfarb. Exploring the impact of artificial intelligence: Prediction versus judgment. *Information Economics and Policy*, 47:1–6, 2019. doi: https://doi.org/10.1016/j.infoecopol.2019.05.001.

Henrik Boström. Conformal prediction in python with crepes. In *Proceedings of the Thirteenth Symposium on Conformal and Probabilistic Prediction with Applications*, volume 230 of *Proceedings of Machine Learning Research*, pages 236–249. PMLR, 2024. URL https://proceedings.mlr.press/v230/bostrom24a.html.

Jerome H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1 – 67, 1991. doi: https://doi.org/10.1214/aos/1176347963.

Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL http://probml.github.io/book2.

Venkat Nemani, Luca Biggio, Xun Huan, Zhen Hu, Olga Fink, Anh Tran, Yan Wang, Xiaoge Zhang, and Chao Hu. Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. *Mechanical Systems and Signal Processing*, 205:110796, 2023. doi: http://dx.doi.org/10.1016/j.ymssp.2023.110796.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

Johan Hallberg Szabadváry, Tuwe Löfström, and Rudy Matela. online-cp: a python package for online conformal prediction, conformal predictive systems and conformal test martingales. Submitted Proceedings of the Fourteenth Symposium on Conformal and Probabilistic Prediction with Applications, under review, 2025.

J. von Neumann and O Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, third edition, 1953.

Vladimir Vovk and Claus Bendtsen. Conformal predictive decision making. In *Proceedings of the Seventh Workshop on Conformal and Probabilistic Prediction and Applications*, volume 91 of *Proceedings of Machine Learning Research*, pages 52–62. PMLR, 2018. URL https://proceedings.mlr.press/v91/vovk18b.html.

Vladimir Vovk and Ivan Petej. Venn-abers predictors, 2014. URL https://arxiv.org/abs/1211.0025.

Vladimir Vovk, Glenn Shafer, and Ilia Nouretdinov. Self-calibrating probability forecasting. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper_files/paper/2003/file/10c66082c124f8afe3df4886f5e516e0-Paper.pdf.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, first edition, 2005. doi: https://doi.org/10.1007/b106715.

Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min ge Xie. Nonparametric predictive distributions based on conformal prediction. In *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, volume 60 of *Proceedings of Machine Learning Research*, pages 82–102. PMLR, 2017. doi: https://doi.org/10.1007/s10994-018-5755-8.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, second edition, 2022. doi: https://doi.org/10.10007/978-3-031-06649-8.