Research Practice II

Final Report

Computational Implementation and Validation of Inductive Conformal Martingales for Change Point Detection

Isabel Mora Restrepo† and Biviana Marcela Suarez Sierra‡*

imorar@eafit.edu.co, bmsuarezs@eafit.edu.co

†Mathematical Engineering, Universidad EAFIT

‡Computing and Analytics, School of Applied Sciences and Engineering, Universidad EAFIT

May 12, 2025

Abstract

Rapidly detecting changes in hydroclimatic time series has become increasingly important and challenging due to climate change, requiring novel online change-point detection methods. This study implements and validates a non-parametric change-point detection method based on Inductive Conformal Martingales (ICMs). The theoretical results are successfully replicated, and an open-source repository is created to enhance reproducibility. The ICM algorithm is tested on synthetic data, focusing on mean delay and false alarm probability. Comparisons with established methods, including PELT and genetic algorithms, show that ICM performs competitively, particularly due to its online detection. However, the results reveal sensitivity to the chosen threshold and betting function. Future work will focus on optimizing these parameters and applying the method to real hydroclimatic data.

Keywords: Change Point Detection, Conformal Prediction, Online Detection, Non-Parametric

1 Introduction

Water resource management faces increasing challenges in a world shaped by climate uncertainty and evolving hydrological patterns that impact communities, industries, and ecosystems. According to the World Meteorological Organization (2021), between 1970 and 2019, climate-related natural disasters accounted for 50% of all global disasters, 45% of reported deaths and 74% of economic losses. In South America, such events were responsible for 60% of fatalities and 38% of economic

*Tutor

damages, with floods being the most frequent and devastating phenomenon. Given these risks, the study of hydroclimatology, particularly extreme events and their long-term trends, is essential for informed decision-making regarding water conservation, distribution, and risk management. Data-driven climate insights are also critical for industries such as agriculture, energy, urban planning, and insurance, where accurate climate understanding can improve resilience against extreme hydrological events.

A key challenge in hydroclimatic analysis is the ability to detect changes in time series data for variables such as precipitation, temperature, and streamflow. Traditional methods, such as the Mann-Kendall test and linear regression, assume stationarity in data distributions, which is often unrealistic given the complex and evolving nature of hydrological systems (Milly *et al.*, 2008). Climate change introduces additional uncertainty, altering the frequency and intensity of extreme events (Behera, 2024). Thus, reliance on outdated statistical assumptions limits the accuracy of change point detection, which is crucial for water resource planning, infrastructure development, and climate adaptation strategies. This highlights the need for advanced change point detection methods that can accurately identify shifts in hydrological patterns without relying on strict assumptions about the underlying data distributions.

This project develops a non-parametric change point detection algorithm for general time series data based on the methodology of Inductive Conformal Martingales (ICMs) introduced by Volkhonskiy *et al.* (2017a). By eliminating strong parametric assumptions about data distributions, ICMs provide a flexible approach to identifying structural breaks, making them particularly well suited for hydroclimatic applications. This research project focuses on the implementation and testing of the algorithm on synthetic data to evaluate its performance in a controlled setting. In the long term, this work is intended to contribute to a larger research initiative at Universidad EAFIT led by Alejandra María Carmona Duque and Biviana Marcela Suárez Sierra, which aims to improve change point detection techniques for real-world hydroclimatic time series, ultimately improving climate resilience, water resource management, and disaster mitigation.

The selection of ICMs for this research is driven by both theoretical and practical motivations, aiming to bridge the gap between theoretical advancements and practical implementation in change point detection (CPD). Theoretically, traditional CPD methods like CUSUM and Shiryaev-Roberts depend on assumptions about data distribution, limiting their applicability to complex, evolving time series. Non-parametric methods such as the Mann-Kendall and Pettitt tests, though more flexible, are primarily designed for offline detection and lack real-time adaptability (Aminikhanghahi & Cook, 2017). In contrast, ICMs offer a distribution-free, online framework capable of sequential monitoring without strong stationarity assumptions (Vovk *et al.*, 2003). Despite the theoretical promise of ICMs, practical implementations are scarce, as most research, predominantly conducted by a small group of authors like Vovk and Gammerman, lacks publicly available code, hindering replication and independent validation. To address this, the project implements ICMs, rigorously evaluates their performance against traditional CPD methods, and makes the implementation openly accessible to enhance reproducibility and applicability in real-world scenarios.

The scope of this research centers on evaluating the effectiveness of ICMs in detecting structural changes in time series using synthetic datasets. The primary goal is to assess the algorithm's detection delay, false alarm rate, and computational efficiency. Implemented in R for compatibility with existing statistical tools, the algorithm is directly compared with CPD methods from the `tidychangepoint` library (Baumer *et al.*, 2024), providing insights into its strengths and limitations. However, while synthetic data offers a controlled environment to validate the method, it does not fully capture the complexity of real-world time series. Therefore, future work is needed to test the method on real data to confirm its practical applicability. By developing a computational implementation, replicating results from Volkhonskiy *et al.* (2017a), and rigorously evaluating ICM performance, this research project aims to enhance the reproducibility of martingale-based CPD methods while laying the groundwork for future real-world applications, particularly for hydroclimatic time series.

This paper is structured as follows. Section 2 outlines the mathematical foundation of Inductive Conformal Martingales for non-parametric change point detection, including the use of non-conformity measures, p-values, and betting functions. This section concludes with a summary of the algorithm to be computationally implemented and validated. Section 3 reviews change point detection methods, comparing traditional parametric and non-parametric approaches. It also examines previous applications of ICMs in various fields, highlighting their potential advantages but limited availability. Section 4 presents the methodology for implementing, validating, and evaluating the ICM algorithm using R. It includes the description of mean delay and probability of false alarm as key metrics for the evaluation process. Section 5 presents the experiments conducted to validate the implementation of the ICM algorithm and assess its performance. Section 6 presents the conclusions and discusses potential areas for future work.

2 Problem Definition

Mathematically, given a time series $\{X_t\}_{t=1}^{T}$, the goal of non-parametric change point detection is to identify the point θ where the statistical properties of the data shift from one unknown distribution f_0 to another f_1, without making strong parametric assumptions about either distribution. This research project focuses on the implementation and validation of a non-parametric change point detection method based on Inductive Conformal Martingales, as proposed by Volkhonskiy *et al.* (2017a). The following subsections present the mathematical foundations and theoretical justifications necessary to understand the algorithm and its effectiveness. The final subsection presents a summary of the algorithm to be implemented and validated.

2.1 Non-Conformity Measures and P-Values

The theoretical validity of the algorithm for change point detection constructed by Volkhonskiy *et al.* (2017a) is based on the Conformal Prediction framework (Vovk *et al.*, 2005), which provides statistically valid methods for measuring the strangeness of observations under the assumption of exchangeability. Thus, the construction of the change point detection algorithm based on Inductive Conformal Martingales starts with a sequence of observations $z_1, z_2, \ldots, z_n, \ldots$, where each observation is a vector that takes values in some observation space $\mathbb{R}^d$. The observations are assumed to be exchangeable, meaning that the joint probability distribution of $z_1, z_2, \ldots, z_N$ for any finite N remains invariant under any permutation of these random vectors.

Once the sequence of observations is defined, it is now desirable to measure how strange a new observation z_n is compared to previous observations $z_1, z_2, \ldots, z_{n-1}$. To accomplish this, Volkhonskiy *et al.* (2017a) introduce the concept of **non-conformity measure** (NCM), defined as a function

$$(z, S) \mapsto A(z, S) \tag{1}$$

that quantifies how different an observation z is from a given (finite) reference set S. Intuitively, the non-conformity measure reflects how unusual the new observation is, with higher values of $A(z, S)$ indicating greater strangeness relative to S. Section 2.1.1 provides examples of non-conformity measures that are used in the implementation of the ICM algorithm.

Based on the values obtained from the nonconformity measure, it is now possible to construct a **p-value** for the observation z_n:

$$p_n = p(z_n, z_{n-1}, ..., z_1) = \frac{\#\{i = 1, ..., n : \alpha_i > \alpha_n\} + U\#\{i = 1, ..., n : \alpha_i = \alpha_n\}}{n} \tag{2}$$

where U is a random number in [0,1] independent of $z_1, z_2, ...$, and the non-conformity scores α_i are defined by

$$\alpha_i = A(z_i, \{z_1, ..., z_{i-1}, z_{i+1}, ..., z_n\}).$$

That is, the p-value for the observation z_n is approximately defined as the proportion of observations with non-conformity scores greater than or equal to α_n. Intuitively, a smaller p-value indicates that the observation is more unusual.

While the non-conformity measure directly quantifies the strangeness of an observation, the p-value normalizes this measure by comparing it with all previous observations, providing a probabilistic interpretation of how likely it is to observe such a strange value. This transformation ensures that p-values lie within the interval $[0, 1]$, making them easier to interpret.

In the absence of change, the p-values are uniformly distributed in $[0, 1]$ (Volkhonskiy *et al.*, 2017a). This property is fundamental for change point detection because:

- Before the change point θ, the observations $z_1, \ldots, z_{\theta-1} \sim f_0(z)$
- After the change point θ, the observations $z_\theta, z_{\theta+1}, \ldots \sim f_1(z)$
- When no change has occurred, all the observation are i.i.d. and the sequence of p-values $p_1, p_2, \ldots, p_n, \ldots$ remains uniformly distributed in $[0, 1]$.
- Once a change occurs at $\theta \geq 2$, the distribution of the subsequent p-values ($n > \theta$) is no longer uniformly distributed.

This shift in the distribution of p-values is the core idea behind the construction of Conformal Martingales for change point detection. The deviation from uniformity serves as an indicator of a potential change point.

2.1.1 Examples of Non-Conformity Measures

According to Equation (1), a non-conformity measure can be any function that takes a new observation z, a reference set S, and outputs a scalar value α. However, for the ICM methodology to function effectively, it is essential to construct measures that quantifies how different a new observation is from the reference set. Since the concept of "different" is inherently subjective, there are numerous possible choices for NCMs. Three examples are presented:

- **Distance to the Mean**: Quantifies how far an observation z_n is from the mean of a reference set. Mathematically, it is defined as:

  $$\alpha_n = A(z_n, S) = ||z_n - \mu||, \tag{3}$$

  where S is the reference set of previous observations and

  $$\mu = \frac{1}{|S|} \sum_{i \in S} z_i$$

  is the mean of the reference set.

  This NCM is useful when the reference distribution is assumed to be approximately centered around the mean, making it suitable for detecting deviations in the average value of the data.

- **k Nearest Neighbors (kNN)**: Measures how far an observation z_n is from its k nearest neighbors in the reference set. It is computed as the average distance to the k nearest neighbors:

  $$\alpha = A(z_n, S) = \frac{1}{k} \sum_{i=1}^{k} d(z_n, z_{(i)}), \tag{4}$$

  where S is the reference set of previous observations and $d(z_n, z_{(i)})$ is the distance between the observation z_n and its i-th nearest neighbor $z_{(i)}$ in the set S. Typically, d is taken as the Euclidean distance, but the kNN non-conformity measure can be used with any valid metric.

  The kNN NCM is advantageous because it does not depend on any distributional assumptions, making it applicable to multi-dimensional data.

- **Likelihood Ratio**: Compares the likelihood of an observation z_n under the pre-change distribution $f_0(z)$ and the post-change distribution $f_1(z)$. It is defined as:

  $$\alpha_n = A(z_n, S) = \frac{f_1(z_n)}{f_0(z_n)} \tag{5}$$

  In practice, the true distributions $f_0(z)$ and $f_1(z)$ are rarely known exactly. Thus, it is assumed that both distributions belong to a parametric family of densities

  $$f_i(z) = f(z \mid \mathbf{c}_i) \quad i = 0, 1$$

  The pre-change parameters $\mathbf{c}_0$ are estimated from a training set of observations, while the post-change parameters are obtained by incorporating a prior distribution $r(\mathbf{c}_1)$ that represents the expected characteristics of the post-change data (Volkhonskiy *et al.*, 2017a).

For the normal distribution in the one-dimensional case, $f(z \mid \mu_1) = \mathcal{N}(z \mid \mu_1, \sigma^2)$ and $r(\mu_1) = \mathcal{N}(z \mid \mu_1, \sigma_r^2)$, the Likelihood Ratio NCM becomes:

$$\alpha_n = A(z_n, S) = \frac{\mathcal{N}(z_n \mid \mu_r, \sigma^2 + \sigma_r^2)}{\mathcal{N}(z_n \mid \hat{\mu_0}, \sigma^2)} \tag{6}$$

where:

- μ_1: Mean of the post-change distribution (unknown).
- μ_r: Prior mean for the post-change distribution (estimate of μ_1).
- σ^2: Variance of the distribution before the change.
- σ_r^2: Variance of the prior distribution for μ_1, representing the uncertainty about the post-change mean.

The likelihood ratio NCM can be useful when there is prior knowledge or reasonable assumptions about the underlying data distributions, allowing for more targeted change point detection. However, relying on parametric assumptions contradicts the fundamental non-parametric philosophy of the ICM method, as it imposes a fixed model structure on a technique specifically designed to avoid such assumptions, potentially limiting its robustness in real-world applications.

2.2 Exchangeability Martingales and Betting Functions

A **test exchangeability martingale** is a sequence of non-negative random variables $S_0 = 1, S_1, S_2, \ldots$ such that:

$$\mathbb{E}(S_{n+1} \mid S_1, \ldots, S_n) = S_n \quad n = 0, 1, 2, \ldots \tag{7}$$

where $\mathbb{E}$ is taken with respect to any exchangeable distribution of the observations.

Condition (7) means that the expected value of the next step, given the current value and all previous values, remains unchanged. In other words, a test exchangeability martingale represents a fair game, where the process does not exhibit any systematic increase or decrease in expected value.

In the context of change point detection, exchangeability martingales are used to monitor deviations from the exchangeability assumption, a core principle of the ICM methodology. Exchangeability ensures that the joint distribution of a set of observations remains invariant under permutation, meaning that the order of the data does not affect their distribution. According to de Finetti's theorem, any infinite sequence of exchangeable random vectors can be represented as a mixture of i.i.d. distributions. This justifies the use of exchangeability martingales because, while the data before and after a change point may each follow an i.i.d. structure, the overall sequence may lose this property when a shift occurs.

Monitoring the growth of the martingale value allows for the detection of deviations from the exchangeability assumption. If the final value of the martingale becomes significantly large, it provides strong evidence against the i.i.d. (or exchangeability) hypothesis according to Ville's inequality (Volkhonskiy *et al.*, 2017a), indicating a potential change point.

With this theoretical justification in mind, Volkhonskiy *et al.* (2017a) construct a test exchangeability martingale based on the p-values obtained through expression (2):

$$S_n = \prod_{i=1}^{n} g_i(p_i) \tag{8}$$

where $g_i(p_i) = g_i(p_i \mid p_1, \ldots, p_{i-1})$ is a **betting function** that must satisfy the condition:

$$\int_0^1 g_i(p)\, dp = 1, \quad \forall i$$

Martingales of the form given in (8) are referred to as **conformal martingales**, and they can be verified to satisfy the property described in (7).

The choice of the betting function g plays a crucial role in the ICM algorithm, as it determines the growth behavior of the martingale. Section 2.2.1 provides examples of betting functions that are used in the implementation of the ICM algorithm. The underlying intuition is that the betting function should amplify the martingale value when the p-values deviate from the expected uniform distribution. In particular, since smaller p-values indicate more unusual observations, the betting function should assign larger weights to lower p-values, thereby penalizing non-uniformity. This strategy ensures that the conformal martingale grows significantly when the observed data increasingly deviates from the exchangeability assumption, indicating a potential change point.

2.2.1 Examples of Betting Functions

According to the definition given by Equation (8), a betting function can be any $g(p)$ that integrates to 1 over $(0, 1)$. Thus, multiple types of betting functions can be constructed, and their characteristics determine the behavior of the corresponding Conformal Martingale. Three examples are presented:

- **Mixture Betting Function:** Proposed by Vovk *et al.* (2003) in the early work on testing exchangeability. It does not depend on previous p-values and is defined as:

  $$g(p) = \int_0^1 \epsilon\, p^{\epsilon - 1}\, d\epsilon \tag{9}$$

  This function is useful when there is little prior knowledge about the distribution of p-values.

- **Constant Betting Function:** Divides the interval $[0, 1]$ into two parts at the midpoint 0.5. It takes higher values for smaller p-values, reflecting the intuition that smaller p-values indicate more unusual observations:

  $$g(p) = \begin{cases} 1.5, & \text{if } p \in [0, 0.5) \\ 0.5, & \text{if } p \in [0.5, 1] \end{cases} \tag{10}$$

  This betting function is useful when the distribution of p-values is expected to be relatively balanced around the midpoint, and when the focus is on quickly identifying unusually small p-values that may indicate change points.

- **Precomputed Kernel Density Betting Function:** Aims to measure how likely it is to observe a given p-value. Since calculating the exact density function underlying the distribution of p-values is impractical, it is approximated using kernel density estimation (Niranjan Pramanik, 2019). The form of the betting function is:

  $$g_n(p_n) = K_{p_{n-L},\dots,p_{n-1}}(p_n) \tag{11}$$

  where $K_{p_{n-L},\dots,p_{n-1}}(p_n)$ represents the Parzen-Rosenblatt kernel density estimate based on the L most recent p-values.

  To avoid having to recalculate the density function for every new p-value, the kernel density is precomputed before constructing the martingale. This process involves:

  - Collecting a set of training p-values that are representative of typical change point scenarios.
  - Applying kernel density estimation (KDE) to the training set of p-values. A Gaussian kernel is typically used, with bandwidth calculated as:

    $$h = 0.9\,\sigma\,n^{-0.2}$$

    where σ is the standard deviation or interquartile range, and n is the number of training samples.
  - Extending the training set by adding reflected values to mitigate boundary effects, ensuring that the density estimate is well-defined within the interval $[0, 1]$.
  - Normalizing the density over the interval $[0, 1]$ to ensure it integrates to 1.

  This betting function is useful when the distribution of p-values can be reasonably estimated from a representative training set, allowing efficient evaluation during the change point detection process.

2.3 Inductive Conformal Martingales

Inductive Conformal Martingales are a computationally efficient modification of the Conformal Martingales (CMs) defined by (8). Unlike the original CMs, which require updating the reference set at each step to compute all the non-conformity scores, ICMs maintain a constant initial training set. This approach significantly reduces computational cost and allows for an efficient online change point detection. A fixed test set $\{z^*_{-(m-1)}, \dots, z^*_0\}$ is used to compute the non-conformity score for each new observation z_n as follows:

$$\alpha_i = A(z_i, \{z^*_{-(m-1)}, \dots, z^*_0\}) \tag{12}$$

Additionally, a drawback of the original CMs is that when the data is i.i.d., the martingale value tends to decrease to almost zero. Since the martingale is represented as a product of betting functions, it struggles to recover quickly to a non-zero value when"strange" observations appear later on. To address this issue, the ICM framework introduces a modified logarithmic formulation for the martingale value, defined as:

$$C_n = \log S_n - \min_{i=1,\dots,n} \log S_i \tag{13}$$

where S_n is the standard conformal martingale value at time n as defined by (8).

This new expression measures the growth of the current martingale value relative to its minimum past value, ensuring that the martingale does not collapse entirely when faced with i.i.d. data. However, since the logarithmic transformation can produce negative values (when S_n drops below 1), this can significantly slow down the recovery of the martingale when a change occurs after a long period of similar data. To address this issue, the ICM framework introduces the **circumscribed martingale**, defined recursively as:

$$C_n = \max\{0, C_{n-1} + \log(g_n(p_n))\} \tag{14}$$

where $C_0 = 0$ and $g_n(p_n)$ is the n-th betting function applied to the p-value at time n.

The key idea behind this modification is to cut off negative values, ensuring that the martingale value never drops below zero. This adjustment allows the circumscribed martingale to grow faster and recover more efficiently when unusual observations appear. Therefore, expression (14) is the one used to compute the actual martingale value within the ICM algorithm. The stopping rule for change point detection has the form:

$$\tau_{CM} = \inf\{n : C_n \geq h\} \tag{15}$$

where h is a predefined threshold indicating significant growth of the martingale.

2.4 Algorithm Summary

The theoretical framework developed by Volkhonskiy *et al.* (2017a) and detailed in the previous subsections is summarized in Algorithm 1.

This research project focuses on the computational implementation and validation of Algorithm 1. The primary objective is to replicate the results presented in the guiding article (Volkhonskiy *et al.*, 2017a) by systematically testing the algorithm on synthetic datasets. Additionally, the project aims to evaluate the effectiveness of ICMs by comparing their performance against well-established change point detection methods, thereby assessing their potential as a reliable alternative for detecting changes in real-world time series data.

Algorithm 1 Inductive Conformal Martingale Algorithm

Input: Training set $\{z^*_{-(m-1)}, \dots, z^*_0\}$, data $\{z_1, z_2, \dots\}$, threshold h, betting function $g(p)$, non-conformity measure A

Output: Inductive Conformal Martingale $(S_n)_{n \geq 1}$ and its modification $(C_n)_{n \geq 1}$

Initialize $S_0 \leftarrow 1$, $C_0 \leftarrow 0$

for $n = 1, 2, \dots$ **do**

  Observe new data point z_n

  Calculate non-conformity score:

$$\alpha_n \leftarrow A(z_n, \{z^*_{-(m-1)}, \dots, z^*_0\})$$

  Calculate p-value:

$$p_n \leftarrow \frac{\#\{i = 1 \dots n : \alpha_i > \alpha_n\} + U\,\#\{i = 1 \dots n : \alpha_i = \alpha_n\}}{n}$$

  Update CM value:

$$S_n \leftarrow S_{n-1} \cdot g_n(p_n)$$

  Update circumcised ICM value:

$$C_n \leftarrow \max\{0, C_{n-1} + \log(g_n(p_n))\}$$

  Set $\tau_n \leftarrow$ TRUE if $C_n \geq h$ else $\tau_n \leftarrow$ FALSE

end for=0

3 State of the art

Change point detection (CPD) is the process of identifying points in a time series where the data's statistical properties, such as mean, variance, or distribution, undergo a significant shift. It is widely applied in domains such as finance (Zhu *et al.*, 2015), climate science (Gallagher *et al.*, 2013) and medicine (Chen *et al.*, 2019), fields where detecting structural changes is crucial for decision-making.

Traditional CPD techniques can be categorized as parametric or non-parametric. Parametric methods assume a known data distribution and often rely on likelihood-based techniques. The Cumulative Sum (CUSUM) test (Page, 1954) monitors cumulative deviations from a baseline and signals a change when a predefined threshold is exceeded. The Shiryaev-Roberts procedure (Shiryaev, 1963) applies Bayesian decision theory to sequentially test for changes in distribution. While effective in structured settings, these methods require prior knowledge of the data distribution, making them less adaptable to real-world variability (Aminikhanghahi & Cook, 2017).

Non-parametric methods make fewer assumptions about the underlying distributions, offering greater flexibility and scalability, but they can be less efficient and heavily dependent on large amounts of data (Aminikhanghahi & Cook, 2017). The Mann-Kendall test is widely used for trend detection, but is not designed for abrupt changes (Kamal & Pachauri, 2018). Pettitt's test, based on the Mann-Whitney statistic, detects a single change point in a time series but lacks scalability for multiple changes (Pettitt, 1979). The Kolmogorov-Smirnov test, which compares empirical distributions before and after a suspected change, is effective for distributional shifts, but computationally expensive for sequential monitoring (Berger & Zhou, 2014).

CPD methods can also be classified as offline or online. Offline methods analyze the entire dataset retrospectively, identifying all change points in batch mode, while online methods operate in real-time, detecting changes as soon as they occur, ideally before the next observation arrives (Aminikhanghahi & Cook, 2017). Most classical approaches, particularly non-parametric tests, are designed for offline detection, limiting their use in dynamic and real-time applications.

To address the limitations of traditional CPD methods, martingale-based approaches have emerged as a flexible alternative, particularly for online detection. Conformal Martingales (CMs), first introduced by Vovk *et al.* (2003), provide a distribution-free approach that sequentially monitors deviations from expected behavior, as explained in Section 2. The first use of CMs for change point detection was done by Ho (2005), who applied two different martingale tests to detect concept drift in data streams. Later, Fedorova *et al.* (2012) extended the framework by introducing plug-in martingales, which improved flexibility by allowing adaptive updates to non-conformity measures. These early studies established CMs as a theoretically valid tool for testing exchangeability and detecting shifts in data distributions.

As explained in Section 2.3, a key drawback of standard CMs is their tendency to decay to near-zero values under i.i.d. conditions. Additionally, their computational demands increase significantly when applied to high-dimensional data or long time series, limiting their scalability. To address these issues, Inductive Conformal Martingales (ICMs) were introduced by Volkhonskiy *et al.* (2017a). The use of a fixed test set during computations allows ICMs to be more practical for real-time applications, particularly in domains that require fast and adaptive change detection.

Eliades & Papadopoulos (2021) applied ICMs with a histogram betting function to detect concept drift in streaming classification tasks, demonstrating improved efficiency compared to existing methods. Similarly, Vovk (2021) reviewed the role of conformal martingales in online randomness testing and developed nonparametric conformal variants of the CUSUM and Shiryaev-Roberts procedures, strengthening their application in change detection without relying on strong distributional assumptions. In the field of geospatial analysis, Wang *et al.* (2023) introduced a transformation of the Conformal Test Martingale to detect change points in geospatial object detectors, improving the efficiency of monitoring domain shifts in remote sensing imagery. Despite these advances, publicly available and open source implementations of ICMs remain limited.

4 Methodology

The ICM algorithm described in Section 2 is implemented, validated, and evaluated using R, chosen for its robust integration with statistical functions and libraries. The complete code, including the algorithm and all performed experiments, is openly accessible in a GitHub repository[1]. Sharing the created code publicly promotes the development of an open-source method that can advance change point detection techniques for real-world data applications.

[1] `https://github.com/isabelmorar/ICM_CP_Detection`

4.1 Algorithm Implementation

The implementation of the Inductive Conformal Martingale algorithm follows the theoretical framework presented in Section 2. It is implemented following the pseudocode presented in Algorithm 1 and consists of the following key components:

- **ICM Main Algorithm:** Implemented by the main function `icm_changepoint_detection`. It takes as input a training set, a set of new observations, a threshold h, a non-conformity measure, and a betting function. The algorithm proceeds as follows:
  - The training set is shuffled to ensure exchangeability.
  - The algorithm iterates over each new observation z_n, processing data sequentially. For each iteration, the non-conformity score α_n is calculated with respect to the training set, and the corresponding p-value is computed according to Equation (2).
  - The martingale value S_n is updated according to Equation (8) using the betting function, and the circumscribed ICM value (C_n) is calculated according to Equation (13).
  - The algorithm records whether a change point is detected based on the condition $C_n \geq h$.
  - Since the set of new observations is finite, the algorithm continues its execution until all the new observations have been processed through the main loop, even if a change point has already been detected. This approach ensures that the martingale values are computed for the entire time series, allowing for a complete visualization of its behavior after the algorithm's execution.
  - The function outputs a data frame containing the results of the ICM algorithm. Each row corresponds to an observation z_n and includes the calculated martingale value S_n, the modified ICM value C_n, and a boolean indicating whether a change point was detected.
- **Non-Conformity Measures (NCMs):** The implemented ICM algorithm supports any non-conformity measure, as described in Section 2.1. The three non-conformity measures introduced in Section 2.1.1 are implemented in the file `Functions/non_conformity_measures.R` as base examples to properly test and validate the algorithm. The following considerations should be noted:
  - The Distance to the Mean and k Nearest Neighbors (kNN) NCMs are implemented from scratch in R, following the exact formulations presented in Section 2.1.1. The implementation leverages built-in functions and libraries to ensure computational efficiency and accuracy.
  - Two versions of the Likelihood Ratio NCM are implemented specifically for normally distributed data. The base version, as described in Section 2.1.1, maintains a fixed μ_r for the post-change distribution. This version is suitable when prior knowledge about the data distribution is available. A more flexible version is implemented using rolling windows to dynamically update the post-change distribution as new data is processed. This approach is beneficial when the data characteristics may vary over time or are not known exactly.

- **Betting Functions (BFs):** The implemented ICM algorithm supports any valid betting function, as described in Section 2.2. However, the current implementation assumes that the same betting function is consistently used throughout a single execution, without varying g for each p-value. This design choice simplifies the code structure. The algorithm can be extended in the future to incorporate cases where the betting function may vary at each iteration.

  The three betting functions introduced in Section 2.2.1 are implemented in the file `Functions/betting_functions.R` as base examples to properly test and validate the algorithm. The following points should be noted:

  - The Constant and Mixture betting functions are implemented from scratch in R, following the exact formulations presented in Section 2.2.1. These implementations make use of built-in functions and libraries to ensure computational efficiency and accuracy.
  - In the case of the Precomputed Kernel Density betting function, the test set used during the algorithm execution is used to calculate a sample of p-values using the corresponding NCM. The kernel density estimated from this sample is then passed to the main algorithm as the betting function.

Overall, the implementation is modular and flexible, allowing the selection of non-conformity measures and betting functions according to data characteristics and the specific change point detection scenario. At the same time, it remains true to the theoretical foundations and original formulations given by Volkhonskiy *et al.* (2017a).

4.2 Algorithm Performance Evaluation

The performance evaluation of the ICM algorithm centers on two key metrics: mean delay and probability of false alarm. These metrics are essential for assessing change point detection methods, as they capture both the speed of detection and the resilience to false positives. However, within the ICM framework, these metrics are often inversely related: achieving a smaller mean delay typically increases the probability of false alarms. To address this, the evaluation performed involves systematically varying the threshold h in the ICM algorithm to analyze the balance between rapid detection and reliability.

The mean delay is calculated as the average number of observations between the true change point and the first correctly detected change point. Mathematically:

$$\text{Mean Delay} = \mathbb{E}(\tau - \theta \mid \tau > \theta) \tag{16}$$

where τ is the time of the first detection and θ is the true change point.

The probability of false alarm quantifies the likelihood of detecting a change point before the actual change occurs. It is defined as:

$$\text{Probability of False Alarm} = \mathbb{P}(\tau \leq \theta) \tag{17}$$

The functions for calculating the key performance metrics and conducting trials are implemented in the file `Functions/metrics_and_trials.R`. Some annotations on their implementation are given:

- **Delay Calculation:** The function `delay()` computes the difference between the detected change point and the true change point if the detection occurs after the actual change. If no valid detection occurs, the function returns `NA`.
- **False Alarm Calculation:** The function `false_alarm()` verifies whether a detected change point occurs before the actual change point. If this is the case, the function returns 1, indicating a false alarm; otherwise, it returns 0.
- **Conducting Trials:** The function `conduct_trials()` performs a specified number of trials of the ICM algorithm using simulated data, following the experimental setup described in the guiding article (Volkhonskiy *et al.*, 2017a). In each trial, the following steps are carried out:
  - The function generates synthetic data where a change occurs at a predefined point θ. The data before the change is sampled from a normal distribution $\mathcal{N} \sim (0, 1)$, while the data after the change is drawn from $\mathcal{N} \sim (\mu_1, 1)$.
  - The ICM algorithm is applied to the generate data to detect a change point using a predefined non-conformity measure and betting function.
  - For each trial, the mean delay and probability of false alarm are calculated according to Equations (16) and (17). The mean delay is computed as the average delay across all trials, excluding cases where detection occurred before the actual change point or did not occur at all (resulting in `NA` values). The probability of false alarm is calculated as the ratio of false detections to the total number of trials.
- **Multiple Threshold Trials:** The function `run_multiple_threshold_trials()` systematically evaluates the algorithm's performance over a range of threshold values. For each value of h in the specified range, the function calls `conduct_trials()` and records the corresponding metrics.
- **Comparative Evaluation:** The function `evaluate_tidychangepoint_method()` runs trials using traditional change point detection methods available in the R package `tidychangepoint` (Baumer *et al.*, 2024). These methods include PELT, Binary Segmentation, and genetic algorithm-based approaches. The function computes the same performance metrics (mean delay and probability of false alarm) as the trials for the ICM algorithm, enabling a direct and consistent comparison of their effectiveness.

5 Experiments and Results

This section presents the experiments conducted to evaluate the functioning and performance of the ICM algorithm described in Section 2, following the methodology outlined in Section 4. The complete code, including the experiments and the generated graphs, is available in the file `ICM_Algorithm_Results.Rmd` of the GitHub repository[2]. Given that the data generation and solution method involve a random component, accessing the complete code and the specific seed used is essential for reproducibility.

[2] `https://github.com/isabelmorar/ICM_CP_Detection`

5.1 Demonstration of Algorithm Functioning

The first experiment conducted involves testing the algorithm on synthetic data to analyze the behavior of the martingale under different conditions. To achieve this, multiple runs were performed while varying the distributions, betting function, non-conformity measure, and threshold. Two representative examples are presented:

- **Normal distribution before and after the change point**: The data consists of 100 observations, where the first 50 observations follow a normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$, representing the period before the change point at $\theta = 50$. After the change point, the remaining 50 observations follow a normal distribution $\mathcal{N}(\mu = 2, \sigma = 1)$. An additional 10 observations are generated from the first distribution to serve as a training set. The algorithm was executed using the 7 Nearest Neighbors NCM, the Constant betting function, and a threshold of 3.5. The experimental setup and results are shown in Figure 1.

  The algorithm's effectiveness is evident from the martingale behavior shown in the graph on the right of Figure 1. Immediately after the indicated change point, the martingale value increases significantly, quickly surpassing the threshold at observation 56, indicating a delay of only 6 observations.

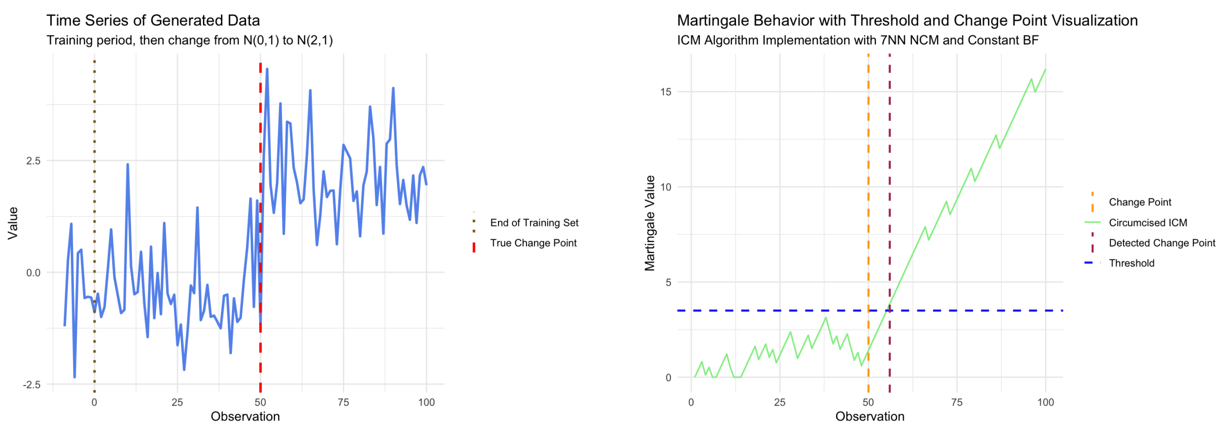

Figure 1

- **Uniform distribution before and exponential distribution after the change point**: The data consists of 100 observations, where the first 50 observations follow a uniform distribution $\mathcal{U}(a = 0, b = 2)$, representing the period before the change point at $\theta = 50$. After the change point, the remaining 50 observations follow an exponential distribution $\text{Exp}(\lambda = 0.5)$. An additional 10 observations are generated from the first distribution to serve as a training set. The algorithm was executed using the distance to the mean NCM, the mixture betting function, and a threshold of 1.5. The experimental setup and results are shown in Figure 2.

  The algorithm's performance is better in this example, as the martingale value surpasses the threshold by observation 52. This could be due to the more pronounced change in the data after the change point (graph on the left of Figure 2), which likely results in higher non-conformity scores. However, the martingale graph also shows a decline towards the end, indicating that the selected combination of NCM and BF is causing the algorithm to adapt to the new observations, reverting to the exchangeability hypothesis.

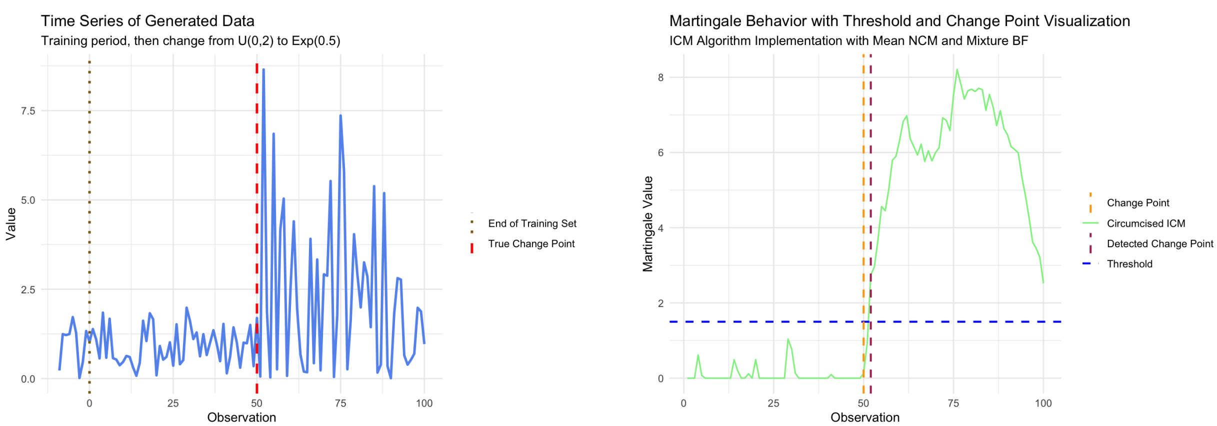

Figure 2

5.2 Replication of Main Figures from the Guiding Article

After confirming the initial functioning of the algorithm, subsequent tests were conducted to replicate the results presented in the guiding article (Volkhonskiy *et al.*, 2017a). This section specifically aims to reproduce the figures related to the method's validity. The replication process was challenging, as the authors did not specify the exact betting functions used to generate the plots or provide the seed for their synthetic data. As a result, achieving an exact match between the graphs proved impossible, given that the martingale behavior can vary significantly depending on the seed used for data generation.

- **Figure 1 - Behavior of Circumcised ICM vs. Original ICM**

  Two sets of 1,000 data points are generated: one without a change point and another with a change point, aiming to demonstrate martingale behavior when the data remains i.i.d. In both cases, an additional set of 200 data points is generated from a normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$ to serve as the initial training data for the algorithm. In the dataset with a change point, the first 500 observations follow a normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$, while the subsequent 500 observations follow $\mathcal{N}(\mu = 1.5, \sigma = 1)$. The 7 Nearest Neighbors NCM is used as specified by the authors, but the results are generated using both the Constant BF and the Precomputed Kernel Density BF to highlight the significant differences between them. To illustrate the necessity of the circumcised version for faster change detection, both the original ICM (generated by equation (13)) and the circumcised ICM (generated by equation (14)) are plotted for the data with the change point. The resulting plots are presented in Figure 3.

  Although achieving an exact match with the article's figures is impossible, a clear similarity can be observed when comparing Figure 3 to Figure 1 of the guiding article (Volkhonskiy *et al.*, 2017a). The plot obtained with the precomputed kernel density betting function exhibits a behavior more closely resembling the original, suggesting that this might have been the BF used by the authors.

In general, Figure 3 highlights how the original ICM introduces negative values, which hinders fast change point detection. Thus, the circumcised ICM is preferred. Also, this figure shows that the martingale correctly stays below the threshold when the data has no change point.

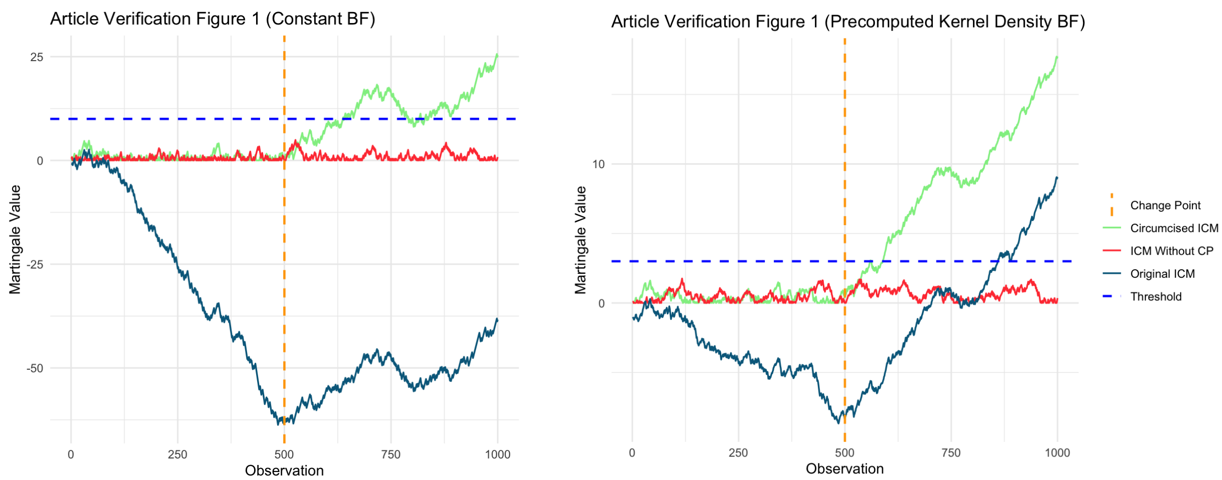

Figure 3

- **Figure 1 (Version 2) - ICM Behavior With Different Non-Conformity Measures**

  A second version of the guiding article (Volkhonskiy *et al.*, 2017b) was found to have a different plot as Figure 1, so this version was also replicated. The same data with and without a change point was generated as described in the previous item. The mixture BF was used, and the circumcised ICM was plotted for different types of non-conformity measures to compare their behavior. The original plot in the article only includes the LR NCM and the 1NN NCM. However, since a rolling version of the LR NCM was also implemented in this project (as described in Section 4.1), the plot was created using all three NCMs, as shown in Figure 4.

  As with the previous figure, achieving an exact match is not possible, but an overall similarity is observed between Figure 4 and Figure 1 of the guiding article (Volkhonskiy *et al.*, 2017b), indicating that the algorithm and non-conformity measures were implemented correctly. The results show that the LR NCM produces higher martingale values and faster detection, while the rolling window version exhibits a similar but less pronounced pattern, suggesting that effective detection can still be achieved even without precise knowledge of the post-change mean. Notably, the 1NN NCM also performs well, achieving nearly the same detection speed without relying on parametric assumptions.

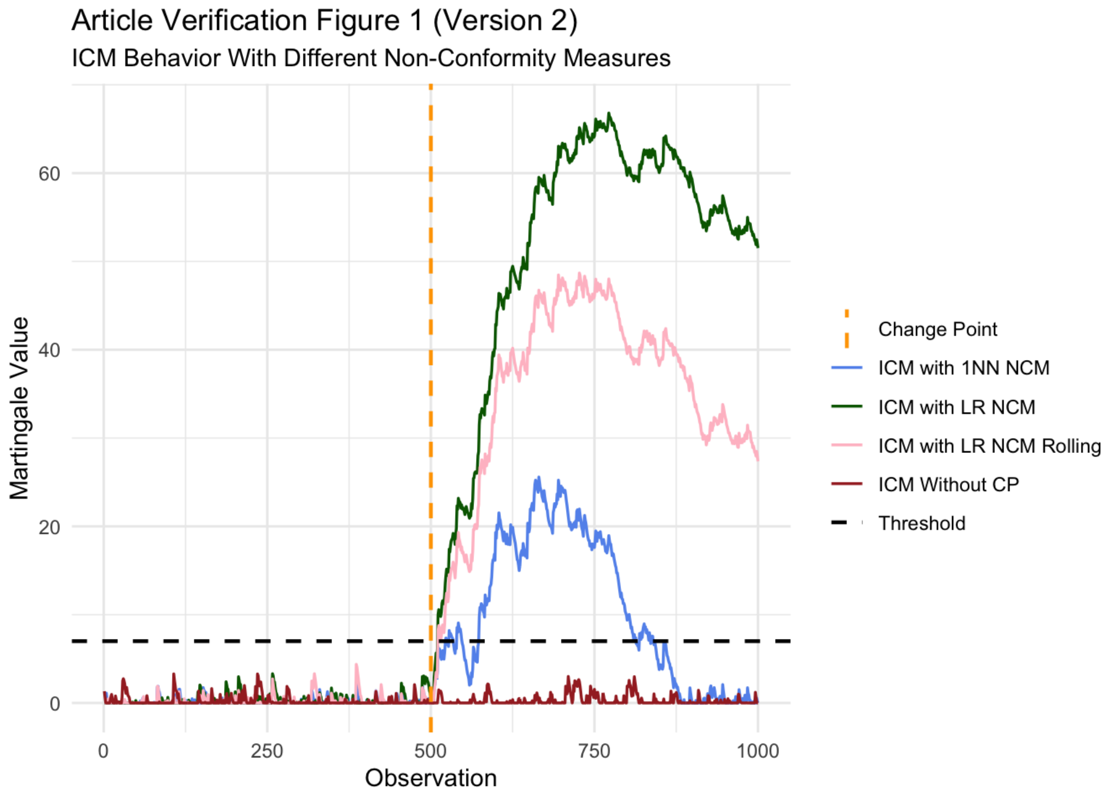

Figure 4

- **Figure 2: Validity of ICM Algorithm with Small Train Sets**

  A set of 1,500 observations without a change point is generated from a normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$. The test set for the algorithm, drawn from the same distribution, varies in size with $m \in \{1, 2, 3, 4, 5\}$. The resulting martingale is plotted for all five cases using the Mixture BF and the K Nearest Neighbors NCM, where $k = \lceil m/2 \rceil$. For comparison, a dataset with a change point and test size $m = 1$ is also included, following the same structure described in the first item of this subsection. The complete results are shown in Figure 5.

  Figure 5 shows a behavior very similar to Figure 2 of the guiding article (Volkhonskiy *et al.*, 2017a). However, it is important to note that this behavior is not consistent and can vary significantly depending on the seed used for data generation (in this case, seed 234). While this example demonstrates that, for small test sizes, the martingale remains bounded below the threshold in data without a change point and can clearly detect a change point even with just one data point as a test set, this may not always hold true. In particular, the observed behavior may differ when the data does not follow distributions as strictly defined as in this synthetic example.

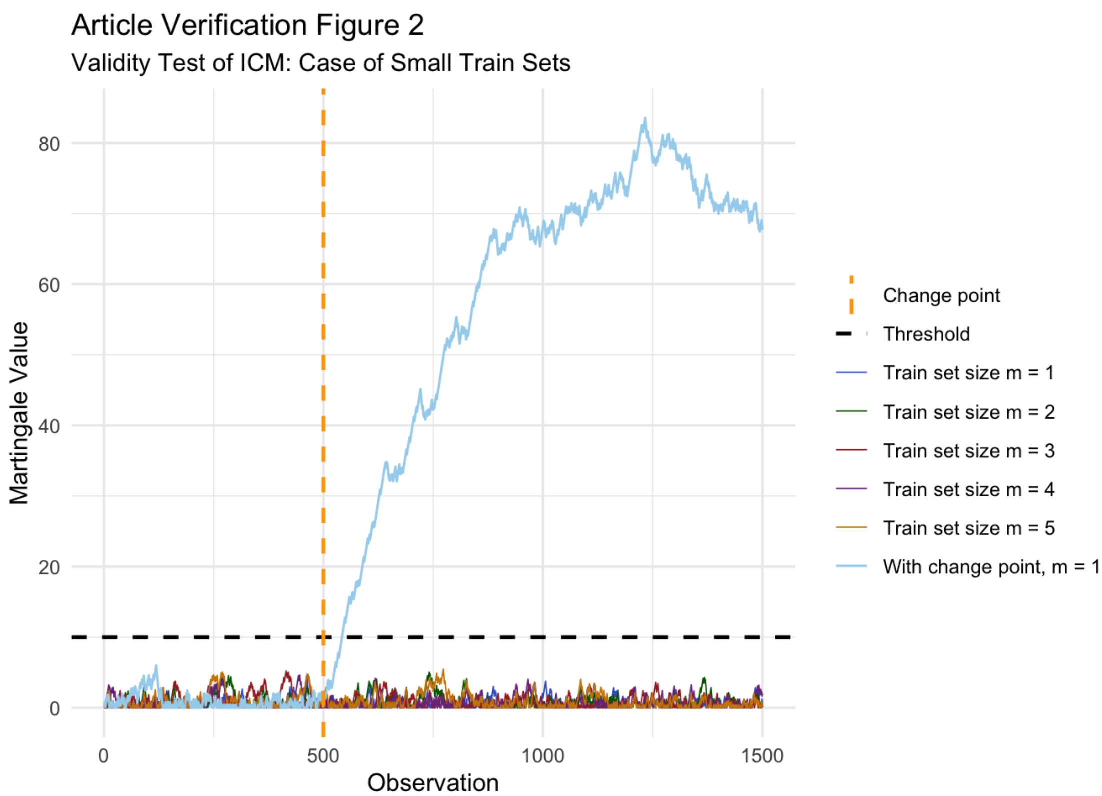

Figure 5

5.3 Evaluation of Algorithm Performance

The methodology outlined in Section 4.2 is implemented to systematically evaluate the algorithm's performance in terms of mean delay and false alarm probability. Each experiment generates 1,000 observations, with pre-change data following a normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$ and post-change data following $\mathcal{N}(\mu = \mu_1, \sigma = 1)$. Six cases are tested for each betting function, varying the change point $\theta \in \{100, 200\}$ and $\mu_1 \in \{1, 1.5, 2\}$. Each run uses a test set of size $m = 200$ drawn from the pre-change distribution. For each betting function, the experiment explores a range of threshold values, conducting 100 trials for each value to calculate mean delay and false alarm probability. The resulting plots display the log mean delay against the false alarm probability for better visualization. The results for the Constant BF, Mixture BF, and Precomputed Kernel Density BF are shown in Figures 6, 7, and 8, respectively.

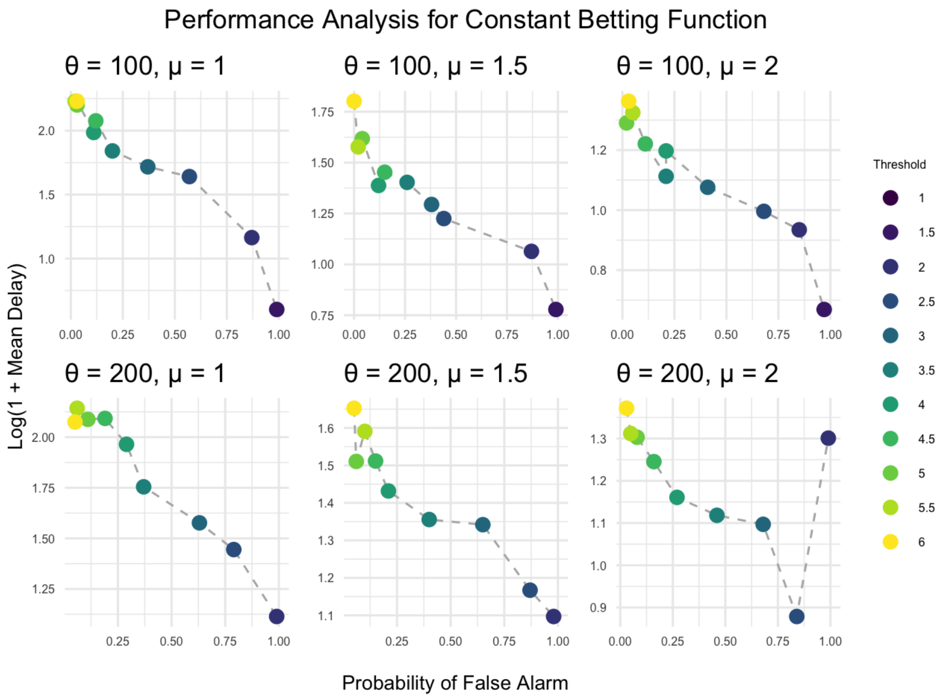

Figure 6

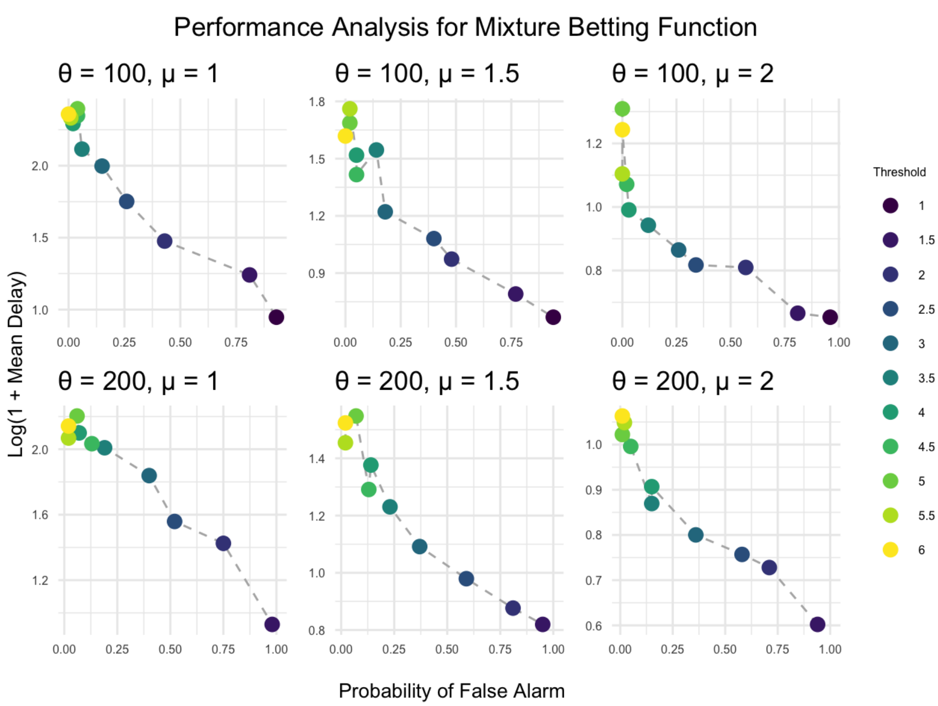

Figure 7

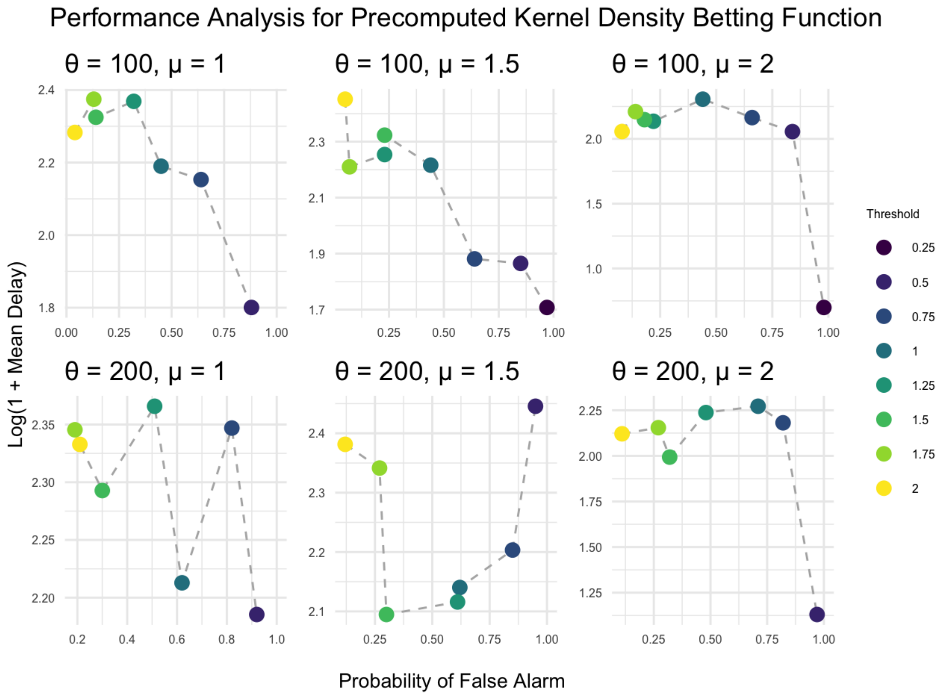

Figure 8

The results for the Constant BF and Mixture BF align well with theoretical expectations, showing an inverse relationship between mean delay and false alarm probability. As depicted in Figures 6 and 7, reducing the threshold decreases the mean delay but increases the probability of false alarms. This behavior is consistent with previous observations: a lower threshold allows the algorithm to detect change points more quickly but also increases the likelihood of false detections. Conversely, higher thresholds result in longer mean delays (as the martingale takes more time to exceed the value of h) but reduce the probability of false alarms. The resulting graphs form smooth, decaying curves in almost all six cases, with smoother transitions expected if more trials and threshold values are considered. Notably, the mean delay is higher when $\mu_1 = 1$, as the smaller change in the mean makes detection more challenging. Additionally, all graphs show a sharp decrease in mean delay when the threshold drops from 6 to 4, without a corresponding increase in false alarm probability. This suggests that an optimal threshold likely lies around this region, where the curve changes concavity (approximately $h = 3$ for both betting functions), balancing detection speed and accuracy. However, the choice of threshold ultimately depends on the desired trade-off between these metrics.

The Precomputed Kernel Density BF exhibits less consistent behavior, as shown in Figure 8. While most cases display a general decreasing trend, the curves are less smooth and occasionally show abrupt jumps. This inconsistency may arise from the variability of the kernel density estimation process, which is highly sensitive to the distribution of p-values of the training set. For this betting function, a threshold of $h = 1$ appears to have a good balance between detection speed and false alarm probability in all cases, as it is located near the point where the curves change concavity.

It is important to note that the range of threshold values for each experiment was chosen empirically, based on a preliminary analysis of the martingale behavior for the generated normal data. This highlights the challenge of selecting an appropriate threshold to ensure the correct functioning of the algorithm, as it is highly dependent on the data characteristics and the chosen betting function.

The experimental setup for this section is inspired by the experiments described in the guiding article (Volkhonskiy *et al.*, 2017a) in Section 5. However, the article lacks clarity regarding the specific threshold values used to generate the curves for mean delay and false alarm probability, or whether they vary another parameter altogether. As a result, although the graphs presented in Figures 6, 7, and 8 exhibit similar patterns to the ICM curves shown in Figures 5, 6, and 8 of the guiding article, a direct comparison is not feasible without complete information about their experimental configuration.

5.4 Comparison with Traditional Methods

In order to accurately evaluate the performance of the implemented ICM algorithm, it is compared against the performance of well-established change point detection methods. Specifically, it is compared against three non-parametric algorithms implemented in the `tidychangepoint` library in R (Baumer *et al.*, 2024):

- **PELT (Pruned Exact Linear Time):** Minimizes a cost function to identify multiple change points. It uses a pruning technique to reduce computational complexity, achieving linear time performance in many scenarios. Well-suited for detecting changes in the mean or variance of large datasets (Gachomo, 2015).
- **Genetic Algorithm:** The genetic algorithm used corresponds to Coen's algorithm, as described in Taimal *et al.* (2024). It applies evolutionary principles such as selection, crossover, and mutation to optimize the selection of change points.
- **Random Basket of Change Points:** This method is equivalent to running the genetic algorithm with only one generation, effectively selecting a random set of candidate change points (Baumer *et al.*, 2024).

5.4.1 Graphical Comparison

An initial graphical comparison is done to gain insight into the performance of the four algorithms. Three representative examples are presented:

- **Normal distribution before and after the change point**: The experimental setup follows the same procedure described in the first item of Section 5.1. For the `tidychangepoint` algorithms, the test set is excluded from the data. The detected change points obtained with the four algorithms are shown in Figure 9.

PELT obtains the best detection, identifying the change point exactly. The ICM algorithm also performs well, detecting the change with a delay of 7 observations. The genetic algorithm, executed with a maximum of 10 iterations, identifies a false change point, although it is relatively close to the actual location. The random algorithm results in a clear false alarm.

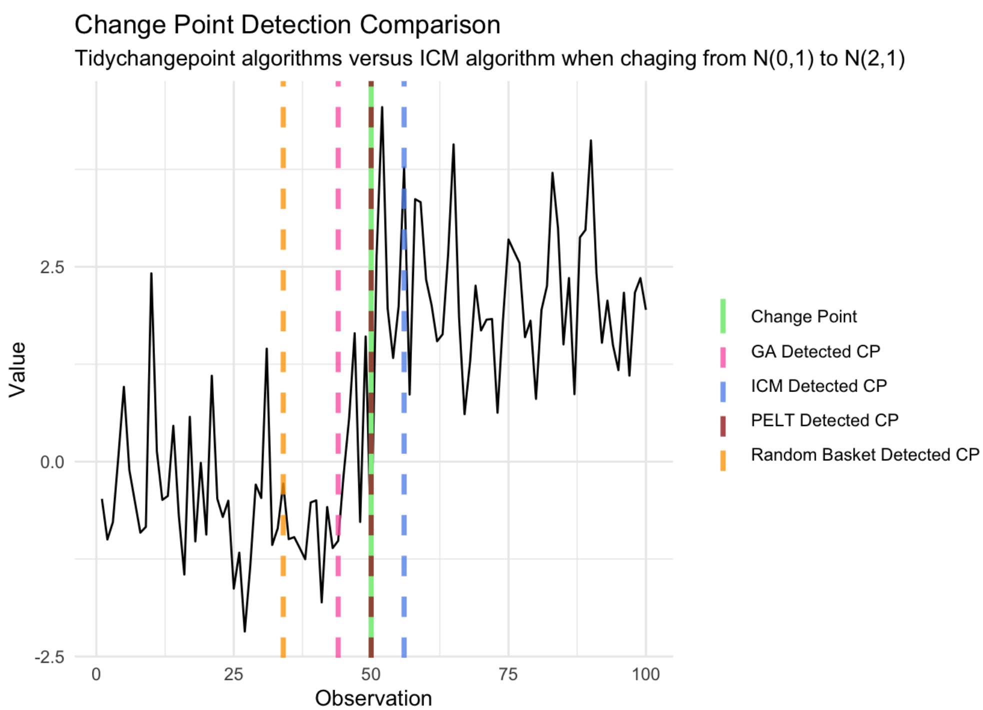

Figure 9

- **Uniform distribution before and exponential distribution after the change point**: The experimental setup follows the same procedure described in the second item of Section 5.1. For the `tidychangepoint` algorithms, the test set is excluded from the data. The detected change points obtained with the four algorithms are shown in Figure 10.

  In this case, the change-point detected by PELT and ICM differs by only one observation, highlighting the strong performance of the new implemented algorithm. The genetic algorithm fails to detect a change-point, as indicated by the absence of the pink dashed line in Figure 10, while the random algorithm once again produces a false detection. These results demonstrate the effectiveness of the ICM algorithm, as it performs nearly as well as PELT—one of the most reliable CP detection methods in the literature—while outperforming both heuristic and entirely random methods.

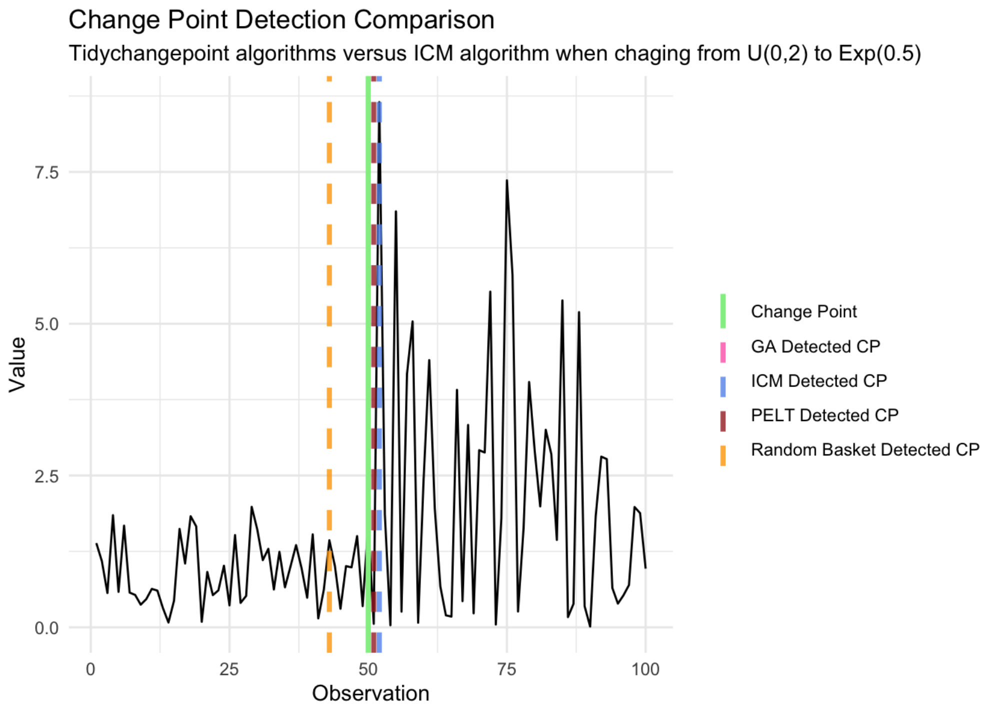

Figure 10

- **Baseball Data**: The performance of the four algorithms is also preliminarily evaluated using real data corresponding to the difference in home runs per plate appearance between the American League and the National League from 1925 to 2023. The change point is known to be 1973, as the introduction of the designated hitter rule in the American League that year led to a significant increase in home run rates compared to the National League (Baumer & Suárez Sierra, 2024). For this analysis, the ICM algorithm is configured with the 1 Nearest Neighbor NCM, the Mixture BF, a threshold of 2.5, and a test set made up of the first 5 observations. The genetic algorithm is executed with a maximum of 50 iterations.

  Figure 11 shows the change points detected by the four algorithms. Despite performing well on synthetic data, PELT detects a false change point in 1946. The random algorithm also produces a false detection in an even earlier year. The genetic algorithm, executed with 50 iterations, fails to detect any change point.

  The ICM algorithm demonstrates the best performance, detecting the change point in 1982, with a delay of 9 years. Although earlier detection would be preferable, Figure 12 illustrates why this is not possible with the chosen configuration. Lowering the threshold to achieve earlier detection would result in a false alarm around 1965. This outcome highlights that, while the implemented ICM algorithm is effective and can outperform other established methods, its performance is highly sensitive to the chosen threshold and betting function.

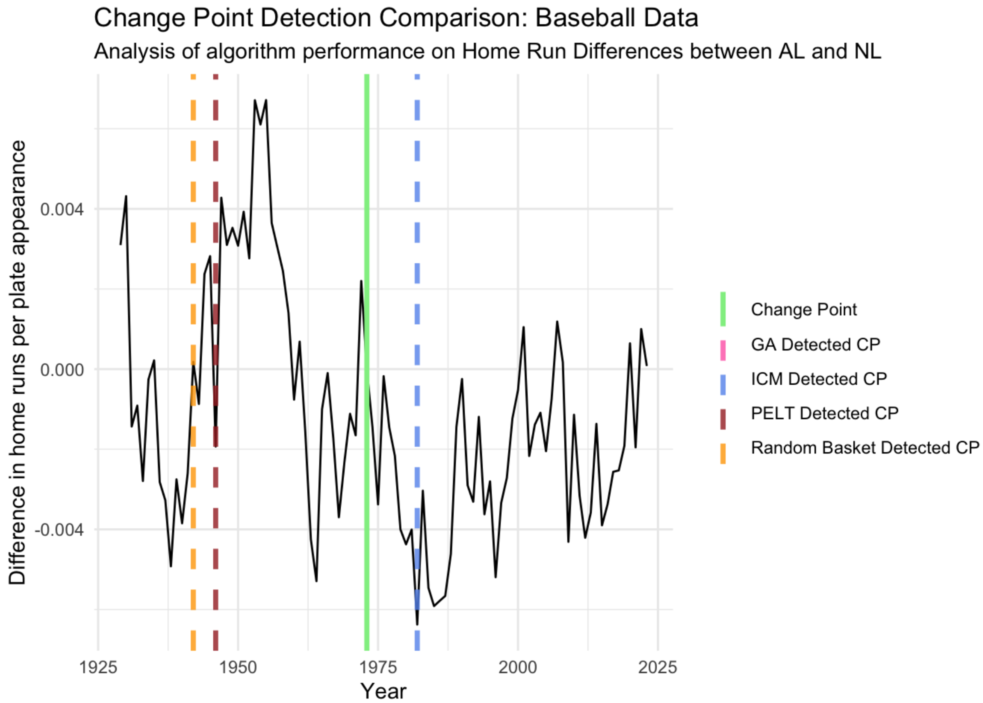

Figure 11

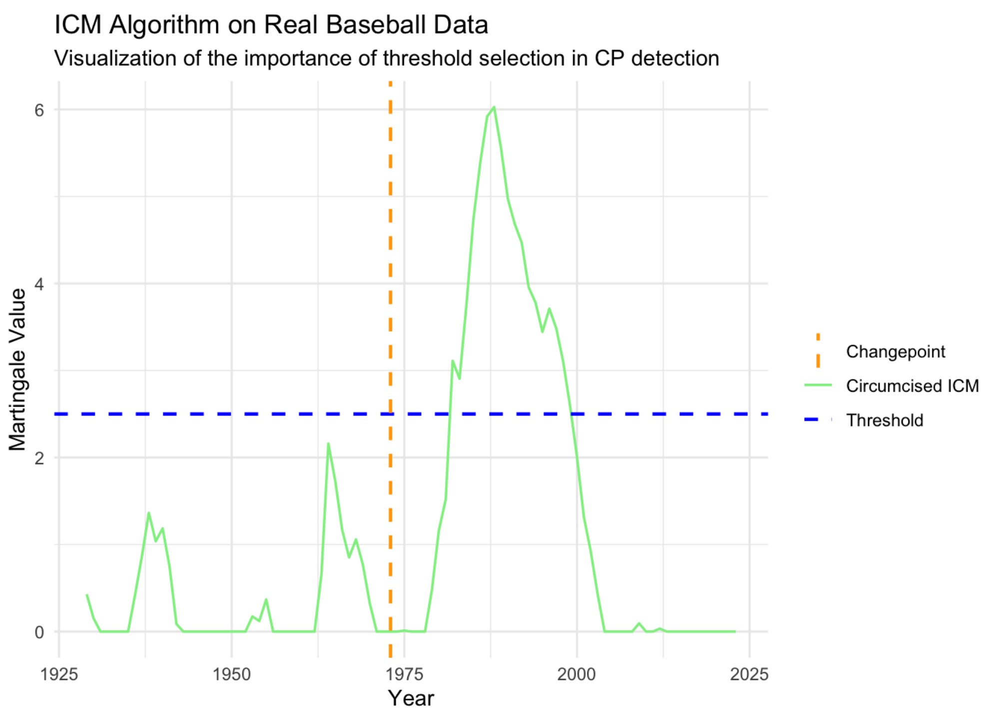

Figure 12

5.4.2 Systematic Comparison

Finally, the performance of the ICM algorithm is compared with the three `tidychangepoint` algorithms in terms of mean delay and false alarm probability. The comparison follows the same experimental setup described in Section 5.3, but focuses on two representative configurations: $\theta = 200$, $\mu_1 = 2$ (Experiment 1) and $\theta = 100$, $\mu_1 = 1$ (Experiment 2). Experiment 1 features an early change point with a small change in mean, while Experiment 2 involves a later change point with a larger change in mean. For each algorithm, 100 trials are conducted under the specified configuration, and the key performance metrics are calculated. The average computation time of each algorithm is included as an additional metric for comparison. The ICM algorithm is executed using the three implemented betting functions, and the best thresholds (in terms of balance between the two metrics) are determined using the graphs presented in Section 5.3. The results obtained for Experiments 1 and 2 are shown in Tables 1 and 2, respectively.

Method	**Mean Delay**	**False Alarm Probability**	**Average Time (s)**
PELT	2.222	0.21	0.002
Random Basket	6.214	0.68	0.047
Genetic Algorithm (`maxiter = 10`)	8.000	0.69	0.969
ICM with Constant BF ($h = 3.5$)	12.365	0.48	0.315
ICM with Mixture BF ($h = 4$)	7.907	0.14	0.326
ICM with Precomputed Kernel Density BF ($h = 1$)	115.448	0.67	0.308

Table 1: Comparison of Change-Point Detection Methods ($\theta = 200$, $\mu_1 = 2$)

Method	**Mean Delay**	**False Alarm Probability**	**Average Time (s)**
PELT	5.571	0.30	0.002
Random Basket	25.237	0.57	0.046
Genetic Algorithm (`maxiter = 10`)	878.000	0.01	0.954
ICM with Constant BF ($h = 3$)	52.015	0.33	0.312
ICM with Mixture BF ($h = 2.5$)	76.449	0.30	0.324
ICM with Precomputed Kernel Density BF ($h = 1$)	149.151	0.41	0.304

Table 2: Comparison of Change-Point Detection Methods: ($\theta = 100$, $\mu_1 = 1$)

In Experiment 1 (Table 1), where the change in mean is more pronounced and occurs later, the ICM algorithm with the Mixture BF performs well, achieving a mean delay of 8 observations and a false alarm probability of 0.14. Although PELT obtains a mean delay of just 2 observations, the ICM's performance with the Mixture BF is competitive, as it achieves a lower false alarm rate. The Genetic Algorithm and Random Basket Algorithm both exhibit higher false alarm probabilities despite similar mean delays, indicating that the ICM with the Mixture BF outperforms both. The Constant BF shows a high false alarm rate, while the Precomputed Kernel Density BF has an excessive mean delay (over 100 observations), making them less suitable in this context. Overall, the Mixture BF proves to be the most effective betting function for ICM in this scenario, although fine-tuning the threshold could further improve the balance between mean delay and false alarms.

In Experiment 2 (Table 2), where the change point is early and the change in mean is smaller, PELT demonstrates clear superiority in terms of mean delay. The ICM algorithm, regardless of the betting function, shows high mean delays and fails to obtain a lower false alarm rate than PELT's. Among the heuristic methods, the Genetic Algorithm performs worst, detecting the change point almost at the end of the series but rarely producing false alarms (0.01 probability). The Random Basket method detects change points faster but at the cost of a high false alarm rate. In this case, the ICM can be said to perform similarly to the Random Basket but with opposite behaviors in the key metrics, indicating that the algorithm struggles with early, subtle changes.

In terms of computation time, both experiments exhibit similar results. As expected, PELT produces results the fastest, as linear time complexity is a key feature of its design. The random algorithm also runs relatively quickly, but its performance is generally poor. The genetic algorithm takes the longest time, as it requires iterative optimization, and it does not obtain good results. The ICM algorithm has an average computation time of 0.3 seconds across all three betting functions. This time is approximately one-third of that required by the genetic algorithm while yielding substantially better results. Given that the ICM method processes data online and handles observations sequentially (with 1,000 observations in the tested dataset), a computation time of 0.3 seconds is acceptable.

In summary, the results indicate that ICM is a competitive alternative to traditional change-point detection methods, particularly when using the Mixture BF and data with pronounced changes. Its online nature provides a distinct advantage, as it does not require the entire time series to perform detection, unlike the non-parametric methods tested.

6 Conclusions and Future Research

This study focused on the computational implementation and validation of a non-parametric change point detection method based on Inductive Conformal Martingales, as presented by Volkhonskiy *et al.* (2017a). The primary objective was to reproduce the results from the guiding article to verify the validity of the method while also creating an open-source repository containing the necessary functions for implementing the algorithm. This initiative aims to enhance the reproducibility and accessibility of the ICM approach. The results obtained demonstrate that the ICM method is indeed valid and effective for change point detection, supporting various choices of non-conformity measures, betting functions, and even small test sizes.

The experiments conducted systematically evaluated the algorithm's performance with a focus on mean delay and probability of false alarm. The results are promising, indicating that the ICM algorithm can effectively detect change points. However, they also reveal that achieving a balance between mean delay and false alarm probability is highly dependent on the choice of betting function and the selected threshold.

The performance of the implemented ICM algorithm was also compared with well-established change-point detection methods. The graphical comparison demonstrated that the ICM algorithm is competitive, detecting change points almost as accurately as PELT in many cases. Notably, when applied to real-world data, the ICM algorithm outperformed the other methods, highlighting its practical applicability. The systematic comparison in terms of mean delay and false alarm probability revealed that ICM performs comparably to PELT and significantly better than heuristic methods, demonstrating its robustness in both synthetic and real data scenarios.

Overall, the ICM algorithm shows strong potential as a reliable change-point detection tool, particularly due to its lack of dependence on parametric assumptions and its ability to perform online detection. These characteristics make it highly suitable for real-time applications where detecting changes as they occur is crucial. Despite these positive results, further research is needed to fully establish the robustness and versatility of the ICM method. Most of the tests in this study were conducted using synthetic normal data, following the experimental setup proposed in the guiding article. Therefore, it is necessary to expand the analysis to other data distributions and to evaluate the algorithm with diverse real-world datasets. Additionally, the sensitivity of the ICM algorithm to the threshold parameter h needs deeper investigation. Currently, it remains unclear how to select an optimal threshold when aiming for online detection, as it is impractical to visualize the entire martingale behavior beforehand.

Further extensions of the method could involve adapting the ICM algorithm for multiple change-point detection and multivariate time series, broadening its applicability in more complex scenarios. Additionally, it is essential to test the ICM algorithm on hydroclimatic time series, given that its computational implementation was done mainly to contribute to the ongoing research initiative at Universidad EAFIT, led by Alejandra María Carmona Duque and Biviana Marcela Suárez Sierra. Demonstrating the algorithm's effectiveness in this context would significantly advance the project and confirm the ICM's utility for real-time environmental monitoring.

In conclusion, while the implemented ICM algorithm demonstrates strong potential as a reliable and flexible change point detection method, further research and adaptation are essential to optimize its performance across a wider range of real-world applications.

Acknowledgements

We express our gratitude to Cristhian Quiroz Castaño, a master's student in Applied Mathematics at Universidad EAFIT, for his invaluable support throughout this project. His guidance and support was crucial in understanding the most challenging theoretical aspects of the implemented method and in addressing the computational challenges that arose during the development process.

References

Aminikhanghahi, Samaneh, & Cook, Diane. 2017. A Survey of Methods for Time Series Change Point Detection. *Knowledge and Information Systems*, **51**(05).

Baumer, Benjamin, & Suárez Sierra, Biviana Marcela. 2024 (July). *tidychangepoint: a unified framework for analyzing changepoint detection in univariate time series.*

Baumer, Benjamin S., Sierra, Biviana Marcela Suárez, Coen, Arrigo, & Taimal, Carlos A. 2024. *tidychangepoint: A Tidy Framework for Changepoint Detection Analysis.* R package version 0.0.1.

Behera, Swadhin Kumar. 2024. Understanding the impact of climate change on extreme events. *Frontiers in Science*, **2**.

Berger, Vance, & Zhou, YanYan. 2014. *Kolmogorov–Smirnov Test: Overview.*

Chen, Guangyuan, Lu, Guoliang, Shang, Wei, & Xie, Zhaohong. 2019. Automated Change-Point Detection of EEG Signals Based on Structural Time-Series Analysis. *IEEE Access*, **7**, 180168–180180.

Eliades, Charalambos, & Papadopoulos, Harris. 2021. Using inductive conformal martingales for addressing concept drift in data stream classification. *Pages 171–190 of: Conformal and Probabilistic Prediction and Applications.* PMLR.

Fedorova, Valentina, Gammerman, Alex, Nouretdinov, Ilia, & Vovk, Vladimir. 2012. Plug-in martingales for testing exchangeability on-line. *Pages 1639–1646 of:* Langford, John, & Pineau, Joelle (eds), *Proceedings of the 29th International Conference on Machine Learning (ICML-12).* Omnipress.

Gachomo, Dorcas. 2015. The Power of the Pruned Exact Linear Time (PELT) Test in Multiple Changepoint Detection. *American Journal of Theoretical and Applied Statistics*, **4**(12), 581.

Gallagher, Colin, Lund, Robert, & Robbins, Michael. 2013. Changepoint Detection in Climate Time Series with Long-Term Trends. *Journal of Climate*, **26**(14), 4994 – 5006.

Ho, Shen-Shyang. 2005. A martingale framework for concept change detection in time-varying data streams. *Page 321–327 of: Proceedings of the 22nd International Conference on Machine Learning.* ICML '05. New York, NY, USA: Association for Computing Machinery.

Kamal, Neel, & Pachauri, Sanjay. 2018. Mann-Kendall Test - A Novel Approach for Statistical Trend Analysis. *International Journal of Computer Trends and Technology*, **63**(09), 18–21.

Milly, P. C. D., Betancourt, Julio, Falkenmark, Malin, Hirsch, Robert M., Kundzewicz, Zbigniew W., Lettenmaier, Dennis P., & Stouffer, Ronald J. 2008. Stationarity Is Dead: Whither Water Management? *Science*, **319**(5863), 573–574.

Niranjan Pramanik, Ph.D. 2019. *Kernel Density Estimation.*

Page, E. S. 1954. Continuous Inspection Schemes. *Biometrika*, **41**(1/2), 100–115.

Pettitt, A. N. 1979. A Non-Parametric Approach to the Change-Point Problem. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **28**(2), 126–135.

Shiryaev, A. N. 1963. On Optimum Methods in Quickest Detection Problems. *Theory of Probability & Its Applications*, **8**(1), 22–46.

Taimal, Carlos A., Suárez-Sierra, Biviana Marcela, & Rivera, Juan Carlos. 2024. An Exploration of Genetic Algorithms Operators for the Detection of Multiple Change-Points of Exceedances Using Non-homogeneous Poisson Processes and Bayesian Methods. *Pages 230–258 of:* Tabares, Marta, Vallejo, Paola, Suarez, Biviana, Suarez, Marco, Ruiz, Oscar, & Aguilar, Jose (eds), *Advances in Computing*. Cham: Springer Nature Switzerland.

Volkhonskiy, Denis, Nouretdinov, Ilia, Gammerman, Alexander, Vovk, Vladimir, & Burnaev, Evgeny. 2017a. *Inductive Conformal Martingales for Change-Point Detection.*

Volkhonskiy, Denis, Burnaev, Evgeny, Nouretdinov, Ilia, Gammerman, Alexander, & Vovk, Vladimir. 2017b. Inductive Conformal Martingales for Change-Point Detection. *Pages 132–153 of:* Gammerman, Alex, Vovk, Vladimir, Luo, Zhiyuan, & Papadopoulos, Harris (eds), *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*. Proceedings of Machine Learning Research, vol. 60. PMLR.

Vovk, Vladimir. 2021. Testing Randomness Online. *Statistical Science*, **36**(4), 595 – 611.

Vovk, Vladimir, Nouretdinov, Ilia, & Gammerman, Alex. 2003. Testing exchangeability on-line. *Page 768–775 of: Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. ICML'03. AAAI Press.

Vovk, Vladimir, Gammerman, Alex, & Shafer, Glenn. 2005. *Algorithmic Learning in a Random World*. Springer.

Wang, Gang, Lu, Zhiying, Wang, Ping, Zhuang, Shuo, & Wang, Di. 2023. Conformal Test Martingale-Based Change-Point Detection for Geospatial Object Detectors. *Applied Sciences*, **13**(15).

World Meteorological Organization. 2021. *WMO Atlas of Mortality and Economic Losses from Weather, Climate and Water Extremes (1970–2019)*. Tech. rept.

Zhu, Xiaoqian, Xie, Yongjia, Li, Jianping, & Wu, Dengsheng. 2015. Change point detection for subprime crisis in American banking: From the perspective of risk dependence. *International Review of Economics Finance*, **38**, 18–28.