# Unsupervised Online Memory-free Change-point Detection using an Ensemble of LSTM-Autoencoder-based Neural Networks

Zahra Atashgahi
Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, the Netherlands
z.atashgahi@utwente.nl

Decebal Constantin Mocanu
Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, the Netherlands
d.c.mocanu@utwente.nl

Raymond Veldhuis
r.n.j.veldhuis@utwente.nl
Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, the Netherlands

Mykola Pechenizkiy
m.pechenizkiy@tue.nl
Eindhoven University of Technology, the Netherlands, Department of Mathematics and Computer Science

## ABSTRACT

Change-point detection (CPD) is recognized as being one of the most significant tasks in time series analysis. While offline CPD has been vastly investigated in the last few years, online CPD still suffers from major challenges, such as high dependency on the choice of hyperparameters and prior information about data. However, in most real-world applications, very little prior information about the data is available, and performing hyperparameter tuning might not be feasible. Our proposed method, Adaptive LSTM-Autoencoder Change-point Detection (ALACPD), aims to address these challenges by performing unsupervised online memory-free CPD and continuously adapting itself to the current behavior of the system.

## 1 INTRODUCTION

Change-point detection (CPD) refers to the problem of finding abrupt changes in the behavior of the system. CPD is among the most significant tasks in time series analysis, as change-points contain vital information about the underlying data generating process. Existing CPD algorithms can be categorized into different types based on various criteria [2], including, (1) *Processing delay* (online or offline), (2) *Availability of labels* (supervised, semi-supervised, or unsupervised), and (3) *Data dimensionality* (univariate or multivariate). Online CPD (detect change-points with minimum possible delay after it occurs) from multivariate time series is of great importance to many real-world problems such as medical condition monitoring, image analysis, and human activity analysis. Besides, since acquiring labeled data might be a laborious task in these problems, we prefer to perform CPD in an unsupervised manner.

Despite the considerable amount of literature on CPD, many issues have not been addressed yet. A key problem with much of the literature on CPD is being highly dependent upon the choice of hyperparameters [1, 5]. However, hyperparameter-tuning might not be feasible in many real-world applications. Another major drawback is requiring prior information regarding the data [2], such as data distribution, number of change-points, or states (period of time where parameters of the data generating process do not change). To address these challenges, this paper presents a new deep learning approach to the problem of CPD, named "Adaptive LSTM-Autoencoder Change-point Detection (ALACPD)". Many works have used recurrent neural networks to perform anomaly detection in time series [4]. However, there are only a few works that have used this type of network to perform CPD. Specifically, our contributions are:

- Exploit an ensemble of LSTM-Autoencoder-based networks in combination with an Auto-regressive model (Figure 1) to perform unsupervised online memory-free CPD
- Adapt the concept of anomaly detection to the problem of change-point detection and being able to discriminate between anomalies and change-points

## 2 METHODOLOGY

**Problem Definition.** We are given a time series $T = \{x_1, x_2, ..., x_n\}$, where $x_t \in \mathbb{R}^d$ is the observation at time $t$. By using an sliding window of length $w$ across $T$, we will have $X_t = \{x_{t-w}, x_{t-w+1}, ..., x_t\}$. Time series $T$ contains $K$ change-points $P = \{t_1^*, t_2^*, ..., t_K^*\}$, where the properties of $T$ changes abruptly at each $t_k^*$, $k = \{1, ..., K\}$. CPD is the problem of approximating indices $t_k^*$.

**Network Architecture.** As shown in Figure 1, the designed network consists of two main components: (1) *LSTM-skip Autoencoder*: This component includes an encoder and a decoder part which are two LSTM-skip neural networks [3]. By adding skip connections of length $s$ between LSTM cells, LSTM-skip networks aim to address the problem of vanishing gradients of LSTMs in learning very long-term dependencies. The LSTM-skip Autoencoder receives $X_t$ as input and reconstructs it in the output $\widehat{X}_{t_{AE}}$. As a result, it can learn the data effectively and extract a hidden representation of data in the middle layer [4]. (2) *Auto-regressive (AR) component*: We employ an AR model to forecast the current $X_t$ based on the $h$ previous observations ($h$ is the time window for the AR model) as $\widehat{X}_{t_{AR}}$. It has been earlier shown in [3], that using a recurrent neural network in combination with an AR model can significantly improve the performance of time series prediction. Our model exploits an AR model to reconstruct/predict the current observation based on the previous inputs. We observed that by using an AR model in combination with an LSTM-autoencoder, the model can

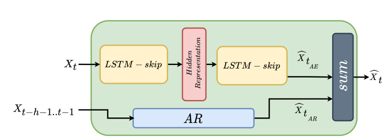


**Figure 1: Network Architecture for ALACPD**

find change-points with smaller delay (the period of time between the occurrence of the change and detecting it). By using the output of both components, we reconstruct the input observation in the output and minimize its reconstruction loss:

$$\widehat{X}_t = \widehat{X}_{t_{AE}} + \widehat{X}_{t_{AR}}, \quad minimize \sum_t \left\| X_t - \widehat{X}_t \right\|_2^2 . \tag{1}$$

**Proposed Method.** This section explains how we use the designed LSTM-Autoencoder network to detect change-points from the time series in an unsupervised online memory-free manner. In short, our proposed method, ALACPD, exploits the concept of anomaly detection [4] to perform CPD; if we receive an observation $X_t$ with low reconstruction error (lower than a threshold $th_{state}$) which is computed based on the final output $\widehat{X}_t$, it means that this sample belongs to the current state. However, if we receive several anomalies (samples with reconstruction error higher than $th_{state}$) in a row, it appears that the underlying data distribution is changing, and we are observing a new state in the system. We should highlight that this assumption has been made based on the nature of anomalies that usually occur in a very short period of time.

**Offline Training.** To start training, we assume that the first 5% of the data contains no change-points. We train the model on these samples. Our model is an ensemble of three sub-networks with the same architecture as Figure 1 and different skip sizes to learn dependencies at different scales. Then, we calculate the threshold $th_{state}$ for each sub-network which is a proportion of the mean reconstruction error of the samples belonging to the current state.

**Online Training.** When a new data sample $X_t$ is collected, first, we compute the reconstruction error for each sub-network using the output of each sub-network. Then, we compare it with $th_{state}$ of the corresponding sub-network: (1) If the error is lower than $th_{state}$ for at least two sub-networks, $X_t$ belongs to the current state; we update the parameters of the model by training on this sample. After that, we update the $th_{state}$ for each sub-network using the reconstruction error of $X_t$ and then discard the sample. (2) If the error is higher than $th_{state}$ for at least two sub-networks, we report this sample as an anomaly, and we do not update the model. However, if we detect $m$ anomalies in a row, we report this as a change-point $\widehat{t}_i^*$. Then, we re-train each sub-network on this sequence of anomalies which are now a new state in the system. Then, we compute $th_{state}$ for the new state for all sub-networks and continue online training.

## 3 EXPERIMENTS

In this section, we evaluate our proposed model and compare it with several state-of-the-art CPD algorithms.

**Settings.** The settings for our experiments are similar to the experiments performed in [5]. In this paper, the authors performed a thorough comparison among the most well-known CPD algorithms on several datasets. They introduced two metrics to measure the performance, covering (to measure the quality of the states), and F-score (to measure the quality of the estimated change-points). We compared our method with six CPD methods and evaluated all methods on three multivariate datasets. More details about the evaluation metrics, datasets, and methods, can be found in [5].

**Results.** The results of the experiments are presented in Table 1. We do not perform hyperparameter tuning in our experiments. As shown in this table, our proposed method outperforms all the other

**Table 1: Performance comparison among methods. Each entry contains the covering score (above), and F-score (below).**

| Dataset | BOCPD | ECP | KCPA | BOC PDMS | RBOC PDMS | ZERO | ALACPD | ALACPD w/oAR |
|---|---|---|---|---|---|---|---|---|
| Apple | 0.365<br>0.513 | 0.305<br>0.513 | 0.007<br>0.029 | 0.298<br>0.318 | 0.424<br>0.373 | 0.425<br>0.594 | **0.688**<br>**0.634** | 0.611<br>0.695 |
| Occupancy | 0.549<br>**0.807** | 0.530<br>0.75 | 0.05<br>0.107 | 0.459<br>0.496 | 0.467<br>0.29 | 0.236<br>0.341 | **0.594**<br>0.663 | 0.565<br>0.49 |
| Run_log | **0.815**<br>**1.00** | 0.657<br>0.792 | 0.128<br>0.139 | 0.302<br>0.42 | 0.439<br>0.543 | 0.304<br>0.446 | 0.73<br>0.844 | 0.73<br>0.891 |

methods on the Apple dataset in terms of covering and F-score and on Occupancy in terms of covering. In addition, on Run_log dataset, ALACPD is the second-best performer in terms of both metrics considered. The major competitor of our proposed method is BOCPD. This method assumes that the time series is i.i.d, and it is known as being sensitive to hyperparameters. However, ALACPD does not make any assumptions about the underlying data distribution. In addition, by adapting the thresholds continuously on the new observations, it performs well with its default hyperparameters on all datasets considered. Moreover, ALACPD extracts a hidden representation of the time series in its hidden layer, which can be used in other tasks, such as classification. Finally, ALACPD can detect anomalies which makes it superior to the methods that are sensitive to outliers. Besides, we performed an ablation study to measure the effect of the AR component on the performance of ALACPD. As shown in the last column of Table 1, ALACPDw/oAR (ALACPD without the AR component) has been outperformed by ALACPD in most of the cases considered.

## 4 CONCLUSION

In this paper, we presented a new deep learning approach for the problem of change-point detection, named ALACPD. Using an LSTM-Autoencoder to learn long-term dependencies, in combination with an Auto-regressive model to respond to any changes in the data rapidly, our model can detect changes in an unsupervised online memory-free manner. Our findings demonstrate that ALACPD outperforms existing statistical CPD algorithms in terms of the quality of the detected change-points, in most cases considered. To further our research, we intend to extend our method to perform hierarchical CPD.

## ACKNOWLEDGMENTS

This research has been funded by the NWO EDIC project.

## REFERENCES

[1] Ryan Prescott Adams and David JC MacKay. 2007. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742* (2007).

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

[3] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.* 95–104.

[4] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148* (2016).

[5] Gerrit JJ van den Burg and Christopher KI Williams. 2020. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222* (2020).