Change point detection with neural online density-ratio estimator

Xiuheng Wang, Ricardo Augusto Borsoi, Cédric Richard, Jie Chen

CHANGE POINT DETECTION WITH NEURAL ONLINE DENSITY-RATIO ESTIMATOR

Xiuheng Wang [*], *Ricardo Augusto Borsoi* [†], *Cédric Richard* [*], *Jie Chen* [‡]

[*] Université Côte d'Azur, CNRS, OCA, Nice, France
[†] Université de Lorraine, CNRS, CRAN, Vandoeuvre-lès-Nancy, France
[‡] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
xiuheng.wang@oca.eu, ricardo.borsoi@univ-lorraine.fr, cedric.richard@unice.fr, dr.jie.chen@ieee.org

ABSTRACT

Detecting change points in streaming time series data is a long standing problem in signal processing. A plethora of methods have been proposed to address it, depending on the hypotheses at hand. Non-parametric approaches are particularly interesting as they do not make any assumption on the distribution of data or on the nature of changes. Nevertheless, leveraging recent advances in deep learning to detect change points in time series data is still challenging. In this paper, we propose a change point detection method using an online approach based on neural networks to directly estimate the density-ratio between current and reference windows of the data stream. A variational continual learning framework is employed to train the neural network in an online manner while retaining information learned from past data. This leads to a statistically-principled fully nonparametric framework to detect change points from streaming data. Experimental results with synthetic and real data illustrate the effectiveness of the proposed approach.

Index Terms— Change point detection, online, density-ratio estimation, neural networks, continual learning.

1. INTRODUCTION

Change point detection (CPD) consists of detecting abrupt changes in the statistical properties of time series measurements [1]. As a fundamental problem in statistics and signal processing, CPD has seen major interest from the community in the past decades, and has been applied to fields as diverse as medical condition monitoring [2], speech recognition [3] and image analysis [4].

Numerous approaches have been proposed to perform CPD. Depending on whether prior information on data distributions is available, recent CPD approaches can be roughly divided into parametric and non-parametric strategies. Parametric ones rely on model assumptions describing the probability density function (PDF) of the data before and after an abrupt change. Examples of parametric CPD strategies include the cumulative sum (CUSUM) [5], the generalized likelihood ratio test (GLRT) [6], and subspace identification (SI) [7]. The CUSUM algorithm [5] assumes that the parameters undergoing changes are known and requires knowledge of the change in either the mean or the variance. The GLRT method [6] assumes that observations are driven by a linear state-space model. By explicitly considering a noise factor in a linear state-space model, the SI approach [7] detects changes using distances between the subspaces spanned by two sequence windows.

Parametric CPD methods operate well when all the assumptions on the problem at hand are met. Nevertheless, deriving a model that accurately describes the data is usually intractable and makes parametric approaches sensitive to modeling errors [8]. Non-parametric CPD methods have been introduced to address this issue. These approaches make weaker assumptions about the data and include, for instance, the use of empirical estimation of the cumulative data distribution, or the deviation of a kernel embedding of the data from its mean [8]. A non-parametric strategy of particular interest is the use of density-ratio estimation. Although the distribution of the pre- and post-change data can be hard to estimate, only their ratio – which can be easier to estimate – is necessary to perform CPD [9]. Several CPD methods based on density-ratio estimation have been proposed in the literature. Examples include the Kullback-Leibler (KL) divergence based importance estimation procedure (KLIEP) [10], the unconstrained least squares importance fitting (uLSIF) and the relative uLSIF [11].

Unlike these offline methods that detect changes in a dataset collected a priori, online CPD algorithms process streaming data iteratively in an adaptive fashion. The method in [12] considers using the k-nearest neighbors (kNN) algorithm to tackle online CPD by extending SI techniques in non-linear subspaces. In [13], the moving average-based algorithm NEWMA is introduced to monitor the mean of the process in a feature space. An online version of the relative uLSIF-based method NOUGAT is designed in [14] to detect change points by learning density-ratios with the kernel trick. Another important branch of non-parametric online CPD is based on virtual classifiers (VC) [15, 16]. These methods train a binary classifier with pseudo labels to learn density-ratio over past and future data, and consider the separability of data to detect change points. An online Bayesian approach using a latent class model for the data whose number of classes can increase over time was proposed in [17]. However, Bayesian methods can have high complexity when compared to approaches such as [13, 14].

Recently, deep learning has become a popular framework for addressing a variety signal processing tasks. Several works considered deep learning for CPD. In [18], an autoencoder is used to learn a time invariant representation of the data which is more amenable for CPD. Neural networks are used for density-ratio estimation in [19]. However, both approaches do not operate online. Another method based on the reconstruction error of an autoencoder is proposed in [20] with real-time preprocessing. However, it relies on strong assumptions about the nature of the changes. A related approach using an autoencoder based on recurrent neural networks was proposed in [21]. Current deep learning and density-ratio learning CPD algorithms are still limited in combining flexibility with the ability of retaining knowledge from past data, while maintaining a low-complexity. Continual learning [22, 23] has the ability to adapt to recent data while at the same time retaining past knowledge. This made it successful in various online learning tasks.

In this paper, a new online CPD strategy based on neural density-

The work of C. Richard was funded in part by the ANR under grant ANR-19-CE48-0002, and by the 3IA Côte d'Azur Senior Chair program. The work of J. Chen was supported in part by NSFC grant 62192713.

ratio estimation and continual learning is proposed. First, density-ratio estimation is represented as a binary classification problem over two sliding (reference and test) data windows. This allows us to leverage state-of-the-art probabilistic classification neural networks to perform CPD in a non-parametric manner. Moreover, to obtain an adaptive detection strategy that leverages past information while operating online, a variational continual learning objective is devised to train the neural network classifier in a Bayesian framework. Specifically, the statistical distribution of the network parameters at each time step is used as a prior for the next classification objective in a regularization-based framework. This allows the trade-off between temporal smoothness and fast adaptation to be controlled using a single regularization parameter. Experimental results with both synthetic and real data show the effectiveness of the proposed strategy.

2. PROBLEM FORMULATION

Let us consider a time series of d-dimensional vector-valued data $\{\boldsymbol{x}_t\}_{t\in\mathbb{N}}$, with $\boldsymbol{x}_t \in \mathbb{R}^d$. We assume that there exists a time index $t_r \in \mathbb{N}$ with an abrupt change in the statistical distribution of $\boldsymbol{x}_t$, that is:

$$t < t_r : \ \boldsymbol{x}_t \sim p(\boldsymbol{x}) \,, \qquad t \geq t_r : \ \boldsymbol{x}_t \sim q(\boldsymbol{x}) \,, \tag{1}$$

where $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$, which are assumed to be different, denote the PDFs of the data before and after t_r. The latter is the so-called *change point*. To make the presentation clearer, without loss of generality, note that we consider in (1) the case of a single change point.

The CPD problem consists of estimating the change point $\hat{t}_r$ that is as close as possible to the true change point t_r. In this work, we consider a more general version of this problem, in which $\{\boldsymbol{x}_t\}$ might contain multiple change points, and $\boldsymbol{x}_t$ is a streaming signal that is observed sequentially over time. We address the requirement that CPs must be detected *online*, i.e., we need to decide whether each time instant $t \in \mathbb{N}$ is a change point based only on past data $\{\boldsymbol{x}_{t'}\}_{t'\leq t}$. This leads to two objectives when designing an online CPD algorithm: minimizing the probability of a false alarm (of flagging $t \neq t_r$ as a change point), and minimizing the detection delay, i.e., $\hat{t}_r - t_r$ for $\hat{t}_r$ being the first detection after t_r. Note that our method is built upon multi-dimensional streaming signals, but it can also address the case of one-dimensional time-series where $\boldsymbol{x}_t$ is a scalar (i.e., $d = 1$). Moreover, we focus on non-parametric strategies, in which no parametric form is assumed for the PDFs $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$.

3. THE PROPOSED METHOD

The basic idea of the proposed CPD strategy consists of estimating change points by means of evaluating the density-ratio between the PDFs of the data over a reference and a test window, given by:

$$r(\boldsymbol{x}_t) = \frac{p_{\text{test}}(\boldsymbol{x}_t)}{p_{\text{ref}}(\boldsymbol{x}_t)} \,, \tag{2}$$

with $p_{\text{test}}(\boldsymbol{x})$ the data PDF over the test window with N samples:

$$\mathcal{X}_t = \big\{\boldsymbol{x}_{t-N+1}, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t\big\} \,, \tag{3}$$

and $p_{\text{ref}}(\boldsymbol{x})$ the data PDF over the reference window with N' samples:

$$\mathcal{X}'_t = \big\{\boldsymbol{x}_{t-N-N'+1}, \ldots, \boldsymbol{x}_{t-N-1}, \boldsymbol{x}_{t-N}\big\} \,. \tag{4}$$

Our objective is to estimate the density-ratio $r(\boldsymbol{x}_t)$, at each time $t \in \mathbb{N}$, given only the data $\boldsymbol{x}_t$ observed sequentially over windows $\mathcal{X}_t$ and $\mathcal{X}'_t$. To this end, we will consider two steps: first, a probabilistic classification-based approach is introduced to estimate the density-ratio; afterwards, we propose to use a Bayesian continual learning strategy in order to learn the classifier online.

3.1. Neural Online Density-ratio Estimator

Without additional knowledge about $p_{\text{test}}(\boldsymbol{x}_t)$ and $p_{\text{ref}}(\boldsymbol{x}_t)$, computing these PDFs can be intractable, and the non-parametric estimation of $r(\boldsymbol{x}_t)$ becomes more desirable. Within this context, kernel [14] or deep learning [19] strategies have been proposed to estimate density-ratio with the design of specific learning objectives. An important property of the density-ratio is that it can be related to probabilistic binary classification, allowing us to leverage state-of-the-art classification methods to address this problem [24]. First, let us annotate the samples in data sets $\mathcal{X}'_t$ and $\mathcal{X}_t$ with pseudo labels 0 and 1, respectively. This way, considering the labels to be a random variable $y_t \in \{0, 1\}$, we can express the distributions $p_{\text{test}}(\boldsymbol{x}_t)$ and $p_{\text{ref}}(\boldsymbol{x}_t)$ in the form of a single conditional PDF:

$$p_{\text{test}}(\boldsymbol{x}_t) = p(\boldsymbol{x}_t|y_t = 1) \,, \tag{5}$$

$$p_{\text{ref}}(\boldsymbol{x}_t) = p(\boldsymbol{x}_t|y_t = 0) \,. \tag{6}$$

Using Bayes' rule and the above definition and assuming the two classes with equal a priori marginal class probabilities, equation (2) can be written as:

$$r(\boldsymbol{x}_t) = \frac{p(\boldsymbol{x}_t|y_t = 1)}{p(\boldsymbol{x}_t|y_t = 0)} = \frac{p(y_t = 1|\boldsymbol{x}_t)}{1 - p(y_t = 1|\boldsymbol{x}_t)} \,. \tag{7}$$

In this way, the density-ratio between $p_{\text{test}}(\boldsymbol{x}_t)$ and $p_{\text{ref}}(\boldsymbol{x}_t)$ can be recovered by the optimal binary classifier $p(y_t|\boldsymbol{x}_t)$ that distinguishes between samples from these two distributions.

By concatenating the two data sets corresponding to the reference and test windows as:

$$\mathcal{D}_t = \big\{(\boldsymbol{x}_t, y_t = 0) : \boldsymbol{x}_t \in \mathcal{X}'_t\big\} \bigcup \big\{(\boldsymbol{x}_t, y_t = 1) : \boldsymbol{x}_t \in \mathcal{X}_t\big\} \,,$$

which leads the CPD problem to be formulated as learning a binary classifier at each time $t \in \mathbb{N}$ based on the training data $\mathcal{D}_t$, also called as virtual classifier [15,16]. We denote this learnable classifier by $p(y_t|\boldsymbol{x}_t, \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ denotes a vector containing its parameters. It has been shown in [24] that a wide range of losses used in binary classification are suitable to perform density-ratio estimation.

It is popular to parameterize this classifier as $p(y_t = 1|\boldsymbol{x}_t) = \sigma(f_{\boldsymbol{\phi}}(\boldsymbol{x}_t))$, where $\sigma(\cdot)$ is the logistic sigmoid function given by $\sigma(x) = e^x/(e^x + 1)$ and $f_{\boldsymbol{\phi}} : \mathbb{R}^d \mapsto \mathbb{R}$ is a neural network parameterized by $\boldsymbol{\phi}$. When the classifier is trained using maximum likelihood estimation, the optimal value for $f_{\boldsymbol{\phi}}$ is $\log r(\boldsymbol{x}_t)$ [25]. This yields the proposed neural online density-ratio estimator (NODE):

$$r_\phi(\boldsymbol{x}_t) = \exp\big(f_{\boldsymbol{\phi}}(\boldsymbol{x}_t)\big) \,, \tag{8}$$

where the subscript ϕ emphasizes that $r_\phi(\boldsymbol{x}_t)$ depends on the learned classifier. CPD is then performed by comparing the test statistic $|\frac{1}{N}\sum_{\boldsymbol{x}\in\mathcal{X}_t}(r_\phi(\boldsymbol{x}) - 1)|$ to a given threshold $\xi \in \mathbb{R}_+$. The average over $\mathcal{X}_t$ is used to obtain more stable detections.

A crucial consideration for the proper estimation of $p(y_t|\boldsymbol{x}_t, \boldsymbol{\phi})$ is that this classifier should avoid overfitting given the limited data set $\mathcal{D}_t$ with $N + N'$ samples. This is particularly important since the window lengths directly impact the performance of the algorithm: they need to be small to limit the detection delay, but large in order to supply enough training data. This issue will be alleviated in the following by considering an online continual learning strategy for NODE, in which information from previous windows is leveraged when learning the current classifier.

3.2. Continual learning strategy

As discussed above, we train a neural classifier and update its parameters $\boldsymbol{\phi}$ using samples in $\mathcal{D}_t$ at each time instant t. Since the overlapping part in training datasets at neighboring time instants, e.g., $\mathcal{D}_{t-1}$ and $\mathcal{D}_t$, is relatively large, it is beneficial to retain the knowledge acquired from $\mathcal{D}_{1:t-1}$ when training the classifier on $\mathcal{D}_t$. This is particularly important to benefit from past information and avoid overfitting when the window length is small. To iteratively learn the classifier while retaining the knowledge acquired from past iterations, we investigate a variational continual learning (VCL) strategy [23] in our CPD algorithm.

Given an independent input $\boldsymbol{x}$, let us consider that the classifier returns a probability distribution $p(y|\boldsymbol{x},\boldsymbol{\phi})$ of its label y, given its parameters $\boldsymbol{\phi}$. Note that the classifier parameters are assumed to be random variables as this allows one to account for their uncertainty, which can be important when training with small amounts of data. In the continual learning setting, we aim to compute the distribution of the parameters at time t, denoted $\boldsymbol{\phi}_t$, given the data set $\mathcal{D}_t$. This is computed using Bayes' rule:

$$p(\boldsymbol{\phi}_t|\mathcal{D}_t) \propto p(\mathcal{D}_t|\boldsymbol{\phi}_t)p(\boldsymbol{\phi}_t)\,, \tag{9}$$

where $p(\boldsymbol{\phi}_t)$ is a properly selected prior for the parameters which captures the information from the past data. To compute $p(\boldsymbol{\phi}_t|\mathcal{D}_t)$ recursively, as in a Bayesian filtering framework, the prior $p(\boldsymbol{\phi}_t)$ is selected as the posterior distribution of the parameters computed at the previous iteration, $p(\boldsymbol{\phi}_{t-1}|\mathcal{D}_{t-1})$. However, the posterior distribution is intractable in general and needs to be approximated. VCL [23] approximates the posterior distribution by another distribution q belonging to a tractable family $\mathcal{Q}$. This is performed by finding the distribution $q \in \mathcal{Q}$ which minimizes the KL divergence to the true posterior:

$$q_t(\boldsymbol{\phi}_t) = \arg\min_{q\in\mathcal{Q}}\ \mathrm{KL}\Big(q(\boldsymbol{\phi})\Big\|\frac{1}{Z_t}p(\mathcal{D}_t|\boldsymbol{\phi})q_{t-1}(\boldsymbol{\phi})\Big)\,, \tag{10}$$

where $q_{t-1}(\boldsymbol{\phi})$ is the approximate parameters posterior that was computed at time $t-1$, and Z_t is a normalizing constant (which will not be required in the optimization process). The zeroth approximated posterior $q_0(\boldsymbol{\phi})$ is defined as the prior distribution of the parameters $p(\boldsymbol{\phi})$. Training the classifier using the variational inference in (10) is equivalent to maximizing the evidence lower bound to the data log-likelihood $\log p(\mathcal{D}_t)$, which leads to the following cost function:

$$\begin{aligned}\mathcal{L}_t(q_t(\boldsymbol{\phi})) = &\sum_{n=0}^{N+N'-1} \mathbb{E}_{\boldsymbol{\phi}\sim q_t(\boldsymbol{\phi})}\big\{\log p(y_{t-n}|\boldsymbol{\phi},\boldsymbol{x}_{t-n})\big\} \\ &-\lambda\,\mathrm{KL}\big(q_t(\boldsymbol{\phi})||q_{t-1}(\boldsymbol{\phi})\big)\,.\end{aligned} \tag{11}$$

Here we introduce a hyperparameter λ to trade-off between stability of the continual learning strategy, and its ability to adapt in the presence of a change point.

We consider the variational family $\mathcal{Q}$ as Gaussian distributions with diagonal covariance matrices (i.e., a mean field assumption), which make the learning process more efficient since the KL divergence in (11) can be computed in closed form. $p(y|\boldsymbol{\phi},\boldsymbol{x})$ is modeled as a Bernoulli distribution. The optimization of $\mathcal{L}_t(q_t(\boldsymbol{\phi}))$ is performed by using the Adam [26] gradient-based optimizer, where the *reparametrization trick* [27] was used to tackle the expectation with respect to $q_t(\boldsymbol{\phi})$. At each time instant, (11) is maximized for M epochs, with the parameters of the distribution $q_t(\boldsymbol{\phi})$ initialized with those of $q_{t-1}(\boldsymbol{\phi})$, obtained as the solution at $t-1$ (i.e., *warm start*). The proposed CPD procedure is summarized in Algorithm 1.

Algorithm 1: CPD with NODE

```
Input: $\{\boldsymbol{x}_t\}$, parameter $\lambda$, number of epochs $M$, threshold $\xi$.
Initialization: optimize (11) with $\lambda = 0$ for $t = 1$ ;
for $t = 2, 3, \ldots$ do
    Update data windows to build the data set $\mathcal{D}_t$ ;
    Optimize cost function (11) for $M$ epochs ;
    Compute the density-ratio using (8) ;
    if $|\frac{1}{N}\sum_{\boldsymbol{x}\in\mathcal{X}_t}(r_{\boldsymbol{\phi}}(\boldsymbol{x})-1)| > \xi$ then
        Flag $t$ as a change point;
    end
end
```

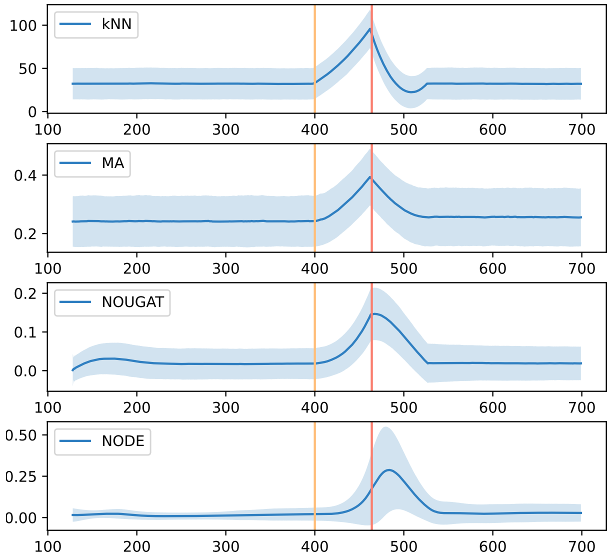

Fig. 1. Mean of the test statistic (± standard deviation) for all compared algorithms. The change point t_r is located at the yellow line and $t_r + N$ at the red line.

4. EXPERIMENTS

In this section, we validate the proposed online CPD method with NODE and compare it with three baselines, namely, the kNN [12], MA [13, 14] and NOUGAT [14]. For all experiments, $f_{\boldsymbol{\phi}}$ was a fully connected network with three hidden layers, where each layer contained 16 units (32 units for real data) with Tanh activations. The reference and test window lengths were both set to $N = N' = 64$ for all algorithms. The network was trained for 20 epochs during initialization, then for $M = 1$ epoch for $t > 1$. We set $\lambda = 20$ for simulated data and $\lambda = 5$ for real data. The codes are made available at `www.github.com/xiuheng-wang/NODE_release`.

4.1. Monte Carlo validation

The simulated signals $\boldsymbol{x}_t$ were sampled from mixtures of k d-dimensional Gaussian distributions $\mathcal{N}_d(\boldsymbol{m}_q, q^{-1}\boldsymbol{C}_q)$ with $q = 1,\ldots,d$. The weights α_q of the mixture model were generated from a flat Dirichlet distribution with concentration coefficient β. The means $\boldsymbol{m}_q$ and the covariance matrix $\boldsymbol{C}_q$ were sampled from $\mathcal{N}_d(\boldsymbol{0},\boldsymbol{I})$ and a Wishart distribution with the scaling matrix $\boldsymbol{I}$ and $d+2$ degrees of freedom. We generated 700 samples and put a change point at $t_r = 400$. We set $d = 6, k = 3, \beta = 5$, and all parameters $\{\boldsymbol{m}_q, \alpha_q, \boldsymbol{C}_q\}$ were resampled at time $t = t_r$.

Fig. 1 shows the mean ± standard deviation of the test statistic

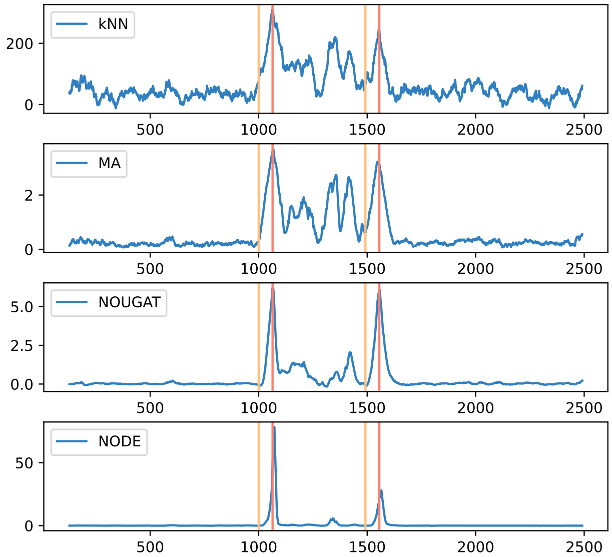

Fig. 2. Credit card fraud detection. The change point t_r is located at the yellow line and $t_r + N$ at the red line.

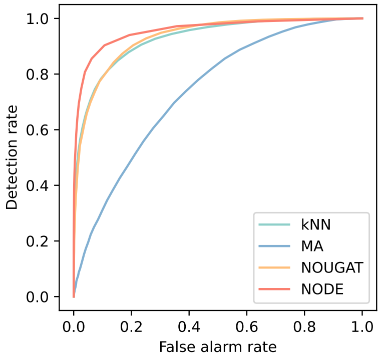

Fig. 3. ROC curves for all compared algorithms. The closer a ROC curve is to the upper left corner, the better the algorithm performs.

of all compared algorithms for 10^4 Monte Carlo runs. Comparing the ratio between the test statistic at the peak at $t_r + N$ and before t_r, NODE achieved the best performance compared to the other methods. This can be seen more clearly in the Receiver Operating Characteristic (ROC) curves computed based on the multiple Monte Carlo runs and shown in Fig. 3, where NODE achieves an improvement of detection rate for false alarm rates from 0 to 0.4.

4.2. Credit card fraud detection

The real data in the credit card fraud detection data set is composed of transactions made in September 2013 by European cardholders[1]. The raw data were preprocessed by applying PCA, and the first five components ($d = 5$) were considered to obtain streaming signals $\boldsymbol{x}_t$. This data set contains 492 frauds out of 284,807 transactions. We inserted the 492 frauds after the first 1000 genuine transactions to create two change points at $t_r = 1000$ and $t_r = 1492$.

Fig. 2 illustrates the detection statistics of kNN, MA, NOUGAT and NODE. The test statistic values for all algorithms were significantly larger in the vicinity of $t_r + N$ compared to intervals not in the vicinity of the change points. However, the test statistics of kNN and MA showed large values between the two change points. NOUGAT performed better than MA and kNN, however, NODE obtained the best performance, with test statistic values that were only non-negligible after the change points, which translates into a very low false alarm rate.

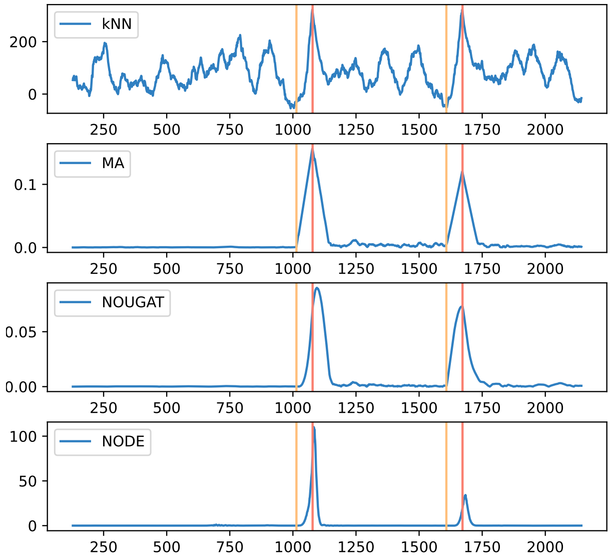

Fig. 4. Text language detection. The change point t_r is located at the yellow line and $t_r + N$ at the red line.

4.3. Text language detection

The real data set for text language detection was created from a data set containing text from 17 different languages[2]. Raw texts were first cleaned by removing symbols and numbers and then represented via a linear embedding of dimensionality $d = 20$ using `word2vec`. Time series $\boldsymbol{x}_t$ was formed by concatenating the representations of 1014 French, 594 Malayalam, and 526 Arabic texts.

The results are provided in Fig 4. The test statistic of kNN produced very large values in the absence of change points when compared to the other algorithms, what led to a large false alarm rate. MA, NOUGAT and NODE provided comparable results. However, NODE's test statistic was more stable outside of the vicinity of change points.

The execution times of NODE were about an order of magnitude larger than the other methods in all experiments. However, NODE was implemented in a different computation platform (Python) than the baselines (Julia), which reduces the appropriateness of comparing their execution times. A more in-depth study of the complexity of the proposed method and the development of more efficient solutions will be the subject of future work.

5. CONCLUSION

In this paper, we introduced a novel strategy for online CPD that leverages the powerful learning ability of neural networks to estimate density-ratio in a non-parametric manner. A continual learning framework was exploited to devise an adaptive detection algorithm that retains past information. Experiments illustrated the superiority of the proposed strategy compared to state-of-the-art methods.

[1] `www.kaggle.com/datasets/mlg-ulb/creditcardfraud`

[2] `www.kaggle.com/datasets/basilb2s/language-detection`

6. REFERENCES

[1] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and information systems*, vol. 51, no. 2, pp. 339–367, 2017.

[2] D. Gajic, Z. Djurovic, J. Gligorijevic, S. Di Gennaro, and I. Savic-Gajic, "Detection of epileptiform activity in eeg signals based on time-frequency and non-linear analysis," *Frontiers in computational neuroscience*, vol. 9, pp. 38, 2015.

[3] D. Rybach, C. Gollan, R. Schluter, and H. Ney, "Audio segmentation for speech recognition using segment features," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 4197–4200.

[4] R. A. Borsoi, C. Richard, A. Ferrari, J. Chen, and J. C. M. Bermudez, "Online graph-based change point detection in multiband image sequences," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 850–854.

[5] C. Inclan and G. C. Tiao, "Use of cumulative sums of squares for retrospective detection of changes of variance," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 913–923, 1994.

[6] F. Gustafsson, "The marginalized likelihood ratio test for detecting abrupt changes," *IEEE Transactions on automatic control*, vol. 41, no. 1, pp. 66–78, 1996.

[7] Y. Kawahara, T. Yairi, and K. Machida, "Change-point detection in time-series data based on subspace identification," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE, 2007, pp. 559–564.

[8] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, pp. 107299, 2020.

[9] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density ratio estimation in machine learning*, Cambridge University Press, 2012.

[10] M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," *Advances in neural information processing systems*, vol. 20, 2007.

[11] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks*, vol. 43, pp. 72 – 83, 2013.

[12] H. Chen, "Sequential change-point detection based on nearest neighbors," *The Annals of Statistics*, vol. 47, no. 3, pp. 1381–1407, 2019.

[13] N. Keriven, D. Garreau, and I. Poli, "Newma: a new method for scalable model-free online change-point detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3515–3528, 2020.

[14] A. Ferrari, C. Richard, A. Bourrier, and I. Bouchikhi, "Online change-point detection with kernels," *Pattern Recognition*, p. 109022, 2022.

[15] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2961–2974, 2005.

[16] M. Yamada, A. Kimura, F. Naya, and H. Sawada, "Change-point detection with feature selection in high-dimensional time-series data," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[17] P. Moreno-Muñoz, D. Ramírez, and A. Artés-Rodríguez, "Continual learning for infinite hierarchical change-point detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3582–3586.

[18] T. De Ryck, M. De Vos, and A. Bertrand, "Change point detection in time series data using autoencoders with a time-invariant representation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3513–3524, 2021.

[19] H. Khan, L. Marcuse, and B. Yener, "Deep density ratio estimation for change point detection," *arXiv preprint arXiv:1905.09876*, 2019.

[20] M. Gupta, R. Wadhvani, and A. Rasool, "Real-time change-point detection: A deep neural network-based adaptive approach for detecting changes in multivariate time series data," *Expert Systems with Applications*, vol. 209, pp. 118260, 2022.

[21] Z. Atashgahi, D. C. Mocanu, R. Veldhuis, and M. Pechenizkiy, "Memory-free online change-point detection: A novel neural network approach," *arXiv preprint arXiv:2207.03932*, 2022.

[22] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[23] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *International Conference on Learning Representations*, 2018.

[24] A. Menon and C. S. Ong, "Linking losses for density ratio and class-probability estimation," in *International Conference on Machine Learning*. PMLR, 2016, pp. 304–313.

[25] C. Durkan, I. Murray, and G. Papamakarios, "On contrastive learning for likelihood-free inference," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2771–2781.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conf. on Learning Representations (ICLR)*, 2015.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. 2nd International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., Banff, AB, Canada, 2014.