# Time-Series Representation Learning via Temporal and Contextual Contrasting

Emadeldeen Eldele , Mohamed Ragab , Zhenghua Chen,

Min Wu, Chee KeongKwoh , Xiaoli Li and Cuntai Guan

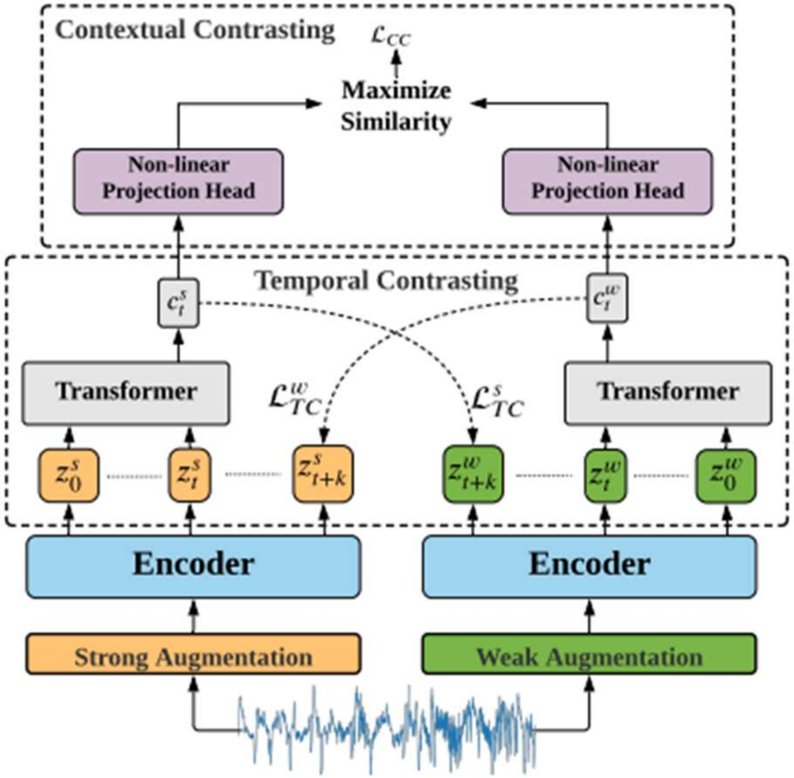
IJCAI 2021

# Introduction

- Time-series data generally do not have human recognizable patterns and require specialists for annotation/labeling.

- Some image-based contrastive learning methods are not able to work on time-series data for the following reasons:
  - They may not able to address the temporal dependencies of data.
  - Some augmentation techniques used for image generally cannot fit well with time-series data.

- A framework, **T**ime-**S**eries representation learning via **T**emporal and **C**ontextual **C**ontrasting (TS-TCC), were proposed
  - Employing simple data augmentations that can fit any time-series data to create two different, but correlated views of input data.
  - **Temporal contrasting** module to learn robust representations by designing a tough cross-view prediction task.
  - **Contextual contrasting** module to further learn discriminative representations.



# Data augmentation

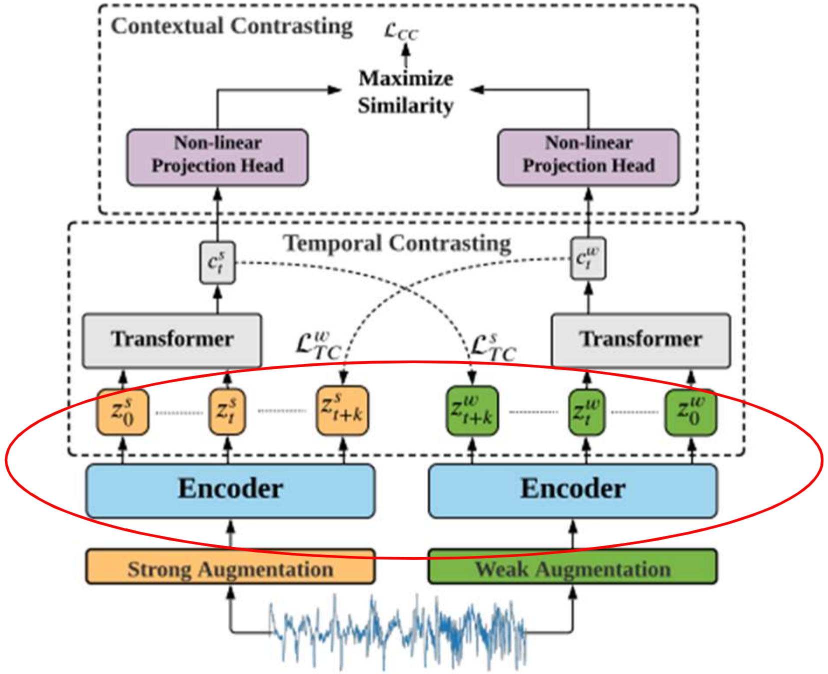
- Contrastive methods try to maximize the similarity among different views of the same sample, while minimizing its similarity with other samples.

- In this paper, two augmentations were proposed, such that one is weak and the other is strong.
  - Weak: jitter-and-scale, adding random variation to the signal and scale up its magnitude.
  - Strong: permutation-and-jitter, splitting signal into a number of segments and randomly shuffling them; next, a random jittered is added to the permuted signal.

- For each sample $x$, we denote it strongly augmented view as $x^s$, and its weakly augmented view as $x^w$.

# Model

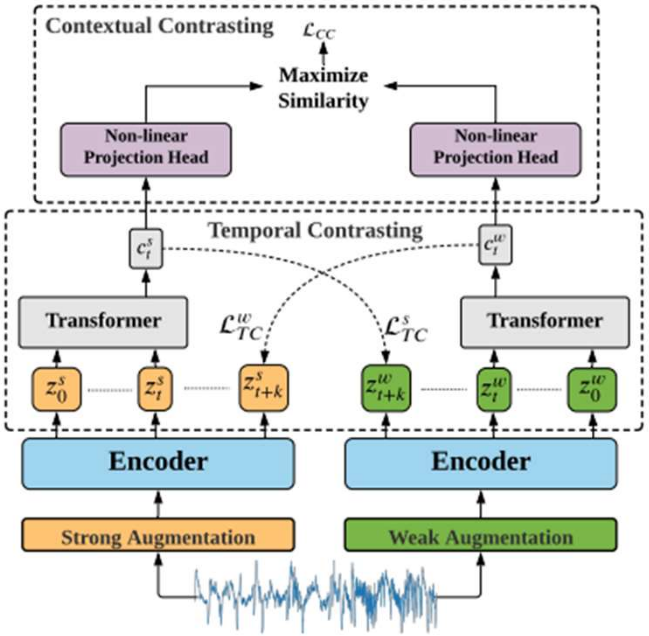
- Encoder
  - 3-block convolution architecture.
  - Maps $x$ into a high-dimensional latent representation $z = f_{enc}(x)$.
  - We get $z^s$ for the strong augmented views, and $z^w$ for the weak augmented views.



# Temporal contrasting

- Extracts temporal features in the latent space with an autoregressive model.
  - Autoregressive model $f_{ar}$ summarizes all $z_{\leq t}$ into a context vector $c_t = f_{ar}(z_{\leq t})$.
  - The context vector $c_t$ is used to predict the timesteps from $z_{t+1}$ to $z_{t+k}$.

- Cross-view prediction task
  - Using the context of the strong augmentation $c_t^s$ to predict the future timesteps of the weak augmentation $z_{t+k}^w$, and vice versa.

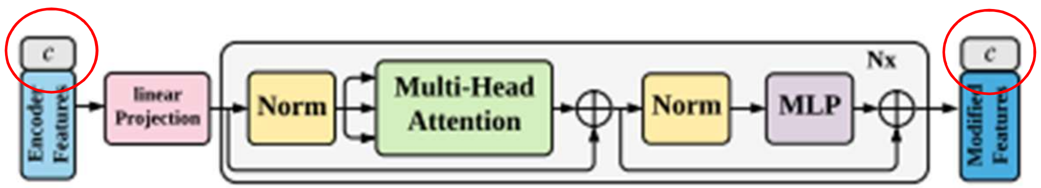


- Two losses

$$\mathcal{L}_{TC}^{s} = -\frac{1}{K}\sum_{k=1}^{K}\log\frac{\exp((\mathcal{W}_k(c_t^s))^T z_{t+k}^w)}{\sum_{n\in\mathcal{N}_{t,k}}\exp((\mathcal{W}_k(c_t^s))^T z_n^w)} \quad (1)$$

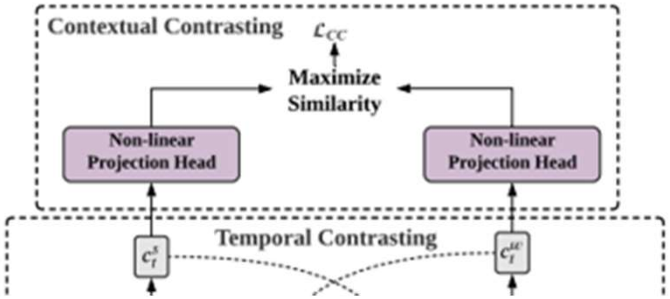
$$\mathcal{L}_{TC}^{w} = -\frac{1}{K}\sum_{k=1}^{K}\log\frac{\exp((\mathcal{W}_k(c_t^w))^T z_{t+k}^s)}{\sum_{n\in\mathcal{N}_{t,k}}\exp((\mathcal{W}_k(c_t^w))^T z_n^s)} \quad (2)$$

- Using Transformer as the autoregressive model.
  - Pre-norm residual connection: for stable gradients.
  - Add a token *c* to the input whose act as a representative context vector in the output (Like [CLS] in BERT).



# Contextual contrasting

- Aims to learn more discriminative representations.
  - It starts with a non-linear transformation to the context.




- Given a batch $N$ samples, we will have two contexts for each sample from its two augmented views, and thus have $2N$ contexts.
  - Positive $c_t^{i^+}$ : comes from the other augmented view of the same input (2).
  - Negative $c_t^{i^-}$ : from other inputs within same batch (2$N$-2).

- Contrastive loss

$$\mathcal{L}_{CC} = -\sum_{i=1}^{N} \log \frac{\exp\left(\operatorname{sim}\left(\boldsymbol{c}_t^i, \boldsymbol{c}_t^{i^+}\right)/\tau\right)}{\sum_{m=1}^{2N} \mathbb{1}_{[m\neq i]} \exp\left(\operatorname{sim}\left(\boldsymbol{c}_t^i, \boldsymbol{c}_t^m\right)/\tau\right)}, \tag{5}$$

- Overall self-supervised loss

$$L = \lambda_1 \cdot (L_{TC}^{s} + L_{TC}^{w}) + \lambda_2 \cdot L_{CC}$$

$\lambda_1, \lambda_2$ are fixed scalar hyperparameters.

# Experimental results

- Datasets
  - Human Activity Recognition (HAR): 6 activities, ex. walking, standing,…
  - Sleep Stage Classification (Sleep-EDF): 5 classes, ex. wake, rapid eye movement,…
  - Epilepsy Seizure Prediction (Epilepsy): 2 classes, ex. True / False
  - Fault Diagnosis (FD): 4 different working conditions, and each contains 3 classes: healthy, inner fault, and outer fault.

| Dataset | # Train | # Test | Length | # Channel | # Class |
|---|---|---|---|---|---|
| HAR | 7352 | 2947 | 128 | 9 | 6 |
| Sleep-EDF | 25612 | 8910 | 3000 | 1 | 5 |
| Epilepsy | 9200 | 2300 | 178 | 1 | 2 |
| FD | 8184 | 2728 | 5120 | 1 | 3 |

- Baselines
  - **Random initialization**: training a linear classifier on top of randomly initialized encoder.
  - **Supervised**: supervised training of both encoder and classifier model.
  - **SSL-ECG**: self-supervised learning through recognition of 6 different transformations.
  - **CPC**: contrastive predictive coding, pretrained by predicting the latent vector on future timesteps.
  - **SimCLR**: using time-series specific augmentations to adapt SimCLR.

- Evaluation metrics
  - Accuracy (ACC)
  - Macro-averaged F1-score (MF1): Arithmetic mean of all the per-class F1 scores.

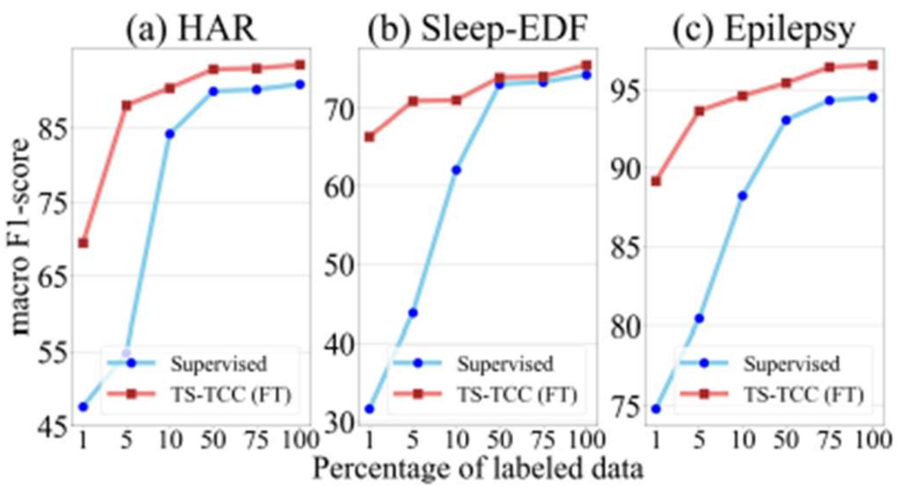
## TS-TCC v.s. baseline methods

- TS-TCC outperforms all the three state-of-the-art methods.
- Contrastive methods generally achieve better results than the pretext-based method, which reflects the power of invariant features learned by contrastive methods. (CPC, SimCLR, TS-TCC <-> SSL-ECG)
- CPC method shows better results than SimCLR, indicating that temporal features are more important than general features in time-series data.

| | HAR | | Sleep-EDF | | Epilepsy | |
|---|---|---|---|---|---|---|
| Baseline | ACC | MF1 | ACC | MF1 | ACC | MF1 |
| Random Initialization | 57.89±5.13 | 55.45±5.49 | 35.61±6.96 | 23.80±7.96 | 90.26±1.77 | 81.12±4.22 |
| Supervised | 90.14±2.49 | 90.31±2.24 | **83.41±1.44** | **74.78±0.86** | 96.66±0.24 | 94.52±0.43 |
| SSL-ECG [P. Sarkar, 2020] | 65.34±1.63 | 63.75±1.37 | 74.58±0.60 | 65.44±0.97 | 93.72±0.45 | 89.15±0.93 |
| CPC [Oord *et al.*, 2018] | 83.85±1.51 | 83.27±1.66 | 82.82±1.68 | 73.94±1.75 | 96.61±0.43 | 94.44±0.69 |
| SimCLR [Chen *et al.*, 2020] | 80.97±2.46 | 80.19±2.64 | 78.91±3.11 | 68.60±2.71 | 96.05±0.34 | 93.53±0.63 |
| TS-TCC *(ours)* | **90.37±0.34** | **90.38±0.39** | 83.00±0.71 | 73.57±0.74 | **97.23±0.10** | **95.54±0.08** |

Table 2: Comparison between our proposed TS-TCC model against baselines using linear classifier evaluation experiment.

- Semi-supervised Training
  - Supervised training v.s. TS-TCC
  - Training the model with 1%, 5%, 10%, 50%, and 75% of randomly selected instances of the training data.
  - TS-TCC (FT): Fine-tuned the pretrained encoder with few labeled samples.
  - TS-TCC (FT) achieves significantly better performance than supervised training with only 1% of labeled data.



## Transfer Learning Experiment

- Fault Diagnosis (FD): Containing 4 different working condition, each has different characteristics from the other working conditions.
- Training the model on one condition (source domain) and test it on another condition (target domain).
- Supervised v.s. TS-TCC

| | A→B | A→C | A→D | B→A | B→C | B→D | C→A | C→B | C→D | D→A | D→B | D→C | AVG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Supervised | 34.38 | 44.94 | 34.57 | **52.93** | 63.67 | **99.82** | **52.93** | 84.02 | 83.54 | **53.15** | 99.56 | 62.43 | 63.83 |
| TS-TCC *(FT)* | **43.15** | **51.50** | **42.74** | 47.98 | **70.38** | 99.30 | 38.89 | **98.31** | **99.38** | 51.91 | **99.96** | **70.31** | **67.82** |

- Ablation study
  - **TC-only**: predict the future timesteps of the same augmented view.
  - **TC + X-Aug**: TC + adding the cross-view prediction.
  - **TC + X-Aug + CC (TS-TCC)**: proposed TS-TCC model.
  - **TS-TCC (Weak only)**: generate two different views from the weak augmentation.
  - **TS-TCC (Strong only)**: generate two different views from the strong augmentation.

| | HAR | | Sleep-EDF | | Epilepsy | |
|---|---|---|---|---|---|---|
| Component | ACC | MF1 | ACC | MF1 | ACC | MF1 |
| TC only | 82.76±1.50 | 82.17±1.64 | 80.55±0.39 | 70.99±0.86 | 94.39±1.19 | 90.93±1.41 |
| TC + X-Aug | 87.86±1.33 | 87.91±1.09 | 81.58±1.70 | 71.88±1.71 | 95.56±0.24 | 92.57±0.29 |
| TS-TCC (TC + X-Aug + CC) | **90.37±0.34** | **90.38±0.39** | **83.00±0.71** | **73.57±0.74** | **97.23±0.10** | **95.54±0.08** |
| TS-TCC (Weak only) | 76.55±3.59 | 75.14±4.66 | 80.90±1.87 | 72.51±1.74 | 97.18±0.17 | 95.47±0.31 |
| TS-TCC (Strong only) | 60.23±3.31 | 56.15±4.14 | 78.55±2.94 | 68.05±1.87 | 97.14±0.23 | 95.39±0.29 |

# Conclusions

- Temporal contrasting module learns robust temporal features by applying a tough cross-view prediction task.

- Contextual contrasting module to learn discriminative features upon the learned robust representations.

- TS-TCC shows high efficiency on few-labeled data and transfer learning scenarios.

*Thank you for your attention.*