$TS-CP^2$

Time-series Change Point Detection with Self-Supervised Contrastive Predictive Coding

Shohreh Deldari, Hao Xue, Daniel V. Smith, Flora Salim

MAIN GOAL

- Extract
  - **Informative**
  - **Compact**
  - **Predictable**

representative feature vectors from semantic *segments* of multi-variate *Time-series* data

CONTRIBUTION

- TS-CP2 is the first that leverages **contrastive learning** as an unsupervised objective function for **time-series data**.
- TSCP2 captures **compact, latent embeddings** that **represent historical and future time intervals** of the times series.
- TSCP2 is the first to apply contrastive learning for the problem of **self-supervised change point detection (CPD)**.

METHOD

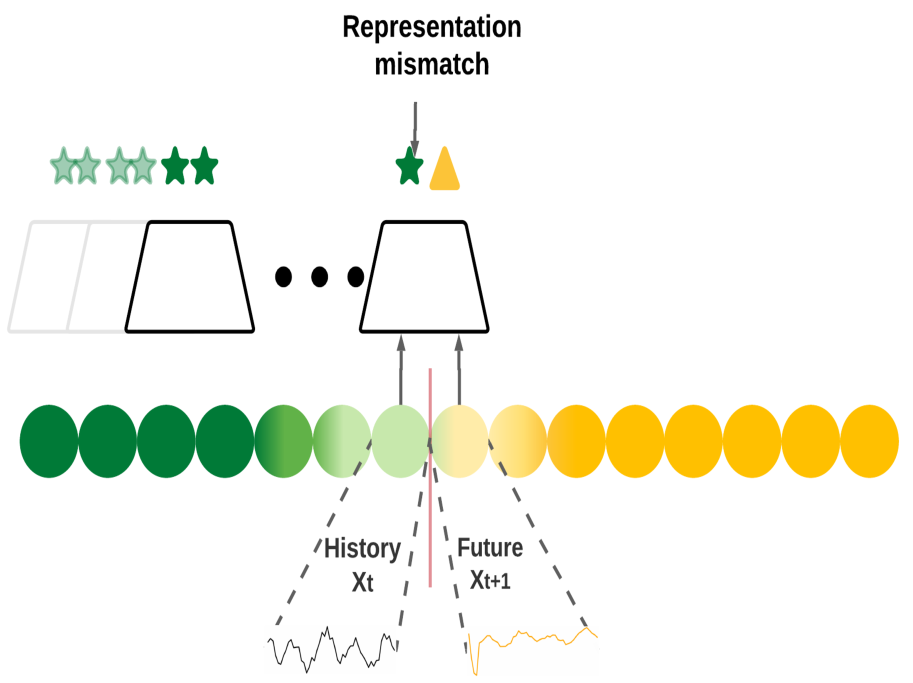


Figure 1- Overview of presented approach based on predictive representation learning.

- We propose a multi-task model to
  1. Learn representation using Contrastive learning
  2. Predict future based on Contrastive Predictive Coding
- We use the representation learning task as an auxiliary task to do Change Point Detection.

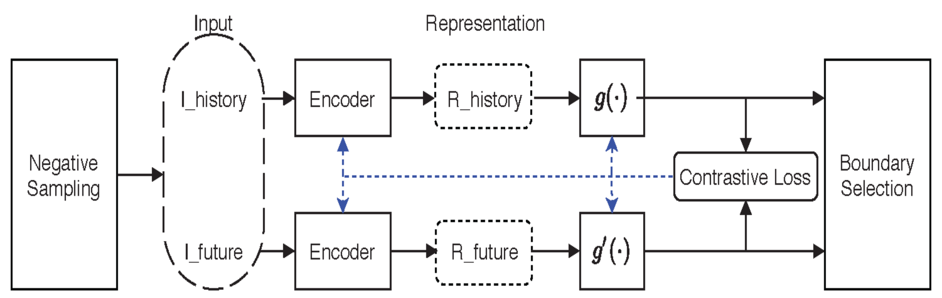


Figure 2- Illustration of the overall architecture of our$TS-CP2$. Blue dash arrows indicate the back propagation in the training phase

TS-CP2 COMPONENTS

- **Negative Sampling**
- we propose a simple sampling strategy where positive sample pairs are randomly sampled and used to construct the negative sample pairs for each batch.
- Each pair must adhere to the constraint of being a minimum temporal distance from the others.
- The intuition is that time series are commonly non-stationary, and hence, windows that are temporally far are likely to exhibit weaker statistical dependencies than adjacent frames.

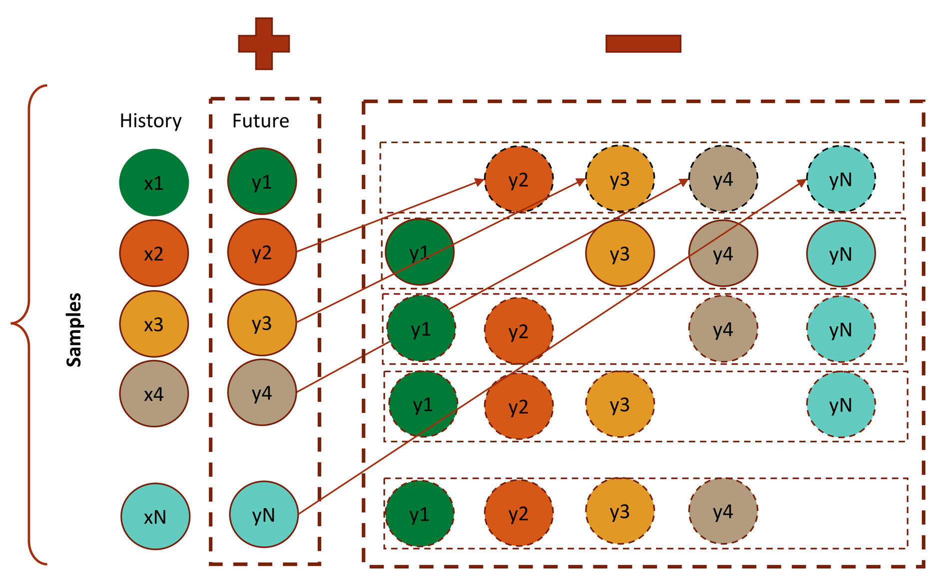


Figure 3- Negative Sampling approach

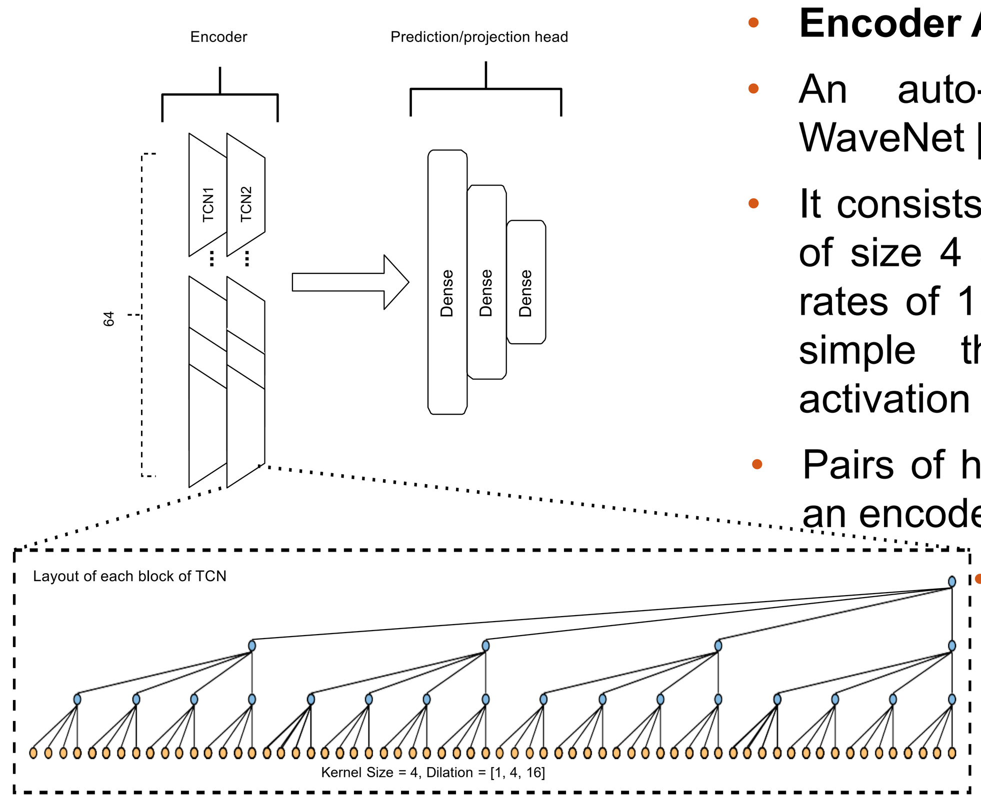


Figure 4- The encoder architecture

- **Encoder Architecture**
- An auto-regressive deep convolution network, WaveNet [1], was employed to encode each window.
- It consists of two blocks of TCN with 64 kernel filters of size 4 and three layers of dilation with respective rates of 1, 4 and 16. The TCN is then followed by a simple three-layer projection head with ReLU activation function and batch normalisation.
- Pairs of history and future time windows are fed into an encoder.
- A projection head (an MLP neural network with three hidden layers) is used ($g(\cdot)$ and $g'(\cdot)$ in Figure 2) to map each window encoding into a lower dimension space.

- **Objective Function**
- we adopt a similar approach to the CPC [2,3] to learn a representation that

maximises the mutual information between consecutive time windows.

- We employed InfoNCE [2] loss :

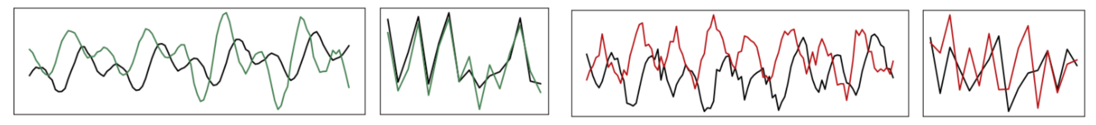
$$\rho_{ij} = \frac{\exp(Sim(h_i, f_j)/\tau)}{\sum_j^K \exp(Sim(h_i, f_j)/\tau)} \qquad Sim(h_i, f_j) = cosine\_sim(h_i, f_j) \qquad LOSS = \sum_i -log(\rho_i)$$

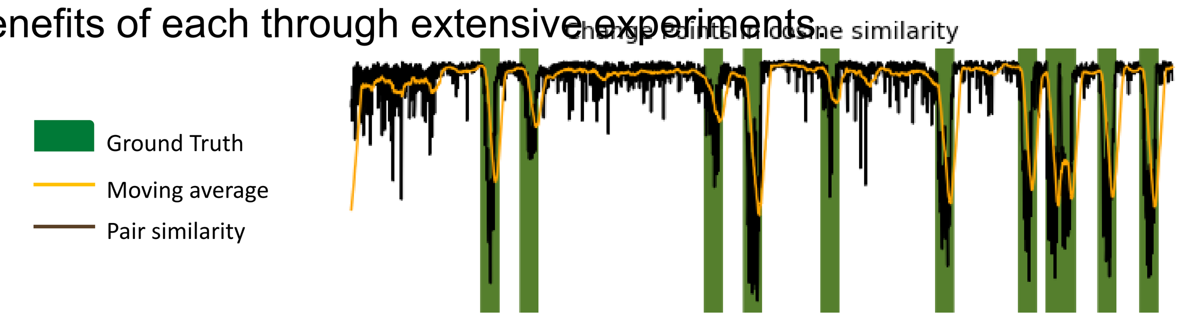
DATASET

Dataset	T	#sequences	#channels	#CP
Yahoo! Benchmark	164K	100	1	208
HASC	39K	1	3	65
USC-HAD	97K	6	3	30

Table 1: Dataset characteristics. T is the total number of data samples, sequences and channels show the number of time-series and dimensions, respectively, and CP is the total number of change points.

EXPERIMENTS

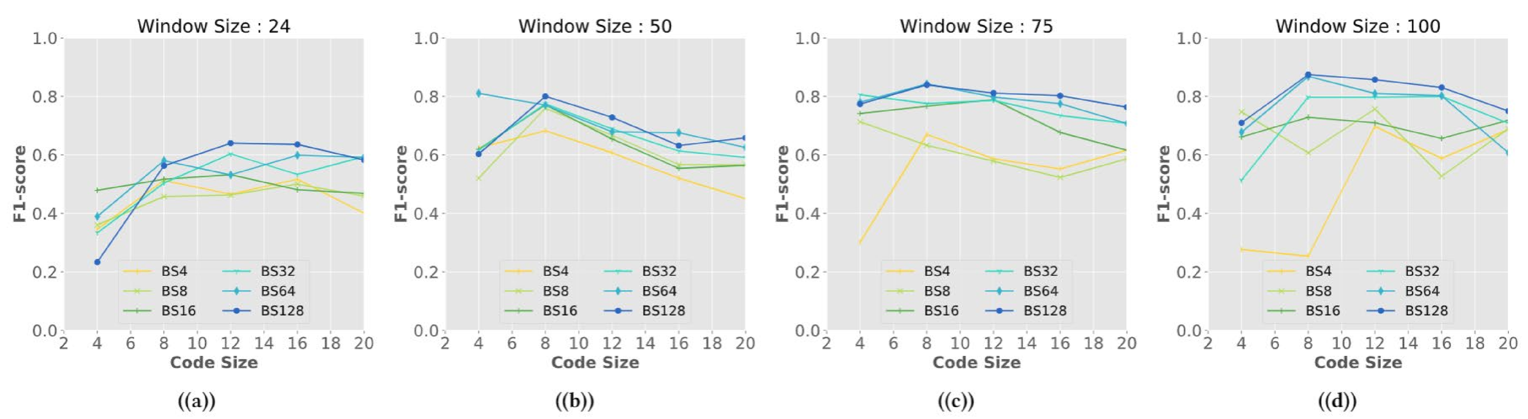
- **Extracted Representations**
- Examples of positive(Left) and Negative (right) pair representation
- 
- **Change Point Detection**
- We compare our proposed method against five state-of-the-art CPD methods, which include deep learning and non-deep learning-based methods, investigate the benefits of each through extensive experiments.



$TS-CP2$ improved performance of Non-DL and DL baselines by 79.4% and 17.0% respect to the F1-score.

Dataset	Maximum Delay	24		50		75	
	Methods	Best Wnd	F1-score	Best Wnd	F1-score	Best Wnd	F1-score
Yahoo	FLOSS	45	0.2083	50	0.3375	55	0.4233
	aHSIC	40	0.4092	40	0.4175	40	0.4392
	RuLSIF	20	0.3175	20	0.3317	20	0.37
	ESPRESSO	50	0.2242	50	0.34	70	0.4442
	KLCPD ICLR2020	24	0.5787	50	0.5760	75	0.5441
	$TS-CP^2$	24	**0.64**	50	**0.8104**	75	**0.8428**
USC	Maximum Delay	100		200		400	
	FLOSS	100	0.2666	100	0.3666	400	0.4333
	aHSIC	50	0.3333	50	0.3333	50	0.4
	RuLSIF	400	0.4666	400	0.4666	400	0.5333
	ESPRESSO	100	0.6333	100	0.8333	100	**0.8333**
	KLCPD ICLR2020	win:100, bs:4	0.7426	win:200,bs:32	0.7180	win:400,bs:16	0.6321
	$TS-CP^2$	win:100, bs:8	**0.8235**	win:200, bs:8	**0.8571**	win:400, bs:32	**0.8333**
HASC	Maximum Delay	60		100		200	
	FLOSS	60	0.3088	60	0.3913	100	0.5430
	aHSIC	40	0.2308	40	0.3134	40	0.4167
	RuLSIF	200	0.3433	200	0.5	200	0.5
	ESPRESSO	100	0.2879	60	0.4233	100	**0.6933**
	KLCPD ICLR2020	win:60,bs:4	**0.4785**	win:100,bs:4	**0.4726**	win:200,bs:64	0.4669
	$TS-CP^2$	win:60,bs:64	0.40	win:100,bs:64	0.4375	win:200,bs:64	0.6316

- **Sensitivity Analysis**
- We investigate the performance impact of the hyperparameters including batch size, code size, and window size



REFERENCES

[1] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, OriolVinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. (2016)

[2] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learningwith contrastive predictive coding. (2018)

[3] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SMEslami, and Aaron van den Oord. 2020. Data-efficient image recognition withcontrastive predictive coding. ICML 2020

Source code is provided on the GitHub page.