Uncertainty Estimation for Change Point Detection models

Alexander Stepikin [1] **Maria Kovaleva** [1] **Alexander Kukharskii** [1]

Abstract

Change Point Detection (CPD) aims to identify moments of abrupt data distribution shifts in sequential data. While there exist deep learning CPD models, they typically are not able to provide any confidence in their predictions which is often required by real-world applications. In this project, we study ensembles of deep change point detectors and develop two ways of taking into account uncertainty in the predicted CP probabilities to produce robust estimates of the change points. Experiments conducted on synthetic and real-world datasets suggest that small ensembles of deep detectors outperform single-model CPD baselines. Uncertainty-aware aggregation of the change point scores obtained by an ensemble with CUSUM-statistic is proven to be beneficial in case of semi-structured high-dimensional data, such as video clips with explosions. The proposed rejection-based procedure helps to decrease the amount of false alarms, thus, optimizing one of the principled CPD metrics.

GitHub repository: InDiD-with-uncertainty
Presentation file: UE for CPD

1. Introduction

Change points (CPs) in sequential data are moments of sudden disorders which are typically modelled as shifts of the distribution the observations follow. These breaks in data streams could be a signal that the nature of the underlying process has changed, which may be caused by an emergency situation. Such changes occur in different real-world scenarios: from industrial production control (Shewhart, 1931; Page, 1954) and well drilling (Romanenkova et al., 2020) to health monitoring systems (Malladi et al., 2013; Yang et al., 2006) and financial data analysis (Spokoiny, 2009; Lavielle & Teyssière, 2007). Thus, avoid potential monetary or even human losses, change points need to be detected as fast and accurately as possible.

In addition, for the sake of reliability, the industry often requires the models not only to produce point estimates of the target values but confidence intervals instead. This means that the predictions of a robust model should be equipped with a measure of how much this model is certain about its outputs. Consequently, researchers pay more and more attention to the development of Uncertainty Estimation (UE) or Quantification (UQ) techniques (Abdar et al., 2021; Chung et al., 2021). This general problem applies to CPD as well. Moreover, for this task, UE may play an even more significant role as it might give insights into how CP detectors actually deal with new unknown samples of data.

In this work, we study and evaluate small ensembles of deep change point detectors. The main contributions of the project are as follows:

- We consider the point-wise standard deviation of CP scores predicted by such ensembles as the most straightforward uncertainty measure for this task;
- We propose an advanced CUSUM-like procedure for the aggregation of models' predictions that takes into account obtained uncertainty estimates;
- We develop a basic approach for CPD with rejection which aims to decrease the number of False Alarms.

The proposed methods help to improve CPD quality significantly for both synthetic and real-world datasets compared to single-model baselines.

2. Related Work

2.1. Change Point Detection

Change point detection is a well-studied problem in terms of mathematical statistics and machine learning. Under strong theoretical assumptions, it is proven to have optimal solutions obtained by CUSUM and Shiryaev-Roberts statistics (Shiryaev, 2017; Tartakovsky & Moustakides, 2010). However, these online procedures underperform or are not applicable in case of real high-dimensional data, which cannot be processed without sophisticated models.

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Alexander Stepikin <alexander.stepikin@skoltech.ru>.

Final Projects of the Machine Learning Course, Skoltech, Moscow, Russia, 2023.

Another class of CPD methods are offline algorithms aimed at finding an optimal partition of the input signal to the segments split by the CPs. For example, they include Binary and Bottom-Up segmentation procedures, as well as the PELT dynamic programming algorithm. These and the other classic CPD methods were thoroughly evaluated on simple low-dimensional data (van den Burg & Williams, 2020), and the study shows that they still can be ahead of more sophisticated approaches, like KernelCPD (Harchaoui et al., 2009), if the hyper-parameters are selected properly. We refer the reader to the papers (Truong et al., 2020; Aminikhanghahi & Cook, 2017) for an in-depth review of these CPD methods.

However, the performance of classical CPD methods mentioned above is restricted as they are not able to learn efficient representations of complex real-world time series. That is where machine learning and deep learning approaches come into play. A couple of the straightforward DL models for CPD were presented (Hushchyn et al., 2020) where authors utilize the likelihood ratio to compare representations obtained by neural networks. The paper (Chang et al., 2018) proposes KL-CPD model of a complicated GAN-like architecture that optimizes the power of the specific two-sample test by training a kernel w.r.t. the Maximum Mean Discrepancy distance (Li et al., 2015). Another recent idea is to apply self-supervised techniques, like Contrastive Predictive Coding (Oord et al., 2018), to CPD. This resulted in TS-CP2 model (Deldari et al., 2021) that detects CPs by computing cosine distance between the embeddings of the subsequent windows sliding through the sequence. Finally, the authors of (Romanenkova et al., 2022) propose to train LSTM-based (Hochreiter & Schmidhuber, 1997) sequence-to-sequence networks with a principled CPD-specific loss function to predict change point probabilities.

2.2. Uncertainty estimation

Uncertainty Estimation aims to quantify the models' confidence in their predictions in order to provide interpretable and more robust results. There exist three types of uncertainty: aleatoric (uncertainty in the data), epistemic (uncertainty in the model) and the mixed one (Abdar et al., 2021), which we assume to be of particular interest for the CPD task.

One way to quantify uncertainty is to use the methods of Bayesian statistics, e.g. Bayesian linear regression or Variational Inference (Ganguly & Earp, 2021) which, by their nature, do not produce point estimates of the target values but rather infer the posterior distribution over these parameters, thus allowing us to obtain confidence intervals. However, these approaches may be restrictive as we often need to make assumptions about the models.

Another simple approach for UE is to fit ensembles of slightly different models for the same task and, afterwards, use the standard deviation of their predictions as the measure of uncertainty (Lakshminarayanan et al., 2017; Liu et al., 2019). To ensure the models' variety, they are typically initialized with different weights, trained on different bootstrapped subsamples of the initial train set, or have different hyper-parameters. This project deals with the second (ensembling) approach for UE.

3. Algorithms and Models

In this section, we describe the main approaches investigated in the project as well as the baseline model we refer to.

3.1. CPD problem statement

Let $X^{1:T} = \{\mathbf{x}_t\}_{t=1}^{T}$, $\mathbf{x}_t \in \mathbb{R}^D$, denote a multidimensional random sequence. The sequence can be finite or infinite with $T \leq \infty$. Suppose there exists a random moment $1 \leq \theta \leq T$ such as, $\forall t < \theta$ the random variable $\mathbf{x}_t$ follows a "normal-data" distribution f_∞, and $\forall t \geq \theta$ the distribution of $\mathbf{x}_t$ switches to the "abnormal" one f_0. In this case, θ is called a change point (CP). Change point detectors aim to identify such situations quickly and accurately.

Note that, in this case, we deal with "at-most-one-change" (AMOC) assumption. However, it can be extended to the scenario of multiple CPs as well.

3.2. Sequence-to-sequence binary classification baseline

As a baseline deep change point detector, we use the binary classification model (BCE) presented in (Romanenkova et al., 2022). In this approach, CPD problem is treated as an instance of a sequence-to-sequence binary classification task meaning that each time step t is classified, whether it has occurred before or after the actual change point θ.

In more detail, the authors consider a dataset of observations $D = \left\{\left(X_1^{1:T}, \theta_1\right), ..., \left(X_N^{1:T}, \theta_N\right)\right\}$ labelled with the true change points $\theta_i \in \{1, \ldots, T\}$. The proposed parametric model $f_{\mathbf{w}}$ takes an input sequence $X_i^{1:T}$ and outputs a series of CP probabilities $\{p_t^i\}_{t=1}^{T}$ with $p_t^i = f_{\mathbf{w}}(X_i^{1:t})$ based only on the information $X_i^{1:t} = \{\mathbf{x}_{ij}\}_{j=1}^{t} \subset X_i$ available up to the current moment t. Thus, the model works in an online mode. The value p_t^i is the estimate of the probability that a change point has already occurred in a sequence X_i by the moment t.

To make a prediction for the i-th sequence, this model warns about a CP the first time τ when the estimated probability p_τ^i exceeds a pre-selected alarm threshold $s \in (0, 1)$. Consequently, the detection quality significantly depends on the choice of the value of s.

The authors choose recurrent neural networks as the core architecture to process sequential data. In this interpretation,

the model is trained with a standard binary cross-entropy loss function. For a single sequence, this loss is given by:

$$BCE(l,p) = -\frac{1}{T}\sum_{t=1}^{T}\left(p_t \log l_t + (1-p_t)\log\left(1-l_t\right)\right), \tag{1}$$

where $l = \{l_1, l_2 \ldots, l_T\}$ are the true CP labels ($l_t = 0$ for $t < \theta$ and $l_t = 1$ for $t \geq \theta$); $p = \{p_1, p_2, \ldots, p_T\}$ are the predicted CP probabilities.

3.3. Ensembles of deep change point detectors

In this work, we study small ensembles of deep change point detectors $\{f_{\mathbf{w}_i}\}_{i=1}^{K}$. As a base model, we use seq2seq BCE models described in Section 3.2. Once an ensemble is trained, for each test sequence $X^{1:T}$, we obtain a set of K sequences $\left\{p_i^{1:T}\right\}_{i=1}^{K}$ with the predicted CP scores. As the final ensemble prediction, we either take the point-wise mean $\mu^{1:T}$ of these sequences, or the series of q-th quantiles $\nu_q^{1:T}$ of these empirical distributions for some probability $q \in (0,1)$. The points-wise standard deviations $\sigma^{1:T}$ serve as the basic uncertainty estimates for an ensemble.

3.4. Uncertainty-aware change point scores aggregation

In this project, we develop a CUSUM procedure for the uncertainty-aware aggregation of the CP scores obtained by an ensemble. CUSUM-statistic (Shiryaev, 2017) is a widely used for change point detection algorithm, which involves the calculation of a cumulative sum of some process x_t:

$$S_0 = 0, S_{t+1} = max\left(0, S_t + x_t\right) \tag{2}$$

The process signals about a change point when the cumulative sum reaches a pre-selected threshold value.

Our goal is not only to detect the growth in mean predictions of the ensemble but also to make it confident. Thus, in our work, we use the differences in mean predictions of the ensemble prediction divided by their standard deviations as the process x_t:

$$x_t = \left(\mu_t - \mu_{t-1}\right)/\sigma_t \tag{3}$$

Thereby, if the ensemble predictions grow on average, but their variance is large, then the CUSUM does not detect the change point. The change point is detected only when the predicted CP scores grow themselves, and the model is confident about it. An example of a situation in which this approach can be beneficial is presented in Figure 1.

3.5. Change Point Detection with Rejection

We develop another basic procedure that takes into account uncertainty estimates to produce more accurate predictions of the change points – CPD with rejection. This approach

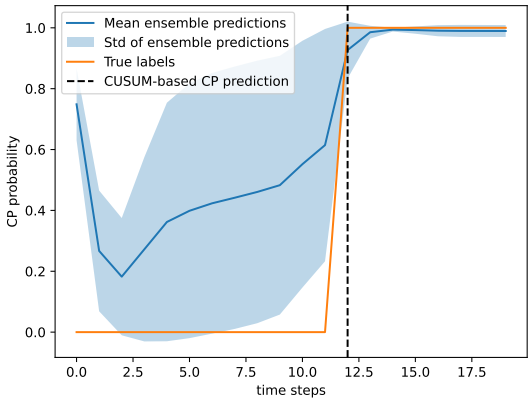

Figure 1. Example of CUSUM-based prediction for the HAR dataset.

is inspired by the works on classification with rejection option (Herbei & Wegkamp, 2006; Bartlett & Wegkamp, 2008). Once we have obtained average ensemble CP scores $\{\mu_t\}$ and their standard deviations $\{\sigma_t\}$, we can modify the decision-making rule in the following way. The model raises the alarm the first time moment when the average CP score exceeds a pre-selected threshold value s and the uncertainty in this prediction is low enough, i.e. the standard deviation is lower than another threshold value r:

$$\tau = \min\left\{t\colon \mu_t > s \text{ AND } \sigma_t < r\right\}. \tag{4}$$

As before, probability threshold s can be varied from 0 to 1, while ranges for threshold r can differ from one dataset to another. In our experiments, we gradually increase r to obtain an optimal value for each particular dataset.

Figure 2 justifies the relevance of this approach as, intuitively, it should decrease the number of False Alarms raised by an ensemble.

4. Experiments and results

In this section, we describe the experiments conducted in the project. We start with the evaluation pipeline, including the datasets and the metrics. Then, we outline the experimental setup and, finally, report the results obtained. The implementation details, such as the models' architecture and training parameters, are given in Appendix D.

4.1. Datasets

Following (Romanenkova et al., 2022), we evaluate proposed CPD methods on both synthetic and real-world data. In particular, we use four datasets described below. The samples may be downloaded from the cloud[1]. Synthetic 1D

[1] https://disk.yandex.ru/d/_PQyni3AhyLu5g

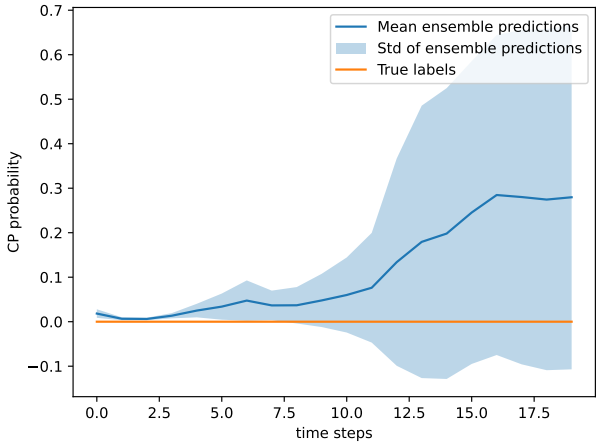

Figure 2. Example of the ensemble predictions for the HAR dataset that proves relevance of the CPD-with-rejection approach.

dataset is generated on the run.

Synthetic 1D. This dataset includes sequences of Gaussian random variables with a fixed variance and a random change in mean from $\mu_1 = 1$ to $\mu_2 = \text{random}(1, \dots, 100)$ at a random time. It also contains an equal amount of sequences without CPs resulting in a total dataset size of 1000. The length of each sequence is equal to 128.

MNIST. The dataset consists of sequences of MNIST (LeCun & Cortes, 2010) images with smooth transitions from one digit to another and without them. To generate data, the authors use a Conditional Variational Autoencoder (Sohn et al., 2015). Each "abnormal" sequence starts with a digit "4", which further transfers into the digit "7" at a random time. For "normal" sequences, both digits belong to the same class ("4"). This dataset is also balanced ,meaning that the amounts of sequences with and without CPs are equal. The length of each sequence is 64, and every single observation has a size of 28×28 pixels.

Human Activity recognition. As the first real-world dataset, we use the USC-HAD (Zhang & Sawchuk, 2012) database. Here, the sequences contain measurements of human-wearable devices, such as heart rate monitors, accelerometers, etc. Consequently, CPs in them correspond to the change in the type of human activity: "walk", "run", "sleep", etc. This dataset is imbalanced, i.e. almost 85% of the sequences include CPs. The total number of human activity types is 12. The length of each sequence is 20 time steps, and every single observation is a numerical vector of size 28.

Explosion. Finally, we use high-dimensional semi-structured video data. In this dataset, CPs correspond to any explosions and fire that occurred inside or outside shoot by CCTV cameras. "Normal" videos do not include any emergencies. Real-world videos were taken from the UCF-Crime anomaly detection dataset (Sultani et al., 2018), the CPD markup for them was provided (Romanenkova et al., 2022) and is available online. In the train set, the amounts of videos with and without CPs are equal, and the test set is imbalanced, with only $\approx 5\%$ of videos being "abnormal". Every clip has a length of 16 frames, and every frame is a tensor of size $240 \times 320 \times 3$.

4.2. Metrics

While change point detectors can be evaluated by standard classification metrics with a special interpretation of the elements of the confusion matrix, there exist task-specific CPD metrics as well. Following the evaluation pipeline of (Romanenkova et al., 2022), we use the following set of quality metrics.

4.2.1. Classification-based metrics.

Let θ be the true CP and τ denote the predicted one. Note that both θ and τ can be equal to T, meaning that the CP has not occurred in a sequence. We distinguish for different situations. The prediction is considered to be: (1) True Positive (TP) if $\theta \leq \tau$; (2) True Negative (TN) if $\theta = \tau = T$; (3) False Positive (FP) if $\theta > \tau$; (4) False Negative (FN) if $\theta < \tau = T$. Given the amounts of TP, TN, FP and FN predictions, we compute the standard F1-score: $F1 = \frac{TP}{TP+0.5(FP+FN)}$. Note that there is an alternative and more common approach (van den Burg & Williams, 2020) – to call predictions TP if $|\tau - \theta| < m$ for some margin m.

4.2.2. Principled CPD metrics.

The two intuitive task-specific CPD metrics are the Detection Delay given by $\max(\tau - \theta, 0)$ and the Time to a False Alarm equal to τ, if $\tau < \theta$, and T, otherwise. We average these values over the dataset to obtain *Mean DD* and *Mean Time to FA* metrics, respectively.

As mentioned in 3.2, the predictions of BCE model depend on the choice of the alarm threshold s. To evaluate an overall performance of the model, the authors of (Romanenkova et al., 2022) propose to vary the values of s from 0 to 1 and obtain different trade-offs between Mean DD and Mean Time to FA. Then, the Area under the Detection Curve (AUDC) is considered to be one of the main model's quality metrics.

Finally, we use the segmentation-based Covering metric (van den Burg & Williams, 2020) given by:

$\text{Covering}(G, G') = \frac{1}{T}\sum_{A\in G}|A| \cdot \max_{A'\in G'}\frac{|A\cap A'|}{|A\cup A'|}$, where G and g' and the partitions of the time interval $[0, T]$ made by the true and the predicted CPs, respectively. High values of Covering indicate good performance of a model.

4.3. Main results

In this section, we describe and discuss the main results of this paper. First, we discuss the average and the quantile-based performance of the ensembles. Second, we present the results for our uncertainty-aware CUSUM-aggregation method. Finally, we demonstrate our findings for CPD with rejection.

We trained ensembles of 10 baseline seq2seq BCE models with different initialization of weights. In the main experiments, the train set was the same for all the models in an ensemble. The main results are summarized in Table 1. For ensemble quantile predictions, we report the results for the quantile that maximizes the F1-score. Similarly, for CUSUM, and rejection-based methods, we choose the metrics for the hyper-parameters that maximize F1-score as well. For a more detailed description of the hyper-parameters search, please, go to Appendix D.

We also provide results of the supplementary experiments, such as training ensemble models on small bootstrapped subsamples of the train set, in Appendix C.

4.3.1. Ensembles of deep change point detectors

Based on Table 1, the following conclusions considering the ensembles' performance can be made.

1. Generally, small ensembles of deep models outperform single-model BCE baseline in terms of all considered CPD metrics for MNIST, HAR and Explosion datasets. For toy 1D Synthetic data, all the approaches show perfect results meaning that there is no room for improvement over the baseline.

2. Advances in performance metrics for quantile-based predictions over the "mean" ones might be due to the few models in the ensembles. In this case, one base model in the ensemble might have produced outlier predictions which significantly impacted on the mean value of the ensemble.

3. For the Human Activity Recognition dataset ensembles do not demonstrate as significant enhancement as for MNIST or Explosion data. It might be because of rapid changes in data in HAR, while the changes are smoother in MNIST and Explosion datasets.

4.3.2. CP scores aggregation with CUSUM

Full results of the CUSUM-based method in terms of the F1-score and the Covering metric are presented in Figure 3.

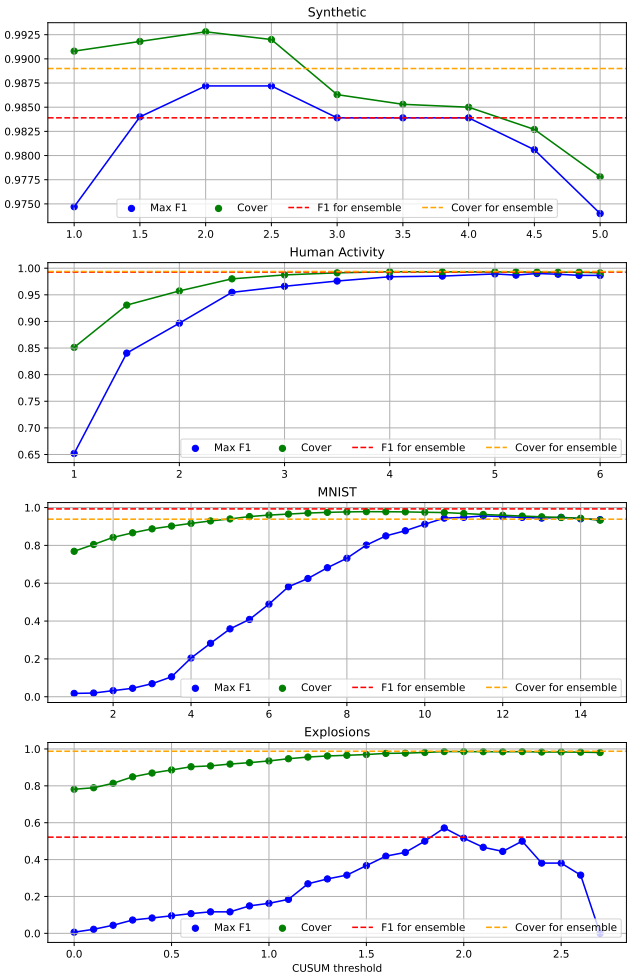

Figure 3. Dynamics of main quality metrics for our uncertainty-aware CUSUM-based approach evaluated for all the 4 datasets.

Note that we do not report AUDC metric for this approach as the alarm threshold for the CUSUM procedure is not bounded between 0 and 1, thus, making it not comparable with the standard one used for other experiments.

Based on Figure 3 and Table 1, the following conclusions can be made.

1. Uncertainty-aware CUSUM aggregation of CP scores with an optimal alarm threshold helps to improve principled CPD metrics for high-dimensional datasets, i.e. Mean DD – for sequences of MNIST images and Explosion; Mean Time for a FA – for MNIST and HAR; Covering – for MNIST.

Table 1. Main quality metrics for the considered CPD methods. ↑ marks metrics that should be maximized, ↓ – minimized. "NA" indicates that we do not compute AUDC metric for the models with CUSUM aggregation of change point scores. Best metrics values are highlighted with **bold** font.

Method	AUDC ↓	Mean Time to FA ↑	Mean DD ↓	Max F1 ↑	Covering ↑
1D Synthetic data					
Single BCE model	**606.00**	94.49	**0.50**	**0.9904**	**0.9941**
Ensemble mean	639.73	95.01	1.21	0.9839	0.9880
Ensemble quantile (best)	627.07	**96.69**	0.84	0.9872	0.9906
CUSUM (best)	NA	94.68	0.67	0.9872	0.9928
Rejection (best)	642.30	95.19	1.41	0.9806	0.9861
Sequences of MNIST images					
Single BCE model	237.94	44.94	3.37	0.9862	0.9120
Ensemble mean	175.32	46.87	2.20	0.9893	0.9386
Ensemble quantile (best)	**164.40**	47.21	1.67	**0.9965**	0.9510
CUSUM (best)	NA	**47.51**	**1.41**	0.9559	**0.9663**
Rejection (best)	209.98	46.94	2.47	0.9893	0.9320
Human Activity Recognition					
Single BCE model	43.22	11.00	0.20	0.9896	0.9851
Ensemble mean	44.95	11.07	**0.10**	0.9927	**0.9938**
Ensemble quantile (best)	44.25	11.10	0.17	**0.9948**	0.9898
CUSUM (best)	NA	**11.28**	0.28	0.9900	0.9927
Rejection (best)	**38.76**	10.58	0.96	0.9927	0.9930
Explosion					
Single BCE model	0.82	14.74	0.13	0.3094	0.9728
Ensemble mean	0.50	**15.79**	0.23	0.5217	**0.9876**
Ensemble quantile (best)	**0.39**	15.78	0.23	0.5217	**0.9876**
CUSUM (best)	NA	15.63	**0.11**	**0.5714**	0.9858
Rejection (best)	0.88	15.25	0.20	0.3889	0.9751

2. This approach also significantly increases F1-score for the Explosion dataset compared to the basic and quantile ensemble results. This might be explained as this dataset is the most sophisticated one. Thus, the models' uncertainty is typically very high and plays the most critical role in the predictions.

3. Optimal alarm thresholds for the CUSUM procedure differ significantly for various datasets meaning that this is a crucial hyper-parameter that should be selected carefully.

4.3.3. CPD WITH REJECTION

Full results for the CPD-with-rejection approach in terms of F1-score, Covering and AUDC metrics are presented in Figures 4 and 5. Based on Figures and Table 1, the following conclusion can be made.

1. The only advance the basic CPD-with-rejection approach gives is the AUDC metric for the HAR dataset, which decreased significantly.

2. In general, the basic CPD-with-rejection approach does not improve the detection quality in terms of most of the metrics. This means the procedure and the decision rule should probably be modified in further experiments.

5. Conclusion and limitations

In this project, we have studied ensembles of deep change point detectors and evaluated them on both synthetic and high-dimensional real-world data. Using the standard deviation of the models' outputs to measure the ensemble's confidence in its predictions, we develop 2 uncertainty-aware approaches for CPD: CUSUM-aggregation of change point scores and CPD with rejection.

Our experiments have proven that even small ensembles significantly outperform single-model deep CPD baselines in terms of the main CPD quality metrics. Moreover, our CUSUM procedure improves task-specific metrics over the standard ensemble predictions in many cases, including the Explosion video dataset.

The main limitation of this approach is the necessity to train several deep learning models, which is time and computationally costly.

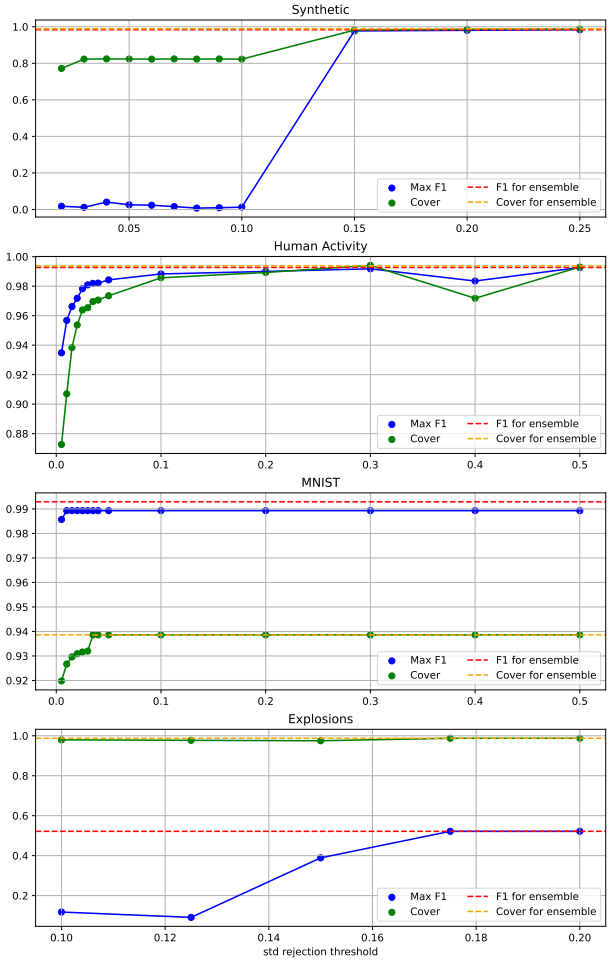

Figure 4. Dynamics of F1 and Covering metrics for our CPD-with-rejection approach evaluated for the all 4 datasets.

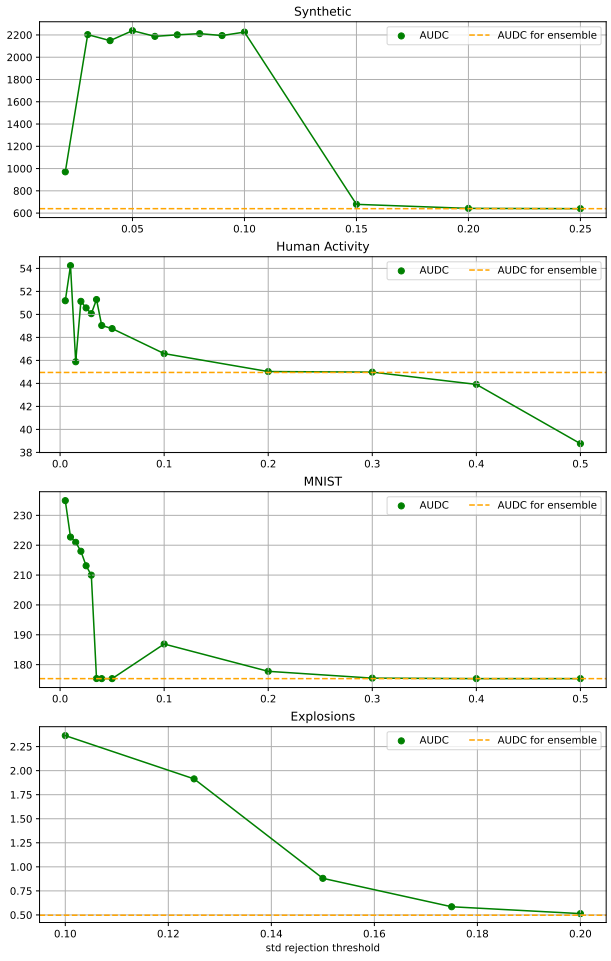

Figure 5. Dynamics of AUDC metric for our CPD-with-rejection approach evaluated for the all 4 datasets.

Acknowledgements

We are grateful to Evgenia Romanenkova for developing ideas of the project and the general supervision of our team.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.05.008. URL https://www.sciencedirect.com/science/article/pii/S1566253521001081.

Aminikhanghahi, S. and Cook, D. J. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

Bartlett, P. L. and Wegkamp, M. H. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(59):1823–1840, 2008. URL http://jmlr.org/papers/v9/bartlett08a.html.

Chang, W.-C., Li, C.-L., Yang, Y., and Póczos, B. Kernel change-point detection with auxiliary deep generative models. In *International Conference on Learning Representations*, Vancouver, Canada, 2018. ICLR.

Chung, Y., Char, I., Guo, H., Schneider, J. G., and Neiswanger, W. Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification. *ArXiv*, abs/2109.10254, 2021.

Deldari, S., Smith, D. V., Xue, H., and Salim, F. D. Time series change point detection with self-supervised contrastive predictive coding. In *Proceedings of The Web Conference 2021*, WWW '21. Association for Computing Machinery, 2021. doi: 10.1145/3442381.

3449903. URL https://doi.org/10.1145/3442381.3449903.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019.

Ganguly, A. and Earp, S. W. F. An introduction to variational inference. *ArXiv*, abs/2108.13083, 2021.

Harchaoui, Z., Moulines, E., and Bach, F. R. Kernel change-point analysis. In *Advances in neural information processing systems*, pp. 609–616, Vancouver, Canada, 2009. NeurIPS.

Herbei, R. and Wegkamp, M. H. Classification with reject option. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 34(4):709–721, 2006. ISSN 03195724. URL http://www.jstor.org/stable/20445230.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hushchyn, M., Arzymatov, K., and Derkach, D. Online neural networks for change-point detection. *arXiv preprint arXiv:2010.01388*, 2020.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf.

Lavielle, M. and Teyssière, G. *Adaptive Detection of Multiple Change-Points in Asset Price Volatility*, pp. 129–156. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-34625-8. doi: 10.1007/978-3-540-34625-8_5. URL https://doi.org/10.1007/978-3-540-34625-8_5.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Li, S., Xie, Y., Dai, H., and Song, L. M-statistic for kernel change-point detection. *Advances in Neural Information Processing Systems*, 28, 2015.

Liu, J., Paisley, J., Kioumourtzoglou, M.-A., and Coull, B. Accurate uncertainty estimation and decomposition in ensemble learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1cc8a8ea51cd0adddf5dab504a285915-Paper.pdf.

Malladi, R., Kalamangalam, G. P., and Aazhang, B. Online bayesian change point detection algorithms for segmentation of epileptic activity. In *2013 Asilomar Conference on Signals, Systems and Computers*, pp. 1833–1837, 2013. doi: 10.1109/ACSSC.2013.6810619.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Page, E. S. Continuous inspetion schemes. *Biometrika*, 41(1-2):100–115, 06 1954. ISSN 0006-3444. doi: 10.1093/biomet/41.1-2.100. URL https://doi.org/10.1093/biomet/41.1-2.100.

Romanenkova, E., Zaytsev, A., Klyuchnikov, N., Gruzdev, A., Antipova, K., Ismailova, L., Burnaev, E., Semenikhin, A., Koryabkin, V., Simon, I., and Koroteev, D. Real-time data-driven detection of the rock-type alteration during a directional drilling. *IEEE Geoscience and Remote Sensing Letters*, 17(11):1861–1865, 2020. doi: 10.1109/LGRS.2019.2959845.

Romanenkova, E., Stepikin, A., Morozov, M., and Zaytsev, A. Indid: Instant disorder detection via a principled neural network. MM '22, pp. 3152–3162, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548182. URL https://doi.org/10.1145/3503161.3548182.

Shewhart, W. A. *Economic control of quality of manufactured product*. Macmillan And Co Ltd, London, 1931.

Shiryaev, A. *Stochastic Change-Point Detection Problems*. MCCME (in Russian), Moscow, 2017.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28: 3483–3491, 2015.

Spokoiny, V. Multiscale local change point detection with applications to value-at-risk. *The Annals of Statistics*, 37 (3):1405–1436, 2009. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/30243672.

Sultani, W., Chen, C., and Shah, M. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6479–6488, Salt Lake City, Utah, USA, 2018. IEEE.

Tartakovsky, A. G. and Moustakides, G. V. State-of-the-art in bayesian changepoint detection. *Sequential Analysis*, 29(2):125–145, 2010.

Truong, C., Oudre, L., and Vayatis, N. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.

van den Burg, G. J. and Williams, C. K. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*, 2020.

Yang, P., Dumont, G., and Ansermino, M. Adaptive change detection in heart rate trend monitoring in anesthetized children. *IEEE transactions on bio-medical engineering*, 53:2211–9, 12 2006. doi: 10.1109/TBME.2006.877107.

Zhang, M. and Sawchuk, A. A. USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 1036–1043, Pittsburgh, Pennsylvania, 2012. ACM.

A. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Alexander Stepikin (40% of work)

- Coding and training ensembles of models
- Experimenting CPD-with-rejection methods
- Preparing the final presentation
- Preparing the GitHub Repository
- Preparing the Report generally and especially Sections 2, 1 of this report

Maria Kovaleva (30% of work)

- Preparing code for CUSUM methods
- Experimenting with the CUSUM methods
- Preparing the final presentation
- Preparing the Report generally and especially Section 3.3 and 4.3.2

Aleksandr Kukharskii (30% of work)

- Preparing code for quantile methods
- Experimenting with the quantile methods
- Preparing the final presentation
- Preparing the Report generally and especially Section 4.3.1 and 3.3.

B. Reproducibility checklist

Answer the questions of following reproducibility checklist. If necessary, you may leave a comment.

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: We have used the code from the original paper (Romanenkova et al., 2022) (GitHub Repo: InDiD). As a baseline, we used seq2seq BCE model implemented there. Our code modifications allow training ensembles of such models, CUSUM-aggregation of CP scores and using CPD-with-rejection approach. Moreover, our code is built into the general pipeline and can be used with any of the other models presented in the original repository.

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

4. A complete description of the data collection process, including sample size, is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

Students' comment: None

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

9. The exact number of evaluation runs is included.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

10. A description of how experiments have been conducted is included.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

12. Clearly defined error bars are included in the report.

    ☐ Yes.
    ☐ No.
    ☑ Not applicable.

    Students' comment: None

13. A description of the computing infrastructure used is included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

C. Extended experimental results

In this Appendix, we provide the results of the additional experiments conducted. In particular, we also tried to train base models of an ensemble on smaller sub-samples of the initial train set different from each other. To generate these subsamples, the standard bootstrap technique was used. Table 2 presents the quality metrics we obtained for two datasets: sequences of MNIST images and Human Activity Recognition. Different sizes of train subsets were used. The predictions are obtained by standard point-wise averaging. None of the other proposed techniques was used here.

Based on Table 2, the following conclusions can be made.

1. The main quality metrics for the ensemble predictions monotonously worsen as the size of the train subsets decreases. The only exception here is Mean Time to a False Alarm. However, this is a sign of a large number of false alarms raised by the model.

2. When using smaller subsets of the initial train set for the base models' fitting, the training time decreases as well, which can be beneficial in case of limited resources. Moreover, the metrics for the relative train subset size of 0.3 are almost equal to the ones for the original ensemble.

3. The metrics could be improved if more base models were trained on these small subsets. Thus, there is a trade-off between time resources and detection quality.

Table 2. Main quality metrics for the ensembles of baseline seq2seq BCE models trained on different sub-samples of the initial train set obtained by bootstrap. The predictions are obtained by standard point-wise averaging; none of the other proposed techniques was used. Results are given for train subsets of different sizes. ↑ marks metrics that should be maximized, ↓ – minimized. Best metrics values are highlighted with **bold** font.

Relative train subset size	AUDC ↓	Mean Time to FA ↑	Mean DD ↓	Max F1 ↑	Covering ↑
Sequences of MNIST images					
1.00 (full)	**639.73**	95.01	**1.21**	**0.9839**	**0.9880**
0.70	641.42	94.16	1.19	0.9808	0.9836
0.30	664.72	95.52	2.03	0.9773	0.9806
0.15	748.58	**96.96**	3.70	0.9467	0.9683
Human Activity Recognition					
1.00 (full)	**44.95**	11.07	**0.10**	**0.9927**	**0.9938**
0.70	46.06	11.07	0.11	0.9918	0.9927
0.30	51.64	11.09	0.21	0.9887	0.9856
0.15	59.54	**11.25**	0.28	0.8467	0.9826

D. Implementation details

This Appendix provides all the implementation details for our experiments: models' architectures, training parameters, data pre-processing procedures, experimental setup, hyper-parameters selection, and the evaluation pipeline description.

D.1. Models' architecture

Following (Romanenkova et al., 2022), we use sequence-to-sequence BCE base models of the same architecture. For all the datasets, core models are LSTM (Hochreiter & Schmidhuber, 1997) networks followed by dense linear layer(-s). More details are provided below.

Synthetic 1D. For this dataset, we use the LSTM block with an input size equal to 1, a hidden size equal to 4, and a dropout probability of 0.5 followed by a Linear(4, 1) layer and the Sigmoid activation.

Sequences of MNIST images For this dataset, we use the LSTM block with an input size equal to 784, a hidden size equal to 32, and a dropout probability of 0.25 followed by a Linear(32, 1) layer and the Sigmoid activation.

Human Activity Recognition For this dataset, we use the LSTM block with the an a input size equal to 28, hidden size equal to 8, and a dropout probability of 0.5 followed by a Linear(8, 1) layer and the Sigmoid activation.

Explosion For this dataset, we first use a pre-trained 3D Convolutional network (Feichtenhofer et al., 2019) that embeds each input frame of original size $240 \times 320 \times 3$ to a feature space with a dimension equal to 12288. Then, we process the obtained representations with the LSTM block with an input size equal to 12288, a hidden size equal to 64, and a dropout probability of 0.5 followed by a Linear(64, 1) layer and the Sigmoid activation.

D.2. Training parameters

All the datasets were split into train and test parts in the ratio of $7:3$. For all the experiments, the base models were trained for a maximum number of 100 epochs with an early stopping criterion monitored by the validation loss. The minimal delta parameter was set to 0 and the patience – to 10 epochs. The models were optimized via the Adam (Kingma & Ba, 2014) optimizer with a learning rate equal to 0.001. For all the experiments, except for the video data, we used a batch size of 64, while for the Explosion dataset, a batch size was equal to 16.

D.3. Evaluation pipeline

For the basic experiments with ensembles of CP detectors, the evaluation pipeline was the following. As mentioned before, the model's predictions depend on the choice of the alarm threshold s. We used 100 different thresholds in a range from 0 to 1 to obtain different trade-offs between Mean Detection Delay and Mean Time to a False Alarm as well as the F1-score and the Covering metric. We choose the threshold that maximizes the F1-score and report the metrics corresponding to this threshold. In addition, we compute the AUDC metric for the obtained detection curves consisting of 100 points and report it in the tables. This metric evaluates the overall performance of a model.

For the CPD-with-rejection approach, there is an additional threshold parameter r. In this case, we repeat the steps described above for a set of different values of r and choose the value that maximizes F1-score as well.

For the experiments with CUSUM aggregation of CP scores, we compute metrics for a set of SUCUM thresholds and choose the one that maximizes F1-score as well. As this threshold has different ranges of possible values for different datasets, we do not report the AUDC metric here.

D.4. Selection of hyper-parameters

Quantile-based ensemble predictions To obtain predictions of the ensembles, we used both simple point-wise averaging and quantiles of different probabilities: 0.3, 0.5, 0.7 and 0.9. For Synthetic 1D data, the best quantile was 0.7; for Human Activity Recognition and Explosion – 0.5; for sequences of MNIST images – 0.3.

CUSUM alarm threshold For Synthetic 1D data, we searched for an optimal CUSUM alarm threshold in a range from 1.0 to 5.0; the optimal value is equal to 2.0. For Human Activity Recognition, we searched for an optimal CUSUM alarm threshold in a range from 1.0 to 6.0; the optimal value is equal to 5.4. For sequences of MNIST images, we searched for an optimal CUSUM alarm threshold in a range from 1.0 to 15.0; the optimal values is equal to 11.5. For Explosion, we searched for an optimal CUSUM alarm threshold in a range from 0.1 to 2.7; the optimal value is equal to 1.9.

Std rejection threshold For all the datasets, we searched through different values of std rejection thresholds ranging from the minimal std of the predictions in the whole test set to the maximal one. For Synthetic 1D data, this resulted in the optimal threshold value equal to 0.20; for Human Activity Recondition – 0.50; for sequences of MNIST images – 0.03; for Explosion – 0.15.

D.5. Experimental setup and computing infrastructure

The source code was written in Python 3. We used PyTorch deep learning framework and its PyTorch Lightning wrapper. The models were mostly trained on the GPU.