Adversarial Density Ratio Estimation for Change Point Detection

Shreyas S
shreyshs@amazon.com
Amazon
Bengaluru, India

Prakash Mandayam Comar
prakasc@amazon.com
Amazon
Bengaluru, India

Sivaramakrishnan Kaveri
kavers@amazon.com
Amazon
Bengaluru, India

ABSTRACT

Change Point Detection (CPD) models are used to identify abrupt changes in the distribution of a data stream and have a widespread practical use. CPD methods generally compare the distribution of data sequences before and after a given time step to infer if there is a shift in distribution at the said time step. Numerous divergence measures, which measure distance between data distributions of sequence pairs, have been proposed for CPD [17, 20] and often the choice of divergence measure depends on the data used. Density Ratio Estimation (DRE) [18, 20] can be used to estimate a broad family of f-divergences, which includes widely used CPD divergences like Kullback-Leibler (KL) and Pearson, and thus DRE is a popular approach for CPD. In this work, we improve upon the existing DRE techniques for CPD, by proposing a novel objective that combines DRE seamlessly with adversarial sample generation. The adversarial samples allows for a robust CPD with DRE to track subtle changes in distribution, leading to a reduction in false negatives. We experiment on a wide variety of real-world, public benchmark datasets to show that our approach improves upon existing state-of-the-art (SoTA) methods, including DRE based CPD methods, by demonstrating an $\sim$ 5% increase in $F-$score.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

KEYWORDS

Change Point Detection, GANs, Density Ratio Estimation

ACM Reference Format:
Shreyas S, Prakash Mandayam Comar, and Sivaramakrishnan Kaveri. 2023. Adversarial Density Ratio Estimation for Change Point Detection. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615248

1 INTRODUCTION

Many user experiences on the internet like recommendations, advertising are now driven by Machine learning (ML) models and user interactions with these experiences changes with time. For example, user interests/activities evolve over time in a Recommender System, or the variation of traffic in web services. Change Point Detection (CPD), aims to detect points of change in an unknown probability distribution over a stream of data, which can be text, images, video etc. In diverse fields like Search/Recommender Systems/Advertising, it is normal for methods to assume stationarity, and hence CPD is a valuable tool to detect changes in user intents [25] and to adapt to user preferences in Recommender System and Advertising [11, 16, 24], monitor Web services [6] and flag DOS attacks/service failures.

In the present work, we consider a general time series setting, which is agnostic of the nature of data/channel employed. Any CPD method [1], detects the change in distribution by dividing the time series into window pairs around the candidate timestamp for a change point. Then it estimates a divergence measure between these window pairs, repeatedly for each timestamp, to infer the change points if any exist. The divergence metrics measure how apart two distributions are, based on the samples drawn from both the distributions. Popular divergence measures for CPD include KL divergence [21], Pearson Divergence [13], which belong to a general class of divergences called f-divergences. Any divergence in the family of f-divergences can be estimated by estimating the ratio of two probability density functions, $r(x) = \frac{p(x)}{q(x)}$ without even learning the distributions p, q directly, and is termed Density Ratio Estimation (DRE). The applicability of a divergence can vary based on the nature of time series, hence numerous DRE methods [15, 20] have been popular in CPD, because of the flexibility they provide in the choice of divergences.

However, DRE methods' performance pales in comparison to other state-of-the-art (SoTA) methods [4, 9], as they are unable to track small changes in distribution, which results in false-negatives. In this paper, we propose a novel objective for CPD, which integrates estimation of density ratio for CPD with the training of an adversarial generator. The adversarial model generates samples from a different distribution p^G, but p^G is still closer to the underlying data distribution p ($p^G \neq p$, but $p^G \approx p$), and the estimator is tasked to differentiate between samples drawn from p^G and p, leading to a reduction in false-negatives. To the best of our knowledge, we are the first to employ a generative adversarial model to estimate density ratio for CPD. The paper, [4] also uses an adversarial model, however it limits itself to a singular measure Mean Maximum Discrepancy (MMD), which is not widely applicable across datasets unlike the versatile f-divergences. Finally, we conclude by experimenting on benchmark, real-world, publicly available human annotated datasets and show that our proposed method exhibits an increase of $\sim$ 5% in F-score on an average, compared to the SoTA methods. We demonstrate a mean improvement of $\sim$ 13% in Recall over the DRE based CPD methods, which shows the benefit of adversarial training in reducing the false negatives.

2 PROBLEM SETTING AND RELATED WORK

We denote the input time series $\{x_t | t \in \mathcal{T} \text{ and } x_t \in \mathbb{R}^D\}$, drawn from input space $\mathcal{X}$ with the time horizon $\mathcal{T} = \{1, 2, \cdots, T\}$. The

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615248

CPD involves determining the time $t^* \in \mathcal{T}$ when an abrupt transition from distribution p to q occurs, i.e., $x_t \sim p$ if $t \leq t^*$ and $x_t \sim q$ if $t > t^*$. The distributions p, q are not known. To estimate the divergence measure empirically at each point t, two windows - the reference window, $X_t^r = \{x_{t-w_r}, \cdots, x_{t-1}\}$, of size w_r and the test window, $X_t^f = \{x_t, \cdots, x_{t+w_f-1}\}$, of size w_f, are constructed. Many traditional parametric methods like CUMSUM [2] and GLR [3], or the popular kernel based non-parametric methods like [10], [17], rely on repeated application of hypothesis tests on the window pairs to detect change points. However, parametric methods require strong assumptions on the form of p at the very least and non-parametric methods are very sensitive to the choice of kernel used. In recent literature, a state of the art deep learning based CPD method [4], employs a GAN like objective and uses MMD metric to detect change points. An autoencoder based architecture is found in [8] and it uses reconstruction loss as the CPD metric. Another SoTA method [9], uses contrastive coding to obtain embeddings for the time series and detect change points using cosine-distance. The paper [5], employs variational autoencoders for change point detection.

Density Ratio Estimation. The setting consists of two sets of observed samples $D_p = \{x_1^p, \cdots, x_m^p\} \sim p(x)$ and $D_q = \{x_1^p, \cdots, x_n^q\} \sim q(x)$, drawn from distributions p, q. Given D_p, D_q, we are interested in estimating the ratio $r(x) = p(x)/q(x)$ directly without explicitly learning the distributions p, q. Estimating the density ratio gives an estimate of the f-divergence between the distributions p, q. Typically, the estimator $\hat{r}$ has the following form [13], $\hat{r}(x;\theta) = \theta^T K(x,\cdot) = \theta^T \phi(x)$ i.e., the density ratio is governed by the parameters θ and RKHS kernel K, and ϕ its corresponding kernel space representation. Numerous kernel based techniques focus on estimating different divergences like KL [21], Pearson [13], Bregman [20], whereas, other DRE methods realize the kernel ϕ via neural network [15] or Gradient Boosted Trees [12]. Another stream of work in DRE, focussed on adapting GANs to be trained on f-divergences (or Bregman divergences, both end up having the same objective) [19, 22], with the common purpose to improve the accuracy of the model by generating adversarial samples. However, their work cannot be trivially extended to CPD.

3 PROPOSED WORK

In the present work, we employ a Deep Neural Network to estimate the f-divergences between the reference and test window distributions and, employ adversarial samples to detect even small changes in the test window distribution. We begin the section by recalling the definition of f-divergence measure $D_f(p,q) = \int_{\mathbb{R}^D} f\left(\frac{p(x)}{q(x)}\right) q(x)dx$ where f is a function such that $f(1) = 0$ (to ensure $D_f(p,q) = 0$ when $p = q$) and $f(x)$ is bounded for $x > 0$. Further, if we set, $f(x) = x \log x$ then we get KL divergence and for $f(x) = (x-1)^2$ we arrive at Pearson divergence. Given we have an estimator $\hat{r}$ for density ratio r, [20] furnishes the following estimator for f-divergence $D_f(p,q)$.

$$\hat{D}_f(p,q) = E_{x\sim p}\left[g(x)\right] - E_{x\sim q}\left[\partial f(\hat{r}(x))\right] \tag{1}$$

where $g(x) = \partial f(\hat{r}(x))\hat{r}(x) - f(\hat{r}(x))$ and ∂f denotes the partial derivative of f with respect to $\hat{r}$, i.e., if $f(\hat{r}(x)) = (\hat{r}(x)-1)^2$ then $\partial f(\hat{r}(x)) = 2(\hat{r}(x)-1)$. The following equation is the empirical version of (1)

$$\hat{D}_f(X_t^r, X_t^f, t) = \max_{\hat{r}} \left(\frac{1}{w_r} \sum_{x\in X_t^f} g(x) - \frac{1}{w_f} \sum_{x\in X_t^r} \partial f(\hat{r}(x)) \right) \tag{2}$$

Let G denote the generative model that generates the adversarial test window and $\hat{r}$ the density ratio estimator as the discriminator which tells apart the organic test window from the adversarial one. The algorithm Adv-DRE trains the generator and the discriminator together on the following objective (defer the details to Algorithm 1), to boost detectability of even the minor changes in the distribution p.

$$Q = Q^{min-max} + \lambda_1 Q^{reg} + \lambda_2 Q^{rec} \tag{3}$$

where $Q^{min-max}$ is the min-max objective on which both generator and ratio estimator are trained using ADAM optimiser. The Q^{reg} is regularization term that comes up due to the requirement that $D_f(p,p) = 0$ should be satisfied for any valid divergence. Q^{rec} is the minimization of the reconstruction error of the encoder-decoders (refer Fig.1) used to learn the embeddings for time series. Finally, for inference we compute $\hat{D}_f(X_t^r, X_t^f, t)$ using (2) and return it as CPD model score, which, when above a threshold (tuned through a peak-finding algorithm – see Sec. 4), is predicted to be a change point.

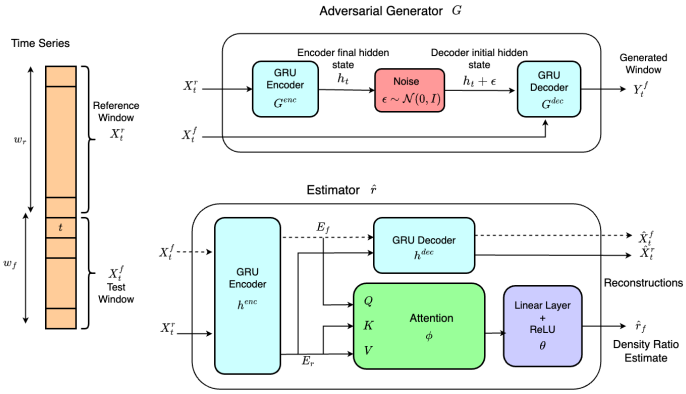

Figure 1: Proposed Adversarial Generator G and Estimator $\hat{r}$ Both employ Encoder-Decoder pair to embed time windows. The $\hat{r}$ applies Attention on the window pair, X_t^r, X_t^f to compute density ratio estimate $\hat{r}_f$. Dashed arrows indicate X_t^f (or adversarial window Y_t^f) is passed after X_t^r through the same Encoder-Decoder found in $\hat{r}$.

Design of Generator Model G. The Generator G is supposed to generate the adversarial distribution p^G such that it closely resembles p. To achieve this, we require the Generator to generate adversarial test window samples Y_t^f autoregressively using the reference samples X_t^r. This forces the estimator $\hat{r}$ to accurately track minor deviations from p. Therefore, from (1) we end up with the following min-max objective[1] for training the Generator-Discriminator network

$$Q^{min-max} = \min_G \max_{\hat{r}} \sum_{x\in Y_t^f} g(x) - \sum_{y\in X_t^f} \partial f(\hat{r}(y)) \tag{4}$$

[1]Now we note the apparent similarity between $Q^{min-max}$ and the f-GAN objective [19], however unlike f-GAN the divergence is being measured in the test window

Algorithm 1: Training of Adv-DRE

Input: C_{est} - No of iterations of Estimator per Generator update, N - Max iterations, Time series x_t for $t \in \mathcal{T}$, Constants λ_1 and λ_2, window sizes w_r, w_f

1 **for** $i \leftarrow 1$ **to** N **do**
2 **for** $j \leftarrow 1$ **to** C_{est} **do**
3 /* Estimator $\hat{r}$ Update loop */
4 Sample $\{X_t^r, X_t^f\}$ and generate $Y_t^f = G(X_t^r, X_t^f)$
5 Obtain the denisty ratio estimates and reconstructed windows tuples $(\hat{X}_f, R_f) = \hat{r}(X_t^r, X_t^f)$ and $(\hat{X}_r, R_r) = \hat{r}(X_t^r, X_t^r)$ and $(\hat{X}_f, R_g) = \hat{r}(X_t^r, Y_t^f)$
6 Minimize Q in (3) w.r.t $\hat{r}$
7 **end for**
8 /* Update Generator G */
9 Sample $\{X_t^r, X_t^f\}$ and generate $Y_t^f = G(X_t^r, X_t^f)$
10 Apply Estimator, $R_g, \hat{X}_f = \hat{r}(X_t^r, Y_t^f)$
11 Minimise $-Q^{min-max} + \lambda_2 ||Y_t^f - \hat{Y}_t^f||_F^2$ w.r.t G
12 **end for**
13 **Estimator** $\hat{r}(Z_r, Z_f)$
14 Get GRU embeddings $E_r, E_f = h^{enc}(Z_r), h^{enc}(Z_f)$
15 Apply Attention $A_f = Attention(E_f, E_r, E_r)$
16 Compute density ratio estimate $\hat{r}_f = \theta^T A_f$
17 where θ is linear layer followed by ReLU
18 Get reconstructions $\hat{Z}_r, \hat{Z}_f = h^{dec}(E_r), h^{dec}(Z_f)$
19 **return** $\hat{Z}_f, \hat{r}_f$
20 **Generator** $G(Z_r, Z_f)$
21 Sample $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
22 Get GRU hidden state $h_r = G^{enc}(Z_r)$
23 **return** $G^{dec}(Z_f, h_r + \epsilon)$

We note that (4) recovers the estimator (1) when the adversarial generator G is absent. The above objective completely does away with the reference window X_t^r, which we can safely assume to be assuredly drawn from p. So we incorporate it back in the objective Q by requiring that $\hat{D}_f(X_t^r, X_t^r, t)$ is small so that the constraint $D_f(p, p) = 0$ is satisfied. Otherwise, the estimator can assign arbitrary large values to X_t^r causing problems during inference (2). Therefore, we have

$$Q^{reg} = \min_{\hat{r}} \lambda_1 \sum_{x \in X_t^r} (\partial f(\hat{r}(x))(\hat{r}(x) - 1) - f(\hat{r}(x))) \quad (5)$$

To generate the adversarial window Y_t^f, we follow [4], wherein we utilize an autoregressive process p^G, which ensures that $p^G \approx p$. We employ GRU based encoder-decoder architecture (refer Fig. 1 and lines 24-26 in Alg.1) where the final hidden state $h_t = G^{enc}(X_t^r, 0)$ is corrupted with noise $\epsilon \sim \mathcal{N}(0, I)$. The corrupted hidden state $h_t + \epsilon$ is passed along with the organic test window X_t^f to the decoder to obtain the adversarial window $Y_t^f = G^{dec}(X_t^f, h_t + \epsilon)$.

Design of Estimator $\hat{r}$. We begin by passing the windows, X_t^r, X_t^f through a GRU-Encoder h^{enc} (refer Fig. 1) to get E^r, E^f, which are K dimensional embeddings for each timestamp in the windows. These embeddings are further passed through a GRU-Decoder h^{dec} to reconstruct the reference, test windows, given by $\hat{X}_t^r, \hat{X}_t^f$. We repeat the same for the generated samples (see lines 6-7 in Alg.1). Thus, we minimize the reconstruction loss as follows

$$Q^{rec} = \min_{\hat{r}} \left(||X_t^f - \hat{X}_t^f||_F^2 + ||X_t^r - \hat{X}_t^r||_F^2 \right) + \min_{G} ||Y_t^f - \hat{Y}_t^f||_F^2$$

We obtain the final density ratio estimate (refer lines 18-20 in Alg.1) as $\hat{r}_f(Z_t^r, Z_t^f) = \theta^T \phi(Z_t^r, Z_t^f)$, this is similar in spirit to earlier works [18]. However, we replace the kernel based embedding ϕ by an attention network $\phi(Z_t^r, Z_t^f) = Attention(Z_t^f, Z_t^r, Z_t^r)$, in order to capture if the window embeddings Z_t^f, Z_t^r are both drawn from the same distribution or otherwise. The attention mechanism is defined by $Attention(Q, K, V) = softmax(\frac{Q'K'^T}{\sqrt{d_k}})V'$ where input is queries Q, keys K and values V and d_k is the dimension of both queries and keys. The $Q' = QW^Q, K' = KW^K, V' = VW^V$, are corresponding projections applied on Q, K, V respectively. The θ is further realized as single linear layer followed by a RELU layer.

Dataset	Length T	Dimension D	No. of Change Points	No. of Annotators
Yahoo!	1432	1	11	1
HASC	11738	3	12	1
Bitcoin	774	1	4	5
Brent-Spot	500	1	6	5
Occupancy	675	4	9.6	5

Table 1: Descriptive statistics of the real-world datasets. For Turing datasets, the number of change points indicates the average change points reported per human annotator.

4 EXPERIMENTS

In this section, we present the evaluation of our proposed algorithm Adv-DRE against 5 competing baselines with respect to following real world, public datasets possessing both univariate and multivariate time series.

- **Yahoo**[7] - includes real time production traffic load data on Yahoo Web services
- **HASC**[14] - human activity tracking data gathered from gyrometers labelled into 6 activities like walking, running etc.
- **Turing Change Point Benchmark**[23] - a recently proposed suite of 37 datasets taken from various sources specifically for benchmarking CPD.

We select 3 large datasets from the Turing benchmark, namely a) Bitcoin - bitcoin daily prices b) Brent-Spot - crude oil prices of the oil company Brent c) Occupancy - room occupancy measured using temperature, humidity, and other related measures. All these datasets have labels obtained from human annotators and are described in the Table 1. We have chosen one representative sub-sequence from Yahoo, HASC datasets as followed in [4].

Evaluation Protocol and Metrics. We divide the time series into train, validation and test periods by considering a 50 : 20 : 30 split. We have padded the data splits so that no window spills over to another split. The considered algorithms, do not provide the change points directly, but they output a CPD score for each point

alone between generated samples $Y_t^f \sim p^G$ and X_t^f which can be drawn from either p (if there is no change in distribution) or q (if a shift in distribution has occured).

Algorithm	Yahoo			HASC			Bitcoin			Occupancy			Brent-Spot		
	AUC	F1	Recall	AUC	F1	Recall	AUC	F1	Recall	AUC	F1	Recall	AUC	F1	Recall
KLIEP	47.4	33.3	20.0	47.7	30.00	19.8	54.7	56.7	40.3	55.5	56.1	48.8	54.9	63.6	45.7
rUSLIF	51.4	35.2	22.5	49.6	32.4	20.4	57.0	58.2	41.40	59.6	61.6	55.3	56.1	64.9	47.2
Deep-DR	50.7	39.3	25.6	25.0	32.4	25.2	44.7	51.9	35.43	65.5	69.2	75.6	55.0	69.7	54.4
KL-CPD	55.4	56.8	38.6	50.1	43.3	33.3	62.5	56.7	41.2	71.8	77.0	76.4	63.17	73.5	59.2
TS-CP2	54.9	42.8	36.14	64.1	34.7	24.8	63.4	59.9	57.7	61.4	64.1	62.4	55.6	65.6	48.8
Adv-DRE (KL Div.)	59.4	57.1	40.2	68.4	47.1	35.7	63.0	62.4	48.1	69.4	81.2	70.4	70.0	76.9	63.6
Adv-DRE (Pearson Div.)	58.5	54.3	35.6	74.4	51.9	40.0	63.1	68.6	54.23	69.0	88.0	80.8	71.1	83.6	72.4
Adv-DRE ($\lambda_2 = 0$)	58.9	52.4	34.8	52.0	43.6	30.6	58.7	57.5	44.3	59.6	79.8	77.6	63.6	74.3	60.8
Adv-DRE (No Attention)	53.7	40.6	33.3	52.0	43.6	33.3	64.6	62.6	46.7	69.6	82.5.6	76.8	63.6	74.3	68.0

Table 2: Results of Experiments. We have marked in blue the winning method for each dataset and performance metric. The Pearson Div. variant of Adv-DREperforms best across all baselines and performs better than KL Div. variant except on Yahoo dataset. When $\lambda_2 = 0$, we observe a drop in metric, but still performs on par with many baselines. The 'No Attention' variant performs better than the non-adverserial DRE baselines, however it performs slightly worse than the SoTA KL-CPD or TS-CP2 on certain datasets which we attribute it to an inferior density ratio estimate in absence of attention.

in the time series. This score has to be further thresholded to obtain the change points. Therefore, we employ the AUC-ROC metric, averaged across annotators, to evaluate the score provided by each algorithm, which is agnostic of number of change points detected.

In order to obtain the change points from the output scores, we employ a simple peak finding procedure. The peak finding procedure, takes the scores of the algorithm, and applies a threshold to obtain a set of intervals over the time horizon, where values are above/below the threshold. In each of these intervals, it gives the timestamp corresponding to the maximum score as the change point. To obtain the threshold, we select the value corresponding to the maximum F1 score on the validation set. Given K annotators, set of predicted change points Y and annotated change points $\{Y_k\}_{k=1}^K$, we employ F1 metric $F = \frac{2PR}{P+R}$ and Recall R to evaluate the obtained change points. We borrow the notion of Precision P and Recall R as defined in the recent benchmark [23]. It is given by

$$P = \frac{TP(Y, Y^*)}{|Y|} \qquad R = \sum_{k=1}^{K} \frac{TP(Y, Y_k)}{|Y_k|}$$

where $Y^* = \cup_{k=1}^K Y_k$ and $TP(Y, X) = \{y \in Y | \exists x \in X, |x - y| \leq \tau\}$ and margin τ is set to 5. So we give a margin of 5 timestamps between the detected and true change points to be considered true positive. The Precision is measured against the union of all annotated change points, whereas Recall is averaged across each annotator. The reported metrics are averaged across 10 runs.

Choice of window size. We restrict the size of the reference and test windows $w_r = 25$, $w_f = 10$ to be set across all the algorithms to ensure fair comparison. In earlier evaluations like [9], which employ binary classification based P, R metrics, require an exact match between true and detected timestamps. Hence, they had to resort to experimenting with different test window sizes. However, since we use the margin τ based metrics, this is not required. Hence, a choice of test window size of 10 captures a margin of 5 on both sides of a true/predicted timestamp. Moreover, a better performance on a smaller test window size w_f, demonstrates detection of minor shifts in distribution.

Baselines. We consider recently proposed, SoTA methods KL-CPD[4], TS-CP2[9] and DRE based CPD methods rUSLIF[18], KLIEP[21], Deep-DR[15] as competing baselines. The rUSLIF, KLIEP are kernel based methods whereas Deep-DR uses deep neural networks. We tune the hyperparameters for each of the methods by employing a grid of values, $\{0.01, 0.1, 1, 10\}$ and selecting the hyperparameter having the highest AUC on the validation set. We perform ablation with 4 variants of the algorithm with i) set $\lambda_2 = 0$ to demonstrate the importance of the regularization ii) set $f = x \log x$ KL divergence iii) set $f = (x-1)^2$ Pearson (PE) divergence iv) Dropping the Attention network.

Results. We observe a significant improvement in all the 4 variants of Adv-DRE over other DRE methods KLIEP, rUSLIF, Deep-DR across all metrics and datasets, with a significant improvement of $\sim$ 13% in recall or a reduction in false negatives. This demonstrates Adv-DRE is capable of capturing small shifts in distribution. Also, with respect to AUC metric Adv-DRE variants outperform other methods, except in Bitcoin, Occupancy, where we are a close second. We note that AUC metric is agnostic of threshold/number of change points, but $F1$ is a better indicator of performance post thresholding, and we show consistent improvement in the $F1$ metric over the SoTA methods KL-CPD, TS-CP2 whilst maintaining a high recall.

Ablation. We note that our PE divergence variant beats the KL divergence variant, except for Yahoo dataset. This can be corroborated from [18], who find PE divergence is a better measure because of its robustness. Both the variants with $\lambda_2 = 0$ and 'No Attention', struggle with respect to divergence based variants, this clearly highlights the importance of Q^{reg} and Attention on performance. Surprisingly, we find variants with $\lambda_2 = 0$ and 'No Attention', perform better than other density ratio based methods, which we can attribute to the adversarial samples that boost the performance of the model. The 'No Attention' variant performs on par or slightly worse compared to the SoTA KL-CPD or TS-CP2 on certain datasets. This which we attribute it to an inferior density ratio estimate by employing just a linear layer and highlights the importance of attention for obtaining a good estimate of density ratio. We also note that, both KL and PE variants beat KL-CPD significantly on Bitcoin and Brent-Spot, implying that MMD based CPD might not be optimal for many datasets. Therefore, Adv-DRE which provides the flexibility of choosing from a family of divergences is of more practical value.

REFERENCES

[1] Samaneh Aminikhanghahi and Diane J. Cook. 2017. A Survey of Methods for Time Series Change Point Detection. *Knowl. Inf. Syst.* 51, 2 (may 2017), 339–367.

[2] Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. 2014. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[3] Michèle Basseville and Igor V. Nikiforov. 1993. *Detection of Abrupt Changes: Theory and Application.* Prentice-Hall, Inc., USA.

[4] Wei-Cheng Chang, Chun-Liang Li, Yiming Yang, and Barnabás Póczos. 2019. Kernel change-point detection with auxiliary deep generative models. *International Conference on Learning Representations (ICLR).*

[5] Sourav Chatterjee. ICML 2021 Time Series Workshop. Changepoint Detection using Self Supervised Variational AutoEncoders.

[6] Yingying Chen, Ratul Mahajan, Baskar Sridharan, and Zhi-Li Zhang. 2013. A Provider-Side View of Web Search Response Time. *SIGCOMM Comput. Commun. Rev.* 43, 4 (aug 2013), 243–254.

[7] Yahoo Research Webscope Dataset. 2011. S5 - A Labeled Anomaly Detection Dataset. https://webscope.sandbox.yahoo.com/.

[8] Tim De Ryck, Maarten De Vos, and Alexander Bertrand. 2021. Change Point Detection in Time Series Data Using Autoencoders With a Time-Invariant Representation. *IEEE Transactions on Signal Processing* 69 (2021), 3513–3524.

[9] Shohreh Deldari, Daniel V. Smith, Hao Xue, and Flora D. Salim. 2021. Time Series Change Point Detection with Self-Supervised Contrastive Predictive Coding. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3124–3135.

[10] Zaïd Harchaoui, Francis Bach, and Éric Moulines. 2008. Kernel Change-Point Analysis. In *Proceedings of the 21st International Conference on Neural Information Processing Systems* (Vancouver, British Columbia, Canada) *(NIPS'08)*. Curran Associates Inc., Red Hook, NY, USA, 609–616.

[11] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2014. Context Adaptation in Interactive Recommender Systems. In *Proceedings of the 8th ACM Conference on Recommender Systems* (Foster City, Silicon Valley, California, USA) *(RecSys '14)*. Association for Computing Machinery, New York, NY, USA, 41–48.

[12] Mikhail Hushchyn and Andrey Ustyuzhanin. 2021. Generalization of change-point detection in time series data based on direct density ratio estimation. *Journal of Computational Science* 53 (2021), 101385.

[13] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. 2009. A Least-Squares Approach to Direct Importance Estimation. 10 (dec 2009), 1391–1445.

[14] Nobuo Kawaguchi, Nobuhiro Ogawa, Yohei Iwasaki, Katsuhiko Kaji, Tsutomu Terada, Kazuya Murao, Sozo Inoue, Yoshihiro Kawahara, Yasuyuki Sumi, and Nobuhiko Nishio. 2011. HASC Challenge: Gathering Large Scale Human Activity Corpus for the Real-World Activity Understandings. *ACM International Conference Proceeding Series*, 27.

[15] Haidar Khan, Lara Marcuse, and Bülent Yener. 2019. Deep density ratio estimation for change point detection. *CoRR* abs/1905.09876 (2019). http://arxiv.org/abs/1905.09876

[16] Chuanhao Li, Qingyun Wu, and Hongning Wang. 2021. When and Whom to Collaborate with in a Changing Environment: A Collaborative Dynamic Bandit Solution *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1410–1419.

[17] Shuang Li, Yao Xie, Hanjun Dai, and Le Song. 2015. M-Statistic for Kernel Change-Point Detection. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc.

[18] Song Liu, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. 2013. Change-Point Detection in Time-Series Data by Relative Density-Ratio Estimation. *Neural Netw.* 43 (jul 2013), 72–83.

[19] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NIPS'16)*.

[20] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2011. Density Ratio Matching under the Bregman Divergence: A Unified Framework of Density Ratio Estimation. *Annals of the Institute of Statistical Mathematics* 64 (10 2011).

[21] Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. 2008. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics* 60 (02 2008), 699–746.

[22] Masatosi Uehara, Issei Sato, Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. 2017. b-GAN: Unified Framework of Generative Adversarial Networks. https://openreview.net/forum?id=S1JG13oee

[23] G. J. J. Van den Burg and C. K. I. Williams. 2020. An Evaluation of Change Point Detection Algorithms. *arXiv preprint arXiv:2003.06222* (2020).

[24] Qingyun Wu, Naveen Iyer, and Hongning Wang. 2018. Learning Contextual Bandits in a Non-Stationary Environment. In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 495–504.

[25] Shengyao Zhuang and Guido Zuccon. 2021. How Do Online Learning to Rank Methods Adapt to Changes of Intent? *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 911–920.