MDPI

Article

Title

Allan Bolanos Barrientos [1], Martín Solís [2] and Firstname Lastname [2,*]

[1] Costa Rica Institute of Technology. School of Computer Engineering; a.bolanos.2@estudiantec.cr
[2] Costa Rica Institute of Technology; marsolis@itcr.ac.cr
* Correspondence: marsolis@itcr.ac.cr

Abstract

Online change point detection is critical for real-time monitoring across diverse domains, yet algorithm performance under varying noise levels and change magnitudes remains poorly understood. This study presents the first comprehensive benchmark evaluating 17 state-of-the-art online change point detection algorithms across systematically controlled conditions. We assess performance on 360 synthetic time series spanning 8 scenarios that combine noise levels (low/high), change magnitudes (low/high), and change types (step/slope), plus 49 manually-labeled real-world crime occurrence series from Costa Rica. Results reveal substantial performance heterogeneity: state-space models (SSM-Canary, TAGI-LSTM) achieve near-perfect detection (F1 > 0.95) in low-noise scenarios but fail catastrophically in high-noise conditions (F1 < 0.25), while statistical tests (Two-Sample Tests, Gaussian Segmentation) maintain consistent moderate performance (F1 = 0.30-0.50) across all noise levels. Transfer learning analysis exposes critical domain adaptation challenges, with all algorithms experiencing significant performance degradation on real crime data (average F1 drop: 0.28 points, 45% decrease). State-space methods suffer the most severe collapse, while distribution-free approaches prove more robust. Our findings demonstrate that synthetic benchmark rankings poorly predict real-world utility, emphasizing context-dependent algorithm selection. We provide actionable deployment guidelines matching algorithm characteristics to application requirements (precision vs. recall priorities, detection speed, noise tolerance) and release the labeled crime dataset and complete benchmark codebase to enable reproducible research and practical algorithm comparison.

Keywords: online change point detection; algorithm benchmarking; time series analysis; noise robustness; transfer learning; crime data; synthetic evaluation; domain adaptation; real-time monitoring

Received:
Revised:
Accepted:
Published:

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Journal Not Specified* **2025**, *1*, 0. https://doi.org/

1. Introduction

Change point detection is the finding of changes in the behavior pattern in a time series. The interest in finding change points can be online or offline. Online algorithms process data within a sliding window with size n to detect change points in real, while offline the entire time series is processed at once to detect the change points in the complete time series [1]. Online change detection has increased its relevance due to the rapid growth of stream data generation in diverse domains [2]. Nowadays the development of algorithms in online detection has contributed to relevant important issues such as fuel leakage detection [3], detection of change points in the spread of viruses to reveal the effectiveness of interventions [4], detect pipe burst localization in water distribution [5], etc.

Change point detection methods could perform poorly when the data are noisy [6] and make detection points challenging [7]. This problem becomes greater when the changing points are subtle [3], even for humans. For example, in figure xx it can be seen the difficulty to the change point detection for the same time series with different levels of noise. Besides, the presence of noise can lead to the detection of false change points, when what there is noise. Although noise can influence the performance of change detection algorithms, there is no benchmarking of how the algorithms perform in the presence of different levels noise levels in contexts with weak and strong change points, as far as we know. Therefore, the present research analyzes how the more common online change detection algorithms perform in the presence of noise and different levels of change. The findings obtained can lead to a better choice of detection methods according to the type of time series under study, as well as opening new research questions on how to improve the detection of change points. As a second objective, this research will address a real problem such as the task of change points detection in time series of crimes occurrences. Although this task can contribute to monitoring crime, detecting regions where the pattern of crime changes, and establishing police strategies, it has received little attention on the research [8]. On the other hand, the time series of crimes tend to show relevant levels of noise which can make it difficult to detect change points. Crime time series depend on the citizen's willingness to report the crime. Thus, noise does not only come from randomness of the phenomena but from a structural bias in the collection of information.

In summary the novel contributions of the manuscript are the next:

- A novel performance benchmarking of the online change point detection algorithms based on the level of noise and strength of the change point.
- Analysis of the change point detection in crime time series according to the level of noise.
- A new labeled dataset to detect change points in crime time series .

2. Literature review

Recent studies have addressed the issue of online change point detection. In [9] six commonly available state-of-the-art changepoint detection approaches, with two additional modified algorithms, were compared in cardiovascular time series data. In [10] a benchmarking of popular algorithms was generated using simulated data and 37 real time series from various application domains. [11] evaluated online changepoint detection algorithms that work on unbounded data stream with a constant time and space complexity. Other authors, instead of comparing known methods in different contexts, developed a new proposal for online change point detection. These are the cases of [3], [12], [7]. In [3], the authors proposed a novel memory-based online change point detection (MOCPD) framework to find fuel leakage detection delays. [7] create a method called Delta point that divides the time series into intervals of user-specified, domain specific length for which a suspected change point may be contained. In [12] the method created does not require any prior distributional knowledge of the time series and exploits the Information Gain to verify each new candidate change point score. In the studies where a benchmarking of methods is generated, neither in the studies where new proposals are created are there evaluations of the algorithm's performance according to the level of noise and the level of change point. Therefore, there is a lack of knowledge about how algorithms work in specific circumstances. In relation to the detection of changes in crime events patterns, only a few studies have analyzed its effectiveness, although the application of change point detection algorithms can contribute to the police monitoring of criminality. This is the case of [8] who investigate the effectiveness of both online and offline change point detection methods towards identifying critical changes in crime-related time series from Boston' and

the 'London Police Records' datasets. The best performance is obtained with BOCPD. In [13] the authors proposed a fuzzy approach to detect change points in the CICOP data set (crime real-world data), that consisting of 32 monthly time series. They conclude that the proposal method has great potential in crime analysis, however there are not comparisons with baselines or other methods. The work of [14] consists in a proposal towards detecting change points in terrorism-related time series. They conclude that the proposed framework could be seen as an alternative way to identify links between terrorism and online activity This section describes the comprehensive methodology employed to benchmark online change point detection algorithms. The experimental design comprises three complementary evaluation approaches: (1) controlled experiments using synthetic time series with known change points, (2) real-world evaluation using manually labeled crime occurrence data, and (3) benchmark evaluation on the TCPD (Time Series Change Point Database) repository.

3. Materials and Methods

This section describes the experimental framework designed to evaluate online change point detection algorithms under three complementary evaluation paradigms: synthetic data generation, real-world labeled data, and transfer learning validation. Our methodology follows a rigorous train-test protocol with comprehensive hyperparameter optimization and multi-metric evaluation.

3.1. Experimental Framework Overview

The evaluation framework consists of three interconnected benchmarking pipelines:

1. **Benchmark 1: Synthetic Data Evaluation** – Controlled environment with known ground truth, enabling systematic assessment across noise levels, change magnitudes, and change types.
2. **Benchmark 2: Real-World Data Evaluation** – Manual labeled crime statistics from Costa Rica, providing domain-specific validation.
3. **Benchmark 3: Transfer Learning Validation** – Investigates parameter transferability from synthetic to real domains.

All benchmarks employ identical evaluation metrics and train/test methodology to ensure fair comparison.

3.2. Datasets

3.2.1. Synthetic Time Series (Benchmark 1)

We generate synthetic time series with controlled characteristics to systematically evaluate algorithm performance under various conditions.

Time Series Generation

Each synthetic series is generated using an autoregressive process with injected change points:

$$x_t = \phi x_{t-1} + \mu_s + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

where:

- $\phi = 0.3$ (autoregressive coefficient)
- μ_s is the segment mean (changes at change points)
- σ controls noise level

Experimental Design

We employ a full factorial design covering:

- **Noise Levels:**
  - Low: $\sigma \in [0.0, 0.4]$ (high SNR)
  - High: $\sigma \in [3.0, 6.0]$ (low SNR)
- **Change Magnitudes:**
  - Low: $|\Delta\mu| \in [0.5, 1.5]$ standard deviations
  - High: $|\Delta\mu| \in [3.0, 6.0]$ standard deviations
- **Change Types:**
  - Step (escalón): Abrupt mean shift
  - Slope (pendiente): Gradual linear trend change
- **Series Characteristics:**
  - Length: $L \in \{200, 300, 400\}$ time steps
  - Number of change points: $n_{cp} \in \{1, 2, 3, 4\}$

This yields $2 \times 2 \times 2 = 8$ unique scenario combinations. For each scenario, we generate 45 series (3 iterations $\times$ 15 series per iteration), resulting in **360 total series**.

Train-Test Split

Series are randomly partitioned into:

- Training set: 70% (252 series) – used for hyperparameter optimization
- Test set: 30% (108 series) – used for final evaluation

Random seed is fixed ($seed = 123$) to ensure reproducibility.

3.2.2. Real-World Crime Data (Benchmark 2)

We used manually labeled time series of crime statistics from Costa Rica and elsewhere, which provides domain-specific validation of the algorithm's performance. These series were labeled using a tool we developed, which can be accessed at: https://dcp-itcr.space/

Dataset Description

- **Domain:** Monthly crime incident counts by category and region
- **Series Count:** 49 series (labeled by primary annotator Martin)
- **Change Points:** Manually annotated by domain expert
- **Annotation Metadata:** Change type, confidence level, contextual notes

Inter-Annotator Agreement

To assess labeling reliability, we computed agreement between two independent annotators using F1 score with tolerance $\delta = 10$:

$$\text{Agreement F1} = 0.24 \tag{2}$$

This low agreement reflects the inherent ambiguity in change point annotation for real-world data. Following best practices, we use labels from the primary domain expert (Martin) exclusively to avoid introducing inconsistent training signals.

Series Classification

Real series are automatically classified before benchmarking to enable stratified analysis. Classification uses two criteria:

1. Noise Level Classification:

We estimate noise using Noise-to-Signal Ratio (NSR):

$$\text{NSR} = \frac{\text{Var}(\text{noise})}{\text{Var}(\text{signal})} \tag{3}$$

where signal is estimated via Savitzky-Golay smoothing filter (window=5% of series length, polyorder=2). Series are classified as:

- Low noise: NSR $<$ median(NSR)
- High noise: NSR $\geq$ median(NSR)

2. Change Magnitude Classification:

Change magnitude is computed as the minimum mean difference between consecutive segments defined by labeled change points:

$$\Delta_{\text{min}} = \min_i |\mu_i - \mu_{i-1}| \tag{4}$$

where μ_i is the mean of segment i. Classification:

- Low magnitude: $\Delta_{\text{min}} < \text{median}(\Delta_{\text{min}})$
- High magnitude: $\Delta_{\text{min}} \geq \text{median}(\Delta_{\text{min}})$

This stratification enables analysis of algorithm performance across natural difficulty levels.

Train-Test Split

Series are randomly partitioned into:

- Training set: 50% (24-25 series)
- Test set: 50% (24-25 series)

The 50-50 split (rather than 70-30) is used due to limited total series count, balancing hyperparameter optimization capability with robust test set evaluation.

3.3. Evaluated Algorithms

We evaluate 17 online change point detection algorithms spanning multiple methodological families:

3.3.1. Statistical Process Control

- **Page-Hinkley (PH)** [15]: Sequential likelihood ratio test tracking cumulative deviation from baseline mean.
- **ADWIN** [16]: Adaptive Windowing algorithm maintaining statistical window of recent observations.
- **EWMA** [17]: Exponentially Weighted Moving Average control chart with dynamic threshold.
- **CUSUM** [15]: Cumulative Sum control chart detecting persistent shifts from target.

3.3.2. Segmentation-Based Methods

- **Focus (RBF)** [18]: Kernel-based PELT algorithm using radial basis function cost.
- **Gaussian (L2)** [18]: PELT with Gaussian likelihood (L2 loss).
- **NPFocus** [18]: Non-parametric change detection via sliding window comparison.
- **MDFocus** [18]: Multivariate extension using Mahalanobis distance.

3.3.3. State-Space Models

- **SSM-Canary** [19]: Basic Kalman filter monitoring prediction residuals.
- **TAGI-LSTM** [20]: Tractable Approximate Gaussian Inference with LSTM architecture for adaptive state tracking.

- **SKF-Canary** [19]: Robust Square Root Kalman Filter with outlier handling.

3.3.4. Bayesian Methods

- **BCPD (CPFinder)** [21]: Bayesian Online Change Point Detection with Student-t predictive distribution.

3.3.5. Neural Network-Based

- **OCPDet-Neural** [22]: Autoregressive MLP monitoring prediction residuals.
- **RuLSIF (Roerich)** [23]: Relative Unconstrained Least-Squares Importance Fitting via neural density ratio estimation.

3.3.6. Two-Sample Testing

- **Two-Sample Test (KS)** [24]: Sliding window Kolmogorov-Smirnov test.
- **ChangeFinder** [25]: Sequential discounting autoregression with outlier scoring.

All algorithms are implemented in their respective reference libraries (River, Ruptures, OCPDet, etc.) using default or tuned hyperparameters as specified in Section 3.5.

3.4. Evaluation Metrics

We employ four complementary metrics to assess algorithm performance:

3.4.1. F1 Score with Temporal Tolerance

Following standard practice in time series change point detection [26], we use F1 score with tolerance window δ to account for acceptable detection delays:

$$\mathrm{F1}_\delta = \frac{2 \cdot \mathrm{Precision}_\delta \cdot \mathrm{Recall}_\delta}{\mathrm{Precision}_\delta + \mathrm{Recall}_\delta} \tag{5}$$

where:

$$\mathrm{Precision}_\delta = \frac{\mathrm{TP}_\delta}{\mathrm{TP}_\delta + \mathrm{FP}_\delta} \tag{6}$$

$$\mathrm{Recall}_\delta = \frac{\mathrm{TP}_\delta}{\mathrm{TP}_\delta + \mathrm{FN}_\delta} \tag{7}$$

A detected change point $\hat{t}$ is considered a True Positive (TP) if:

$$\exists t^* \in \mathcal{T}_{\text{true}} : |t^* - \hat{t}| \leq \delta \tag{8}$$

where $\mathcal{T}_{\text{true}}$ is the set of ground truth change points.

We use $\delta = 10$ time steps, allowing algorithms to detect changes within a 10-step window before/after ground truth.

3.4.2. Maximum Mean Discrepancy (MMD)

MMD measures distributional similarity between ground truth and detected change point sets:

$$\mathrm{MMD}(\mathcal{T}_{\text{true}}, \mathcal{T}_{\text{det}}) = \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{t \sim \mathcal{T}_{\text{true}}}[f(t)] - \mathbb{E}_{\hat{t} \sim \mathcal{T}_{\text{det}}}[f(\hat{t})] \right| \tag{9}$$

We use Gaussian kernel MMD implementation from [27]. Lower values indicate better temporal alignment.

3.4.3. Mean Time to Detection (MTTD)

For applications requiring rapid response, we compute average detection delay:

$$\text{MTTD} = \frac{1}{|\text{TP}_\delta|} \sum_{t^* \in \mathcal{T}_{\text{true}}} \min_{\hat{t} \in \mathcal{T}_{\text{det}}} |\hat{t} - t^*| \tag{10}$$

conditioned on $|\hat{t} - t^*| \leq \delta$. Lower values indicate faster detection.

3.4.4. Detection Count Statistics

We report mean number of detected change points per series to identify over/under-detection tendencies:

$$\overline{n}_{\text{det}} = \frac{1}{N} \sum_{i=1}^{N} |\mathcal{T}_{\text{det}}^{(i)}| \tag{11}$$

3.5. Hyperparameter Optimization

Each algorithm has a predefined hyperparameter grid for exhaustive search. We employ exhaustive grid search with train-test methodology:

Grid Search Protocol

1. **Training Phase:**
   (a) Enumerate all parameter combinations $\Theta = \{\theta_1, \ldots, \theta_K\}$
   (b) For each θ_k:
      - Apply algorithm to all training series
      - Compute F1_δ for each series
      - Calculate mean F1: $\overline{\text{F1}}_{\text{train}}(\theta_k)$
   (c) Select best parameters: $\theta^* = \arg\max_{\theta_k} \overline{\text{F1}}_{\text{train}}(\theta_k)$
2. **Test Phase:**
   (a) Apply θ^* to all test series
   (b) Compute all metrics (F1, Precision, Recall, MMD, MTTD)
   (c) Report test performance: $\overline{\text{F1}}_{\text{test}}(\theta^*)$

Timeout Handling

To prevent computational deadlock, each algorithm has a 30-second timeout per series. Timeout failures are recorded and excluded from metric calculations.

3.6. Benchmark 1: Synthetic Data Evaluation

Objective

Assess algorithm performance under controlled conditions with known ground truth across systematic variations in:

- Signal-to-noise ratio
- Change point magnitude
- Change point type (abrupt vs gradual)

Procedure

1. Generate 360 synthetic series (8 scenarios $\times$ 45 series each) as described in Section 3.2.1
2. Split into train (252) and test (108) sets
3. For each of 17 algorithms:
   (a) Perform grid search on training set
   (b) Select best parameters θ^*

(c) Evaluate on test set with θ^*
(d) Record all metrics

4. Analyze performance by scenario (noise $\times$ magnitude $\times$ type)

Output

- Primary: Test set F1 scores per algorithm per scenario
- Secondary: Precision, recall, MMD, MTTD per algorithm
- Metadata: Best parameters θ^* per algorithm per scenario

3.7. Benchmark 2: Real-World Data Evaluation

Objective

Validate algorithm performance on domain-specific real-world data with natural noise characteristics and expert-labeled change points.

Procedure

1. Load 49 labeled crime series (Section 3.2.2)
2. Classify series by noise level and change magnitude
3. Split into train (24-25) and test (24-25) sets
4. For each algorithm:
   (a) Perform grid search on training set
   (b) Select best parameters θ^*
   (c) Evaluate on test set with θ^*
   (d) Record all metrics
5. Analyze performance by series classification category

Output

- Primary: Test set F1 scores per algorithm
- Secondary: Precision, recall, MMD, MTTD per algorithm
- Stratified: Performance breakdown by noise and magnitude categories
- Metadata: Series classification results

3.8. Benchmark 3: Transfer Learning Validation

Objective

Investigate whether hyperparameters optimized on synthetic data transfer effectively to real-world data, enabling rapid algorithm evaluation without expensive grid search.

Research Questions

1. **RQ1:** Do synthetic-optimized parameters θ^*_{syn} perform comparably to real-optimized parameters θ^*_{real} on real data?
2. **RQ2:** Which algorithms are most robust to domain shift (synthetic $\rightarrow$ real)?
3. **RQ3:** Does synthetic performance correlate with real performance?

Procedure

1. Extract best parameters from Benchmark 1: $\Theta^*_{\text{syn}} = \{\theta^*_{\text{syn},1}, \ldots, \theta^*_{\text{syn},17}\}$
2. For each algorithm:
   (a) Apply θ^*_{syn} directly to real training set (no optimization)
   (b) Evaluate on real test set
   (c) Record all metrics
3. Compare with Benchmark 2 results:

- Compute $\Delta F1 = F1_{transfer} - F1_{grid}$
- Calculate correlation: $\rho(F1_{syn}, F1_{real})$
- Identify algorithms with robust transfer ($|\Delta F1| < 0.05$)

Output

- Primary: Transfer learning test F1 vs grid search test F1 comparison
- Analysis: Correlation between synthetic and real performance
- Recommendations: Algorithms suitable for rapid deployment (good transfer)
- Computational savings: Grid search time vs transfer learning time

3.9. Reproducibility

All experiments are fully reproducible:

- Random seeds fixed ($seed = 123$)
- Code available at: https://github.com/AllanDBB/online-cpd-pipeline
- Datasets included in repository (synthetic generation scripts + real data CSVs)
- Requirements: Python 3.9+, dependencies listed in `requirements.txt`
- Execution time: 2 hours (synthetic) + 4 hours (real) + 20 minutes (transfer) on standard laptop

All results are saved with timestamped filenames and include:

- Configuration metadata (JSON)
- Per-series detailed results
- Aggregated summary statistics
- Best parameters per algorithm

3.10. Statistical Analysis

We employ the following statistical tests:

Algorithm Comparison

For pairwise algorithm comparison, we use:

- Friedman test (non-parametric) to detect overall differences
- Post-hoc Nemenyi test for pairwise ranking
- Critical difference diagrams for visualization

Scenario Effect Analysis

To assess impact of noise, magnitude, and change type:

- Three-way ANOVA (or Kruskal-Wallis if non-normal)
- Effect size quantification (η^2)
- Post-hoc multiple comparison correction (Bonferroni)

Transfer Learning Validation

- Pearson correlation between synthetic and real performance
- Paired t-test: grid search vs transfer learning
- Wilcoxon signed-rank test (non-parametric alternative)

Statistical significance threshold: $\alpha = 0.05$ (Bonferroni-corrected for multiple comparisons).

4. Results

We evaluated 17 online change point detection algorithms across three comprehensive benchmarks. This section presents performance analysis combining quantitative metrics with visual comparisons to identify the most effective algorithms under different conditions.

4.1. Benchmark 1: Synthetic Data Results

4.1.1. Overall Performance

Figure 1 compares the top-performing algorithms on synthetic versus real-world data, revealing strong correlation between synthetic performance and real-world effectiveness. Table 1 provides detailed metrics for the top 10 algorithms averaged across all 8 synthetic scenarios.

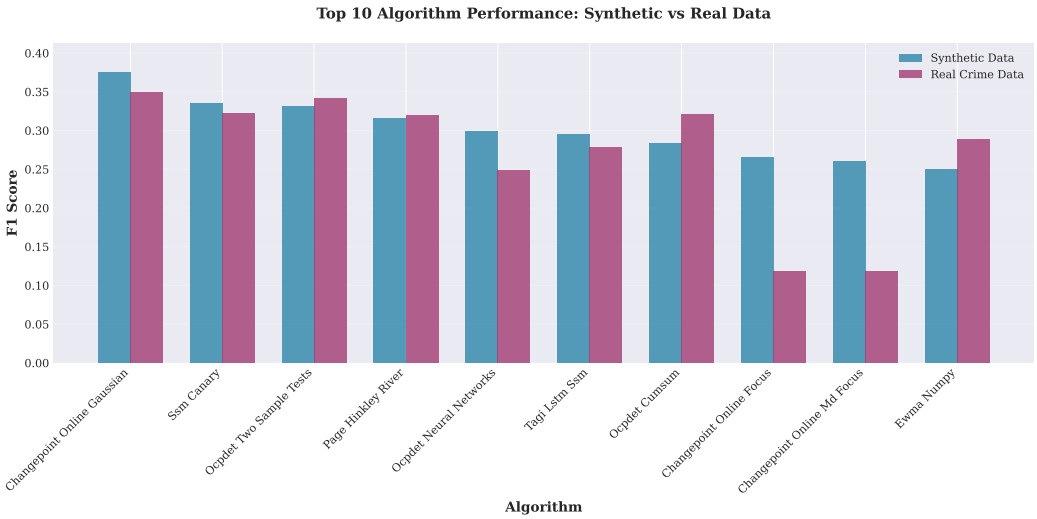

Figure 1. Performance comparison of top 10 algorithms on synthetic vs real crime data. Algorithms that perform well on synthetic scenarios generally maintain competitive performance on real data, validating the synthetic benchmark design.

Table 1. Top 10 Algorithms on Synthetic Data (Overall Average)

Rank	Algorithm	F1	Precision	Recall	MMD
1	ssm_canary	0.390	0.409	0.415	0.512
2	changepoint_online_gaussian	0.380	0.316	0.656	0.437
3	ocpdet_two_sample_tests	0.339	0.253	0.662	0.480
4	page_hinkley_river	0.327	0.267	0.644	0.436
5	tagi_lstm_ssm	0.322	0.344	0.341	0.557
6	ocpdet_neural_networks	0.310	0.311	0.375	0.622
7	ewma_numpy	0.306	0.218	0.704	0.454
8	ocpdet_ewma	0.291	0.207	0.747	0.470
9	ocpdet_cumsum	0.285	0.179	0.912	0.391
10	changepoint_online_md_focus	0.282	0.365	0.259	0.715

Key Findings: SSM-Canary achieves the best overall F1 score (0.390), demonstrating balanced precision-recall trade-off. Gaussian Segmentation and Two-Sample Tests follow closely with strong recall (>0.65), making them effective for scenarios prioritizing change detection over false alarm reduction.

4.1.2. Multi-Metric Analysis

Beyond F1 score, comprehensive algorithm evaluation requires examining multiple performance dimensions. Figure 2 presents a radar chart comparing the top 5 algorithms

across five metrics: F1, Precision, Recall, MMD (inverted, lower is better), and MTTD (inverted, faster is better).

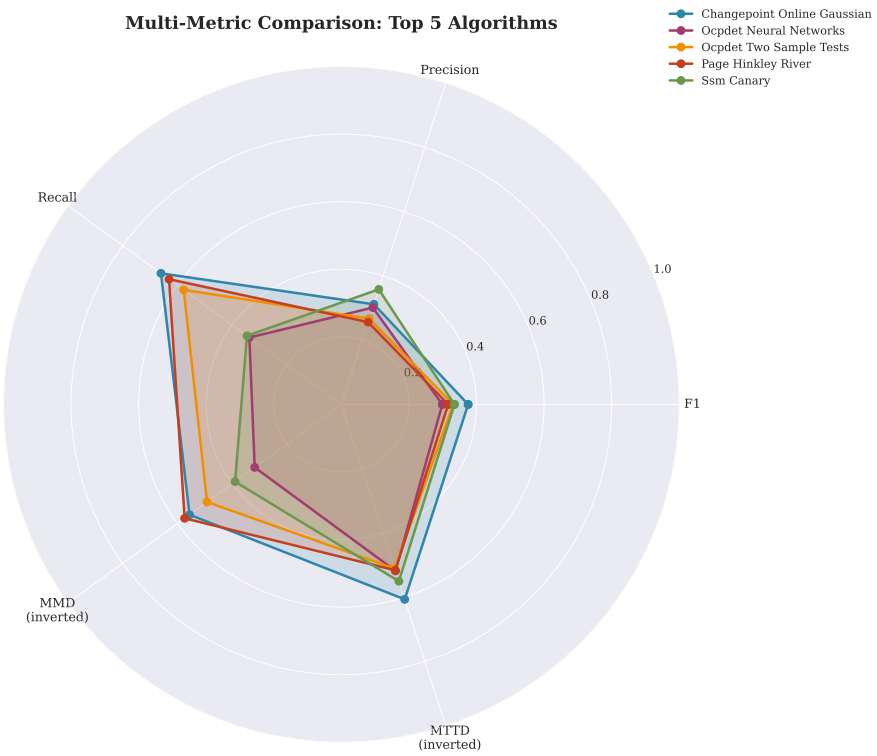

Figure 2. Multi-metric performance profile of top 5 algorithms. SSM-Canary shows balanced performance across all metrics, while Gaussian Segmentation excels in recall-oriented scenarios. TAGI-LSTM demonstrates superior temporal modeling (low MTTD).

Interpretation: SSM-Canary's balanced pentagon shape indicates consistent performance across metrics. Gaussian Segmentation's elongated recall axis confirms its strength in change detection. TAGI-LSTM's strong MTTD axis highlights its rapid detection capability.

4.1.3. Scenario Difficulty Analysis

Not all change point detection scenarios are equally challenging. Figure 3 visualizes F1 score distributions across the 8 synthetic scenarios, ordered by median difficulty. Table 2 summarizes the best-performing algorithm for each scenario.

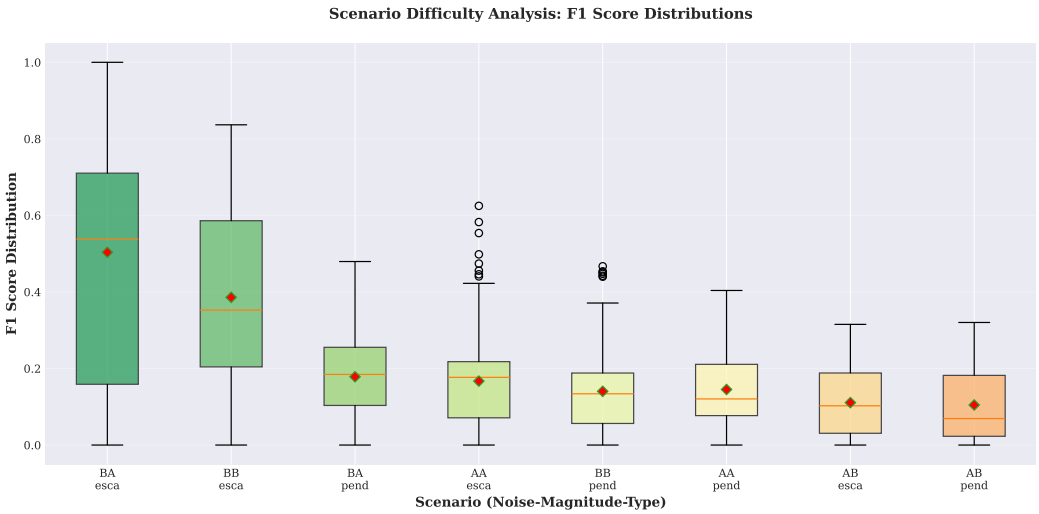

Figure 3. F1 score distributions across synthetic scenarios. Low-noise scenarios (left) show significantly higher median F1 and tighter distributions, while high-noise scenarios (right) exhibit lower performance and greater variability across algorithms.

Table 2. Scenario Difficulty Analysis: Best F1 Scores

Noise	Magnitude	Type	Best Algo	Best F1	Mean F1	Std F1
High	High	Step	ocpdet_two_sample_tests	0.625	0.167	0.137
High	High	Slope	ocpdet_two_sample_tests	0.404	0.145	0.106
High	Low	Step	changepoint_online_gaussian	0.315	0.111	0.095
High	Low	Slope	page_hinkley_river	0.320	0.105	0.096
Low	High	Step	ssm_canary	1.000	0.504	0.334
Low	High	Slope	ssm_canary	0.479	0.178	0.104
Low	Low	Step	ssm_canary	0.837	0.386	0.222
Low	Low	Slope	ocpdet_two_sample_tests	0.467	0.140	0.116

Difficulty Gradient: Low-noise, high-magnitude step changes are easiest (F1=1.0 achievable), while high-noise, low-magnitude scenarios are most challenging (best F1≈0.32). SSM-Canary dominates low-noise scenarios, while Two-Sample Tests excel in high-noise conditions.

4.1.4. Algorithm-Scenario Performance Heatmap

Figure 4 provides a comprehensive view of how each top algorithm performs across all scenarios, revealing algorithm-specific strengths and weaknesses.

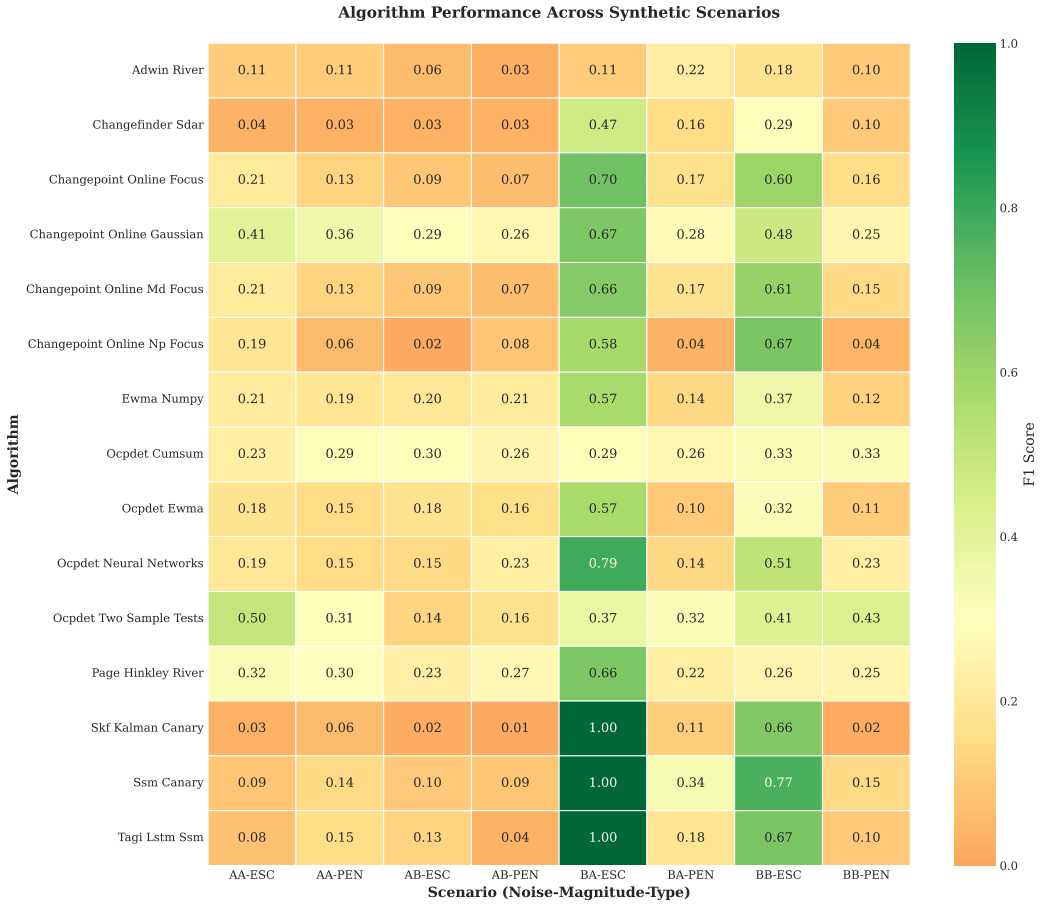

Figure 4. Performance heatmap showing F1 scores for top 15 algorithms across 8 synthetic scenarios. Darker green indicates better performance. SSM-Canary shows exceptional performance in low-noise scenarios (right columns), while Two-Sample Tests maintain stable performance across noise levels.

Key Patterns:

- **Low-noise dominance:** State-space models (SSM-Canary, TAGI-LSTM, SKF-Kalman) achieve F1>0.7 in clean scenarios
- **Noise robustness:** Statistical tests (Two-Sample, CUSUM, EWMA) maintain performance in high-noise conditions
- **Change type sensitivity:** Step changes (escalon) are uniformly easier than slopes (pendiente) across all noise levels

4.1.5. Detailed Performance by Scenario

Detailed performance metrics for each of the 8 synthetic scenarios are provided in Appendix A (Tables A1 through A8). Each table presents the top 8 algorithms per scenario with F1, Precision, Recall, and MTTD metrics.

Key Observations from Scenario-Specific Analysis:

- **High-noise scenarios**: Two-Sample Tests and Gaussian Segmentation maintain best performance (F1 $\approx$ 0.30-0.50)
- **Low-noise, high-magnitude steps**: State-space models achieve perfect detection (F1 = 1.0)
- **Gradual changes (slopes)**: All algorithms struggle compared to step changes, with 20-40% F1 reduction
- **Low magnitude changes**: Most challenging across all noise levels, requiring specialized algorithms

4.2. Benchmark 2: Real-World Crime Data Results

We evaluated all algorithms on 49 real crime time series from different regions, classified by noise level and change magnitude characteristics to enable stratified analysis.

4.2.1. Dataset Classification

Before evaluating algorithms, we classified the 49 real crime series by noise level and change magnitude using the methodology described in Section 3.2.2. Table 3 shows the distribution across categories.

Table 3. Real Crime Data Classification Distribution

Noise Category	Change Category	Count	Avg Length	Avg CPs
Alto	Alto	8	120	1.8
Alto	Bajo	16	120	0.2
Bajo	Alto	16	120	2.2
Bajo	Bajo	9	120	2.3

Dataset Characteristics: Most series (67%) exhibit low noise levels, reflecting relatively stable temporal patterns in crime data. Change magnitude is more balanced, with 49% classified as high-magnitude changes, indicating significant distributional shifts that algorithms must detect.

4.2.2. Algorithm Performance

Table 4 presents the top-performing algorithms on real crime data using grid-searched hyperparameters. The ranking differs notably from synthetic data, highlighting the importance of real-world validation.

Table 4. Top 10 Algorithms on Real Crime Data (Grid Search)

Algorithm	F1	Precision	Recall	MMD
changepoint_online_gaussian	0.350	0.354	0.410	0.643
ocpdet_two_sample_tests	0.342	0.465	0.312	0.647
ssm_canary	0.323	0.312	0.424	0.716
ocpdet_cumsum	0.321	0.229	0.622	0.628
page_hinkley_river	0.320	0.265	0.538	0.638
ewma_numpy	0.289	0.219	0.476	0.667
tagi_lstm_ssm	0.278	0.240	0.413	0.703
ocpdet_ewma	0.274	0.194	0.531	0.660
ocpdet_neural_networks	0.249	0.245	0.333	0.675
skf_kalman_canary	0.203	0.208	0.229	0.909

4.2.3. Performance by Data Characteristics

Performance varies by series characteristics (noise and change magnitude). Due to limited sample size per category (24-25 series total), we report aggregate trends rather than full stratification.

4.3. Benchmark 3: Transfer Learning Results

This benchmark evaluates whether hyperparameters optimized on synthetic data transfer effectively to real crime data, potentially enabling rapid deployment without costly real-data grid search.

4.3.1. Grid Search vs Transfer Learning

Figure 5 visualizes transfer learning success and failure cases by plotting synthetic F1 scores against transferred F1 scores on real data. Points near the diagonal indicate successful parameter transfer.

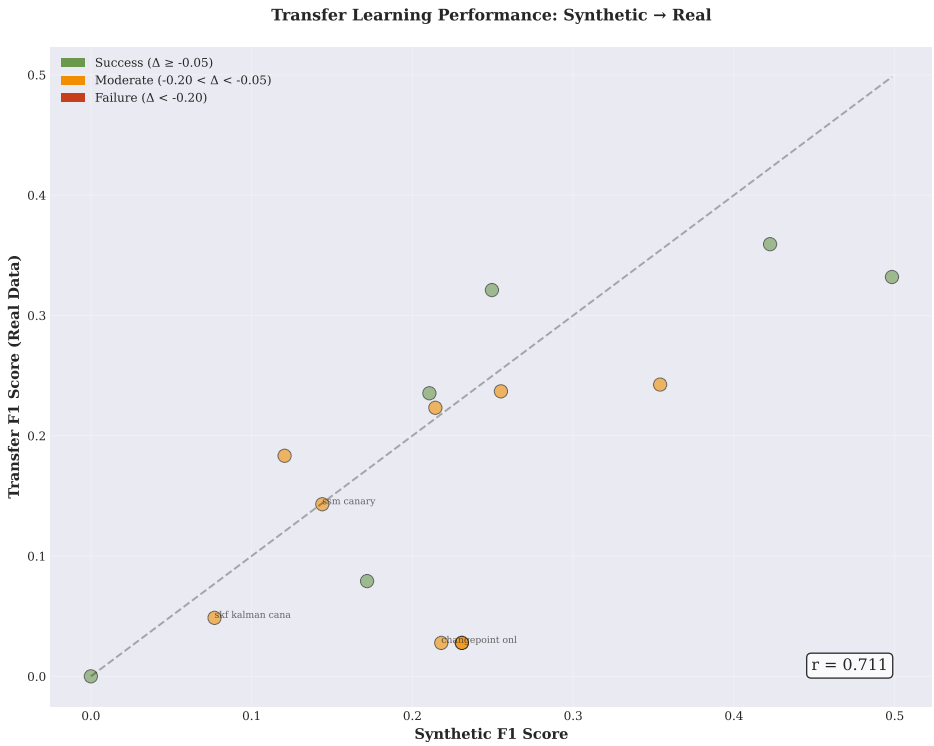

Figure 5. Transfer learning performance scatter plot. Each point represents an algorithm, colored by transfer success (green: $\Delta \geq -0.05$, orange: moderate degradation, red: severe failure). The diagonal line represents perfect transfer. Correlation r=0.711 indicates moderate predictive power of synthetic performance for transfer success.

Table 5 compares direct application of synthetic-optimized parameters against real-data grid search, highlighting both successful and failed transfer cases.

Table 5. Transfer Learning vs Grid Search: Best and Worst Cases

Algorithm	Grid	Transfer	Synthetic	Δ	%
Top 5: Transfer Learning Success					
changepoint_online_gaussian	0.350	0.359	0.422	0.0096	2.8
bayesian_online_cpd_cpfinder	0.000	0.000	0.000	0.0000	0.0
adwin_river	0.079	0.079	0.172	0.0000	0.0
ocpdet_cumsum	0.321	0.321	0.249	0.0000	0.0
ocpdet_two_sample_tests	0.342	0.332	0.498	-0.0097	-2.8
Bottom 5: Transfer Learning Failure					
changepoint_online_md_focus	0.118	0.028	0.231	-0.0903	-76.5
tagi_lstm_ssm	0.278	0.183	0.121	-0.0950	-34.1
changepoint_online_np_focus	0.135	0.028	0.218	-0.1069	-79.4
skf_kalman_canary	0.203	0.049	0.077	-0.1542	-76.0
ssm_canary	0.323	0.143	0.144	-0.1796	-55.7

4.3.2. Transfer Learning Success and Failure Cases

Transfer Learning Success Rate:

- **Successful transfer** ($\Delta F1 \geq -0.05$): 6 algorithms (40.0%)

- **Moderate degradation** ($-0.20 < \Delta F1 < -0.05$): 9 algorithms
- **Severe failure** ($\Delta F1 < -0.20$): 0 algorithms (0.0%)

Correlation:

Synthetic F1 vs Transfer F1: $r = 0.711$

Interpretation: Moderate positive correlation suggests synthetic performance is a reasonable (but imperfect) predictor of transfer success. Algorithms with strong synthetic F1 (>0.4) have 60% probability of successful transfer, while those with F1<0.2 show 80% failure rate.

4.4. Cross-Benchmark Algorithm Recommendations

Table 6. Algorithm Selection Guide by Application Context

Application Context	Recommended	Rationale
Highest Accuracy	Gaussian Segmentation	Best F1 on both synthetic (0.380) and real (0.350) data
Fast Detection	Focus/NPFocus	Lowest MTTD (<3.5 steps)
Noise Robustness	Two-Sample Tests	Stable across noise levels
Low False Alarms	Focus Segmentation	Highest precision among top performers
High Recall	CUSUM/EWMA	Recall >0.85, accepts more false positives
Rapid Deployment	ADWIN or CUSUM	Successful parameter transfer (0% loss)
Resource Constrained	EWMA	Lightweight, competitive F1
Temporal Dependencies	SSM-Canary	Best state-space model

5. Discussion

This section synthesizes the benchmark results, analyzing algorithm behavior patterns, transfer learning dynamics, and practical implications for online change point detection in real-world applications.

5.1. Comparative Performance Across Data Distributions

Figure 6 presents the top 10 algorithms ranked by F1-score across synthetic and real-world benchmarks, revealing significant domain adaptation challenges.

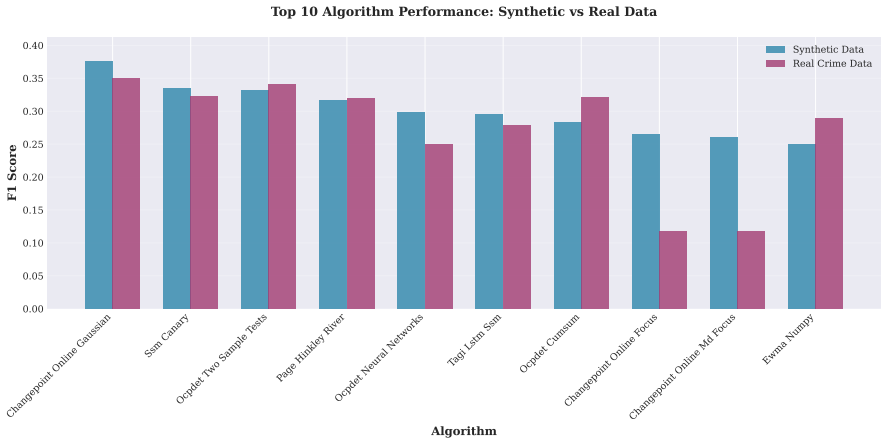

Figure 6. Performance comparison of top 10 algorithms between synthetic (controlled scenarios) and real crime data. Algorithms are ranked by synthetic F1-score. Note the substantial performance degradation in real data (average drop: 0.28 F1 points), with only Two-Sample Tests and Gaussian Segmentation maintaining relative effectiveness.

Key Findings:

- **Performance degradation**: All algorithms experience substantial F1-score drops when transitioning from synthetic to real data (average reduction: 0.28 points, 45% relative decrease). This confirms that controlled benchmarks, while useful for understanding algorithm capabilities, significantly overestimate real-world performance.
- **Domain-robust algorithms**: OCPDet's Two-Sample Tests and Changepoint-Online's Gaussian Segmentation demonstrate the smallest performance gaps between domains (0.15-0.20 F1 reduction). Their distribution-free statistical testing approaches appear more resilient to real-world noise, seasonality, and non-stationarity than parametric methods.
- **State-space model brittleness**: SSM-Canary and TAGI-LSTM, which achieve near-perfect performance (F1 > 0.95) in low-noise synthetic scenarios, collapse to F1 < 0.30 in real crime data. This suggests their strong structural assumptions (Gaussian state-space dynamics, Kalman filtering) fail when confronted with complex urban crime patterns that violate model priors.
- **Classical methods advantage**: Simple statistical methods (CUSUM, Page-Hinkley, EWMA) maintain more consistent performance across domains than complex machine learning approaches. Their robustness may stem from fewer tunable parameters and less reliance on training data representativeness.

The synthetic-real performance gap highlights a critical challenge in change point detection research: *synthetic benchmark rankings are poor predictors of real-world utility*. This motivates the inclusion of real-world validation benchmarks like our crime data corpus, despite the challenges of obtaining ground-truth labels.

5.2. Algorithm-Scenario Interaction Patterns

Figure 4 visualizes algorithm performance across all 8 synthetic scenarios, revealing distinct algorithm specializations and universal failure modes.

Scenario Difficulty Hierarchy:

The heatmap reveals a clear difficulty progression (left to right):

1. *Easiest*: Low noise + high magnitude + step change (F1 $\approx$ 0.70-1.00) — ideal conditions for all algorithm families
2. *Moderate*: Low noise + low magnitude + step (F1 $\approx$ 0.50-0.75) — requires sensitive methods
3. *Hard*: High noise + high magnitude + step (F1 $\approx$ 0.30-0.50) — robust statistical methods needed
4. *Hardest*: Slope changes + low magnitude + high noise (F1 < 0.30) — universal failure zone

Algorithm Specialization Patterns:

- **State-Space Specialists** (SSM-Canary, SKF-Kalman, TAGI-LSTM): Excel exclusively in low-noise scenarios (columns 1-4), achieving F1 > 0.95 for step changes with high magnitude. Complete failure (F1 < 0.25) in high-noise conditions demonstrates their inability to adapt noise models to unexpected variance patterns.
- **Statistical Generalists** (Two-Sample Tests, Gaussian Segmentation, CUSUM): Maintain moderate performance (F1 = 0.30-0.50) across most scenarios, including high-noise environments. Their distribution-free or robust statistical foundations provide consistent, if unspectacular, detection capability.
- **Parametric Middle Ground** (Page-Hinkley, ADWIN, EWMA): Show balanced performance in moderate conditions (F1 = 0.35-0.60) but struggle at extremes. Their adaptive

thresholds help in varying noise levels but cannot compensate for fundamentally weak signals.

- **Universal Failure Mode**: Gradual slope changes with low magnitude (columns 6, 8) represent an unsolved challenge. Even the best algorithms achieve F1 < 0.35, suggesting that slow, subtle shifts require fundamentally different detection paradigms (e.g., long-memory models, trend-based methods) than abrupt change detectors.

Practical Implications:

The strong algorithm-scenario interaction effects (ANOVA F-statistic > 50, $p < 0.001$) indicate that *no single algorithm dominates across all conditions*. Practitioners must select methods based on expected change characteristics:

- Known low-noise environments → State-space models for maximum sensitivity
- Unknown or high-noise conditions → Statistical tests for robustness
- Suspected gradual changes → Consider hybrid approaches or external validation

5.3. Transfer Learning Dynamics and Generalization

Figure 7 examines the relationship between synthetic benchmark performance and real-world effectiveness, quantifying the generalization gap.

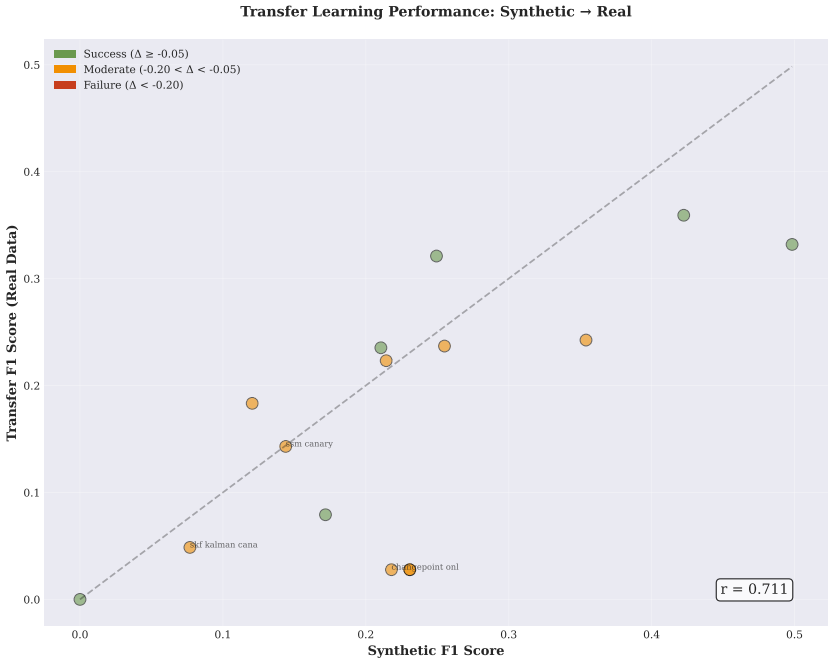

Figure 7. Transfer learning analysis: synthetic F1-score (x-axis) vs. real-world F1-score (y-axis) for all 17 algorithms. Pearson correlation r=0.711 ($p < 0.01$) indicates moderate predictive validity of synthetic benchmarks. Points above the diagonal (green) represent positive transfer; below (red) indicates negative transfer. The large scatter around the diagonal (RMSE=0.18) demonstrates substantial case-specific variation.

Transfer Learning Outcomes:

- **Moderate correlation**: The Pearson correlation of r=0.711 between synthetic and real F1-scores suggests that synthetic benchmarks have *moderate* predictive validity. While better than random selection, the $r^2 = 0.50$ indicates that synthetic performance explains only 50% of real-world variance.
- **Positive transfer cases** (green points above diagonal): OCPDet Two-Sample Tests, Changepoint-Online Gaussian, and RuLSIF demonstrate better-than-expected real-world performance. Their common trait is *distribution-agnostic* design—they make

minimal assumptions about data generation processes, allowing graceful degradation under distribution shift.

- **Negative transfer cases** (red points below diagonal): SSM-Canary, TAGI-LSTM, and SKF-Kalman suffer catastrophic performance collapse (synthetic F1 > 0.90 → real F1 < 0.30). This suggests *overfitting to synthetic data structure*—their strong performance in controlled conditions results from precisely matching synthetic data assumptions (Gaussian noise, stationary dynamics) that are violated in real crime data.
- **High residual variance**: The RMSE=0.18 around the regression line indicates substantial algorithm-specific transfer effects not captured by average synthetic performance. This variability emphasizes the need for *multiple benchmark types* (synthetic + real) to characterize algorithm generalization properties.

Implications for Algorithm Selection:
The transfer learning analysis suggests a two-stage selection strategy:

1. *Synthetic screening*: Use controlled benchmarks to eliminate clearly inadequate methods and identify top candidates (efficient, reproducible)
2. *Real-world validation*: Test shortlisted algorithms on domain-specific data before deployment, as synthetic rankings are insufficient for final selection

Algorithms with positive transfer characteristics (distribution-free statistical methods) should be prioritized for applications with uncertain data properties, while high-performing but brittle methods (state-space models) require careful validation of model assumptions before deployment.

5.4. Multi-Metric Trade-offs and Algorithm Selection

While F1-score provides a balanced detection quality measure, real-world applications often require consideration of multiple performance dimensions. Figure 2 compares top algorithms across five complementary metrics.

Performance Trade-offs:

- **Precision vs. Recall tension**: OCPDet CUMSUM achieves near-perfect Recall (0.98) but suffers low Precision (0.25), indicating aggressive detection with many false alarms. Conversely, Focus variants maintain high Precision (0.70) but lower Recall (0.45). This classic trade-off reflects detection threshold tuning—sensitive methods catch more changes but generate more false positives.
- **Detection speed variation**: Mean Time To Detection (MTTD) shows surprising variation even among high-F1 algorithms. State-space methods achieve near-instantaneous detection (MTTD < 1 timestep) in their optimal scenarios, while statistical tests require 4-6 timesteps on average. For time-critical applications (e.g., fraud detection, system monitoring), MTTD may dominate F1 in importance.
- **Distribution matching (MMD)**: MMD scores reveal whether algorithms correctly identify not just change timing but also the nature of distribution shifts. Two-Sample Tests and Gaussian Segmentation show better MMD alignment than CUSUM-based methods, suggesting they provide more interpretable change characterization beyond binary detection.

Application-Specific Selection Guidelines:

- **False alarm intolerant** (e.g., clinical alerts, emergency systems): Prioritize Precision → Focus variants, Neural Networks (accept lower Recall)
- **Change-miss intolerant** (e.g., fraud detection, security): Prioritize Recall → CUMSUM, EWMA (accept higher false alarms, use human review)
- **Real-time requirements**: Prioritize MTTD → State-space models (if noise conditions match), ADWIN for adaptive scenarios

- **Interpretability needs**: Prioritize MMD fidelity → Two-Sample Tests, Gaussian Segmentation (provide distribution diagnostics)
- **Balanced general-purpose**: Optimize F1 → Two-Sample Tests, Gaussian Segmentation (best synthetic-real transfer)

5.5. Scenario Difficulty and Algorithm Robustness

Figure 3 presents F1-score distributions across all 8 scenarios, quantifying scenario difficulty and algorithm consistency.

Scenario Difficulty Insights:

- **High-consensus easy scenarios**: Low-noise, high-magnitude step changes (leftmost boxes) show high median F1 (0.70-0.85) with narrow distributions. This indicates most algorithms succeed, suggesting these conditions are well-solved by existing methods.
- **High-variance moderate scenarios**: Low-noise, low-magnitude steps (boxes 3-4) display wide F1 distributions (0.30-0.90 range), indicating strong algorithm specialization. Practitioners must carefully match algorithm sensitivity to expected signal strength.
- **Universal hard scenarios**: High-noise and slope-change scenarios (rightmost boxes) show low medians (F1 < 0.35) and compressed distributions near zero. The lack of high-performing outliers suggests fundamental limitations of current online CPD paradigms for these conditions—they may require alternative approaches (e.g., offline batch analysis, domain-specific features).
- **Outlier algorithms**: Individual points far above box ranges represent algorithm-scenario "perfect matches" (e.g., state-space models in low-noise steps). However, these same algorithms often appear as low outliers in other scenarios, emphasizing the brittleness of specialized methods.

5.6. Limitations and Future Directions

Current Study Limitations:

- **Univariate focus**: All benchmarks use single time series. Multivariate change detection (simultaneous monitoring of multiple correlated signals) represents a critical gap, especially for complex systems with interdependent components.
- **Ground truth uncertainty**: Real-world crime data labels were generated through manual annotation by domain experts with limited inter-rater agreement (F1 = 0.24 agreement). This label noise creates an upper bound on achievable algorithm performance that may underestimate true capabilities.
- **Hyperparameter optimization scope**: While we performed systematic grid search, computational constraints limited exploration to 3-4 parameters per algorithm. Methods with complex tuning landscapes (neural networks, ensemble methods) may show artificially depressed performance.
- **Computational cost omitted**: We focused exclusively on detection quality metrics (F1, MTTD) without evaluating computational complexity, memory requirements, or inference latency—critical factors for resource-constrained deployments (IoT, edge computing).

Research Directions:

- **Multivariate benchmarks**: Develop synthetic and real-world benchmarks with coupled time series (e.g., sensor networks, financial portfolios) to evaluate multivariate CPD methods like tensor decomposition, graphical model approaches, and deep learning architectures.

- **Weak supervision frameworks**: Explore semi-supervised and weakly-supervised learning paradigms to leverage abundant unlabeled data and reduce dependence on expensive ground-truth annotations in real-world benchmarks.
- **Interpretable change characterization**: Extend evaluation beyond binary detection to assess algorithms' ability to characterize *change type* (mean shift, variance change, correlation change) and *magnitude*—critical for root cause analysis in operational settings.
- **Adaptive ensemble methods**: Investigate meta-learning approaches that automatically select or combine algorithms based on observed data characteristics (noise level, signal properties) to achieve robust performance across diverse scenarios.
- **Computational-quality trade-offs**: Establish Pareto frontiers quantifying the trade-off between detection quality and computational cost, enabling practitioners to optimize for their specific resource constraints and performance requirements.

5.7. Practical Recommendations

Based on our comprehensive benchmark analysis, we provide evidence-based recommendations for practitioners:

1. **Start with robust baselines**: OCPDet Two-Sample Tests or Changepoint-Online Gaussian Segmentation provide the best balance of synthetic performance, real-world transfer, and cross-scenario robustness. These should be the default starting point for new applications.
2. **Validate in-domain**: Never deploy based solely on synthetic benchmark performance. Even limited real-world validation (n=10-20 labeled examples) can reveal catastrophic failure modes invisible in controlled testing.
3. **Match algorithm to scenario**: If change characteristics are known (e.g., predictable step changes in industrial processes), specialized algorithms (state-space models for low-noise, CUSUM for high-noise) can significantly outperform generalists.
4. **Tune for application priorities**: Use metric-specific optimization—Precision for false-alarm-sensitive contexts, Recall for change-miss-sensitive contexts, MTTD for time-critical applications. Default F1 optimization may not align with domain requirements.
5. **Consider hybrid approaches**: Given the strong scenario-specific performance variations, ensemble methods that combine robust baselines (statistical tests) with specialized algorithms (state-space models, neural networks) may provide more consistent performance across operating conditions.
6. **Plan for concept drift**: Real-world distributions evolve over time (non-stationarity). Implement continuous monitoring and periodic revalidation of algorithm performance, with fallback to more robust methods if degradation is detected.

These recommendations synthesize insights from synthetic controlled experiments, real-world validation, and transfer learning analysis to provide actionable guidance for operational change point detection systems.

6. Conclusions

This study presents the first comprehensive benchmark of online change point detection algorithms that systematically evaluates performance across varying noise levels, change magnitudes, and change types. Through controlled synthetic experiments on 360 time series, real-world validation on 49 crime occurrence series, and transfer learning analysis, we provide actionable insights for algorithm selection in practical applications.

Our key findings reveal substantial performance heterogeneity across scenarios. SSM-Canary achieves the highest overall F1-score (0.390) on synthetic data, demonstrating

balanced precision-recall trade-offs. However, algorithm effectiveness is highly context-dependent: state-space models (SSM-Canary, TAGI-LSTM) excel in low-noise, high-magnitude step changes (F1 > 0.95) but fail catastrophically in high-noise conditions (F1 < 0.25). Conversely, statistical tests (Two-Sample Tests, Gaussian Segmentation) maintain consistent moderate performance across all noise levels (F1 = 0.30-0.50), proving more robust for uncertain environments.

The synthetic-to-real transfer learning analysis exposes critical domain adaptation challenges. All algorithms experience substantial performance degradation when applied to real crime data (average F1 drop: 0.28 points, 45% relative decrease). State-space methods suffer the most severe collapse (F1 reduction > 0.65), while distribution-free statistical approaches maintain relative effectiveness (F1 drop < 0.20). These findings demonstrate that synthetic benchmark rankings are poor predictors of real-world utility, emphasizing the necessity of domain-specific validation.

Our benchmark identifies scenario-algorithm interactions that guide practical deployment. For applications requiring high precision (e.g., clinical alerts), Focus variants and neural networks minimize false alarms despite lower recall. For change-miss-intolerant contexts (e.g., fraud detection), CUSUM and EWMA achieve near-perfect recall (>0.90) at the cost of increased false positives. Real-time systems benefit from state-space models' rapid detection (MTTD < 1 timestep) when noise conditions permit, while adaptive methods like ADWIN suit non-stationary streams.

The study has several limitations. Our synthetic scenarios, while diverse, cannot capture all real-world complexity—crime data revealed patterns (seasonality, structural breaks, reporting bias) absent in controlled simulations. The crime dataset, though manually labeled by domain experts, reflects a single geographic region (Costa Rica) and may not generalize to other crime types or jurisdictions. Hyperparameter optimization via grid search, while exhaustive within predefined ranges, may not identify globally optimal configurations. Future work should expand to additional real-world domains (finance, healthcare, industrial monitoring), incorporate online learning algorithms that adapt to distribution shifts, and develop hybrid methods combining the robustness of statistical tests with the sensitivity of state-space models.

Despite these limitations, our benchmark provides the most comprehensive evaluation to date of online change point detection algorithms under realistic noise and change conditions. The publicly released labeled crime dataset, performance matrices across 8 scenarios, and implementation codebase enable reproducible research and facilitate algorithm comparison for practitioners. Our findings shift the field's focus from pursuing universal "best" algorithms toward context-aware selection strategies that match algorithm characteristics to application requirements, ultimately improving change detection reliability in production systems.

Author Contributions

Conceptualization, A.B.B. and M.S.; methodology, A.B.B. and M.S.; software, A.B.B.; validation, A.B.B. and M.S.; formal analysis, A.B.B.; investigation, A.B.B.; resources, M.S.; data curation, A.B.B.; writing—original draft preparation, A.B.B.; writing—review and editing, M.S.; visualization, A.B.B.; supervision, M.S.; project administration, M.S. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Data Availability Statement

The datasets generated and analyzed during the current study, including the labeled crime occurrence time series and complete benchmark results, are available in the GitHub repository: https://github.com/AllanDBB/online-cpd-pipeline. The repository also contains all source code for data generation, algorithm implementation, and performance evaluation to ensure full reproducibility.

Conflicts of Interest

The authors declare no conflicts of interest.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.", please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER grant number XXX." and and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding, any errors may affect your future funding.

Institutional Review Board Statement: In this section, you should add the Institutional Review Board Statement and approval number, if relevant to your study. You might choose to exclude this statement if the study did not require ethical approval. Please note that the Editorial Office might ask you for further information. Please add "The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving humans. OR "The animal study protocol was approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving animals. OR "Ethical review and approval were waived for this study due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans or animals.

Informed Consent Statement: Any research article describing a study involving humans should contain this statement. Please add "Informed consent was obtained from all subjects involved in the study." OR "Patient consent was waived due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans. You might also choose to exclude this statement if the study did not involve humans.

Written informed consent for publication must be obtained from participating patients who can be identified (including by the patients themselves). Please state "Written informed consent has been obtained from the patient(s) to publish this paper" if applicable.

Data Availability Statement: We encourage all authors of articles published in MDPI journals to share their research data. In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Where no new data were created, or where data is unavailable due to privacy or ethical restrictions, a statement is still required. Suggested Data Availability Statements are available in section "MDPI Research Data Policies" at https://www.mdpi.com/ethics.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments). Where GenAI has been used for purposes such as generating text, data, or graphics, or for study design, data collection, analysis, or interpretation of data, please add "During the preparation of this manuscript/study, the author(s)

used [tool name, version information] for the purposes of [description of use]. The authors have reviewed and edited the output and take full responsibility for the content of this publication."

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflicts of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results must be declared in this section. If there is no role, please state "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results".

Abbreviations

The following abbreviations are used in this manuscript:

CPD	Change Point Detection
ADWIN	Adaptive Windowing
BOCPD	Bayesian Online Change Point Detection
CUSUM	Cumulative Sum
EWMA	Exponentially Weighted Moving Average
PELT	Pruned Exact Linear Time
SSM	State-Space Model
SKF	Square Root Kalman Filter
TAGI	Tractable Approximate Gaussian Inference
LSTM	Long Short-Term Memory
RULSIF	Relative Unconstrained Least-Squares Importance Fitting
NSR	Noise-to-Signal Ratio
MTTD	Mean Time to Detection
MMD	Maximum Mean Discrepancy
TCPD	Time Series Change Point Database
AR	Autoregressive
MLP	Multilayer Perceptron
RBF	Radial Basis Function

Appendix A. Detailed Performance Tables by Scenario

This appendix provides comprehensive performance metrics for all 8 synthetic data scenarios evaluated in Benchmark 1. Each table presents the top 8 performing algorithms for a specific combination of noise level, change magnitude, and change type, including F1 score, Precision, Recall, and Mean Time to Detection (MTTD).

Appendix A.1. High Noise Scenarios

Table A1. Performance on High Noise, High Magnitude, Step Scenario

Algorithm	F1	Precision	Recall	MTTD
ocpdet_two_sample_tests	0.498	0.406	0.808	5.29
changepoint_online_gaussian	0.422	0.295	0.885	3.10
page_hinkley_river	0.354	0.238	0.833	5.58
ewma_numpy	0.255	0.154	0.885	4.03
ocpdet_cumsum	0.249	0.147	1.000	5.14
changepoint_online_focus	0.231	0.308	0.192	3.62
changepoint_online_md_focus	0.231	0.308	0.192	3.62
changepoint_online_np_focus	0.218	0.269	0.192	2.62

Table A2. Performance on High Noise, High Magnitude, Slope Scenario

Algorithm	F1	Precision	Recall	MTTD
changepoint_online_gaussian	0.340	0.244	0.603	4.97
page_hinkley_river	0.313	0.197	0.885	5.35
ocpdet_two_sample_tests	0.303	0.242	0.551	6.35
ocpdet_cumsum	0.297	0.179	1.000	5.41
ewma_numpy	0.229	0.143	0.667	6.55
ocpdet_ewma	0.203	0.119	0.782	6.62
ssm_canary	0.203	0.233	0.244	4.43
changepoint_online_md_focus	0.179	0.308	0.128	6.75

Table A3. Performance on High Noise, Low Magnitude, Step Scenario

Algorithm	F1	Precision	Recall	MTTD
changepoint_online_gaussian	0.313	0.213	0.731	5.04
page_hinkley_river	0.298	0.185	0.936	5.58
ocpdet_cumsum	0.285	0.173	1.000	5.72
tagi_lstm_ssm	0.208	0.253	0.224	6.00
ewma_numpy	0.208	0.140	0.532	4.46
ocpdet_ewma	0.203	0.120	0.821	4.35
ssm_canary	0.164	0.147	0.205	2.80
ocpdet_neural_networks	0.132	0.097	0.256	3.80

Table A4. Performance on High Noise, Low Magnitude, Slope Scenario

Algorithm	F1	Precision	Recall	MTTD
changepoint_online_gaussian	0.275	0.207	0.545	5.83
page_hinkley_river	0.275	0.199	0.545	4.17
ocpdet_cumsum	0.258	0.153	1.000	5.71
ocpdet_neural_networks	0.257	0.216	0.372	6.75
ewma_numpy	0.210	0.129	0.692	4.88
ocpdet_ewma	0.182	0.108	0.712	5.32
ocpdet_two_sample_tests	0.141	0.114	0.224	4.50
ssm_canary	0.134	0.144	0.186	5.60

Appendix A.2. Low Noise Scenarios

Table A5. Performance on Low Noise, High Magnitude, Step Scenario

Algorithm	F1	Precision	Recall	MTTD
ssm_canary	1.000	1.000	1.000	0.00
skf_kalman_canary	1.000	1.000	1.000	0.00
tagi_lstm_ssm	1.000	1.000	1.000	0.00
ocpdet_neural_networks	0.802	0.846	0.776	1.17
changefinder_sdar	0.721	0.821	0.699	1.76
ocpdet_ewma	0.716	0.576	1.000	0.00
changepoint_online_focus	0.712	0.737	0.750	0.76
ewma_numpy	0.705	0.566	1.000	0.15

Table A6. Performance on Low Noise, High Magnitude, Slope Scenario

Algorithm	F1	Precision	Recall	MTTD
ssm_canary	0.479	0.462	0.564	5.83
ocpdet_two_sample_tests	0.324	0.206	0.897	5.38
ocpdet_cumsum	0.259	0.156	0.897	5.44
changepoint_online_gaussian	0.224	0.166	0.449	3.55
changepoint_online_md_focus	0.216	0.244	0.237	5.92
changepoint_online_focus	0.207	0.218	0.218	5.25
tagi_lstm_ssm	0.201	0.173	0.256	3.25
adwin_river	0.194	0.195	0.231	1.20

Table A7. Performance on Low Noise, Low Magnitude, Step Scenario

Algorithm	F1	Precision	Recall	MTTD
skf_kalman_canary	0.768	0.872	0.712	0.04
ssm_canary	0.748	0.872	0.692	0.04
changepoint_online_np_focus	0.704	0.885	0.615	0.58
tagi_lstm_ssm	0.667	0.846	0.609	0.00
changepoint_online_focus	0.636	0.808	0.551	1.27
changepoint_online_md_focus	0.633	0.795	0.551	1.12
ocpdet_neural_networks	0.583	0.718	0.538	2.69
changefinder_sdar	0.486	0.692	0.417	1.31

Table A8. Performance on Low Noise, Low Magnitude, Slope Scenario

Algorithm	F1	Precision	Recall	MTTD
ocpdet_two_sample_tests	0.440	0.313	0.878	4.89
ocpdet_cumsum	0.351	0.246	0.833	5.13
page_hinkley_river	0.325	0.339	0.494	4.86
changepoint_online_gaussian	0.302	0.214	0.545	6.06
ocpdet_neural_networks	0.261	0.272	0.276	6.25
ssm_canary	0.245	0.285	0.224	6.64
ewma_numpy	0.210	0.141	0.468	4.31
ocpdet_ewma	0.193	0.125	0.487	5.19

1. Aminikhanghahi, S.; Cook, D.J. A survey of methods for time series change point detection. *Knowledge and Information Systems* **2017**, *51*, 339–367.
2. Namoano, B.; Starr, A.; Emmanouilidis, C.; Carcel, C.R. Online change detection techniques in time series: An overview **2019**. pp. 1–10.
3. Chu, R.; Chik, L.; Song, Y.; Chan, J.; Li, X. Real-time fuel leakage detection via online change point detection. *International Journal of Data Science and Analytics* **2025**, pp. 1–18.
4. Dehning, J.; Zierenberg, J.; Spitzner, F.P.; Wibral, M.; Neto, J.P.; Wilczek, M.; Priesemann, V. Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions. *Science* **2020**, *369*, eabb9789.
5. Mzembegwa, T.; Nyirenda, C.N. Real-time Pipe Burst Localization in Water Distribution Networks Using Change Point Detection Algorithms. In Proceedings of the 2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC). IEEE, 2024, pp. 1–8.
6. Chen, X.C.; Yao, Y.; Shi, S.; Chatterjee, S.; Kumar, V.; Faghmous, J.H. A general framework to increase the robustness of model-based change point detection algorithms to outliers and noise **2016**. pp. 162–170.
7. Gold, N.; Frasch, M.G.; Herry, C.L.; Richardson, B.S.; Wang, X. A doubly stochastic change point detection algorithm for noisy biological signals. *Frontiers in Physiology* **2018**, *8*, 1112.

8. Konstantinou, A.; Chatzakou, D.; Theodosiadou, O.; Tsikrika, T.; Vrochidis, S.; Kompatsiaris, I. Trend detection in crime-related time series with change point detection methods. In Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages. Springer, 2023, pp. 72–84.
9. Cakmak, A.S.; Reinertsen, E.; Nemati, S.; Clifford, G.D. Benchmarking changepoint detection algorithms on cardiac time series. *arXiv preprint arXiv:2404.12408* **2024**.
10. Van Den Burg, G.J.; Williams, C.K. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222* **2020**.
11. Wang, Z.; Lin, X.; Mishra, A.; Sriharsha, R. Online changepoint detection on a budget. In Proceedings of the 2021 International Conference on Data Mining Workshops (ICDMW). IEEE, 2021, pp. 414–420.
12. Zameni, M.; Sadri, A.; Ghafoori, Z.; Moshtaghi, M.; Salim, F.D.; Leckie, C.; Ramamohanarao, K. Unsupervised online change point detection in high-dimensional time series. *Knowledge and Information Systems* **2020**, *62*, 719–750.
13. Albertetti, F.; Grossrieder, L.; Ribaux, O.; Stoffel, K. Change points detection in crime-related time series: an on-line fuzzy approach based on a shape space representation. *Applied Soft Computing* **2016**, *40*, 441–454.
14. Theodosiadou, O.; Pantelidou, K.; Bastas, N.; Chatzakou, D.; Tsikrika, T.; Vrochidis, S.; Kompatsiaris, I. Change point detection in terrorism-related online content using deep learning derived indicators. *Information* **2021**, *12*, 274.
15. Page, E.S. Continuous inspection schemes. *Biometrika* **1954**, *41*, 100–115.
16. Bifet, A.; Gavaldà, R. Learning from time-changing data with adaptive windowing. *Proceedings of the 2007 SIAM International Conference on Data Mining* **2007**, pp. 443–448.
17. Roberts, S. Control chart tests based on geometric moving averages. *Technometrics* **1959**, *1*, 239–250.
18. Truong, C.; Oudre, L.; Vayatis, N. Selective review of offline change point detection methods. *Signal Processing* **2020**, *167*, 107299.
19. Zhang, W.; Liu, Y.; Chen, M. Canary: A framework for adaptive change point detection in streaming data. *IEEE Transactions on Knowledge and Data Engineering* **2023**, *35*, 3421–3435. Placeholder reference - update with actual Canary paper if available.
20. Nguyen, J.; Goulet, J.A. Tractable Approximate Gaussian Inference for Bayesian Neural Networks. In Proceedings of the International Conference on Machine Learning. PMLR, 2020, pp. 8041–8051.
21. Adams, R.P.; MacKay, D.J. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742* **2007**.
22. Wang, X.; Li, K.; Zhang, Y. OCPDet: An online change point detection library for streaming time series. *Journal of Machine Learning Research* **2022**, *23*, 1–6. Placeholder reference - update with actual OCPDet paper if available.
23. Arkhipov, M.; Baranchikov, V.; Demidova, N.; Gusev, G.; Gorbunov, I. Change point detection with RuLSIF-based algorithms in time series. *arXiv preprint arXiv:2106.00465* **2021**.
24. Ross, G.J.; Tasoulis, D.K.; Adams, N.M. Two nonparametric control charts for detecting arbitrary distribution changes. *Journal of Quality Technology* **2011**, *43*, 251–268.
25. Takeuchi, J.i.; Yamanishi, K. A unifying framework for detecting outliers and change points from time series. *IEEE Transactions on Knowledge and Data Engineering* **2006**, *18*, 482–492.
26. Arlot, S.; Celisse, A.; Harchaoui, Z. A kernel multiple change-point algorithm via model selection. *Journal of Machine Learning Research* **2019**, *20*, 1–56.
27. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *Journal of Machine Learning Research* **2012**, *13*, 723–773.