# TIME-SERIES REPRESENTATION LEARNING VIA TEMPORAL AND CONTEXTUAL CONTRASTING (TS-TCC)

Shubham Gurjar (210260048), Atharv Tambade (210260057), Atharv Deshpande (210260012), Siddharth Stephen (210260049), Godwin Arunim Beck (210260016)

## **Abstract**

Learning decent representations from unlabeled time-series data with temporal dynamics is a very challenging task. Here we use an unsupervised Time-Series representation learning framework via Temporal and Contextual Contrasting (TS-TCC), to learn time-series representation from unlabeled data. First, the raw timeseries data are transformed into two different yet correlated views by using weak and strong augmentations. Second, we use a temporal contrasting module to learn robust temporal representations by designing a tough cross-view prediction task. Last, to further learn discriminative representations, a contextual contrasting module is built upon the contexts from the temporal contrasting module. It attempts to maximize the similarity among different contexts of the same sample while minimizing similarity among contexts of different samples. Experiments have been carried out on three real-world time-series datasets. TS-TCC shows high efficiency in few-labeled data and transfer learning scenarios.

#### 1. Introduction

Time-series data are often collected without humanrecognizable patterns and are challenging to label. This lack of labeled data makes it difficult to apply traditional supervised learning methods, which typically require a large amount of labeled training data. There are various limitations of labeled data, so it's in our interest to use self supervised learning. where we can train models on unlabeled data by solving pretext tasks. This approach allows for effective representation learning even with limited labeled data.

# 2. Problems we are addressing

- Pretext Tasks Limitations: Traditional self-supervised methods, which rely on pretext tasks like image rotation prediction or puzzle solving, may not be suitable for timeseries data. As they won't capture the temporal dependencies
- Contrastive Learning for Time-Series: valuating and comparing automated approaches because most methods are compared to a single human scorer's annotations, so it may not provide an accurate assessment of the performance
- Lack of Generalizable Methods: While some contrastive learning methods have been proposed for specific applications in time-series data, they may not be generalizable to various types of time-series data

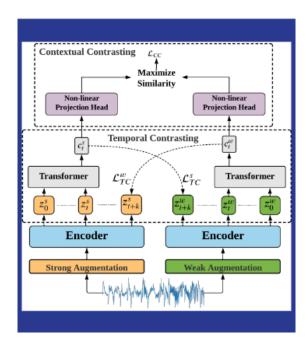


Figure 1: TS-TCC architecture

# 3. Input Sources:

- 3.1. Sleep EDF
  - EEG signal data

Preprint submitted to TS-TCC November 26, 2023

- To classify into five different stages: Wake (W), Non-rapid eye movement (N1, N2, N3) and Rapid eye movement (REM).
- Includes whole-night PSG sleep recordings, with a single EEG channel (sampling rate 100Hz).

## 3.2. Human Activity Recognition

- The UCI HAR dataset
- To classify activity of a human into six categories walking, walking upstairs, downstairs, standing, sitting, and lying down.
- Data collected from a wearable device

#### 3.3. Epilepsy

- The Epilepsy Seizure Recognition dataset, again EEG recordings
- The five classes in the dataset merged into two seizure and no seizure
- So then, binary classification

## 4. Implementation

We split the data into 60%, 20%, 20% for training, validation and testing, with considering subject-wise split for Sleep-EDF dataset to avoid overfitting and data leakage The pretraining and downstream tasks were done for 40 epochs, as we noticed that the performance does not improve with further training.

## 5. Framework of TS-TCC model

## 5.1. Data Augmentation for Time-Series

The framework starts by generating two different, yet correlated views of the input time-series data Two types of data augmentations are proposed: weak augmentation and strong augmentation Weak augmentation includes a jitter-and-scale strategy, adding random variations to the signal and scaling its magnitude. Strong augmentation includes a permutation-and-jitter strategy, which involves splitting the signal into segments and shuffling them, followed by adding random jitter. Careful consideration of augmentation hyperparameters is essential to match the nature of the time-series data.

# 5.2. Temporal Contrasting

The Temporal Contrasting module is designed to explore the temporal features of the data using an autoregressive model, particularly a Transformer model. The Transformer model includes successive blocks of multi-headed attention (MHA) and a multi-layer perceptron (MLP). The autoregressive model summarizes previous time steps into a context vector. A tough cross-view prediction task is defined, where one view's context is used to predict future timesteps in the other view and vice versa. A contrastive loss is employed to encourage agreement

between the predicted representations and the true ones for the same sample while minimizing similarity with other samples.

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

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

Figure 2: The Temporal Contrasting Loss Functions

These loss functions are implemented as shown in figure 1.

## 5.3. Contextual Contrasting

The Contextual Contrasting module is introduced to learn more discriminative representations. Non-linear transformations are applied to the context vectors using a non-linear projection head. Positive pairs are defined, consisting of the context vector and its corresponding positive sample from the other augmented view of the same input. Negative pairs are formed by the context vector and other contexts within the same batch. A contextual contrasting loss is defined to maximize the similarity between positive pairs and minimize the similarity between negative pairs.

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

Figure 3: The Contextual Contrasting Loss

## 6. Results

	HAR		Sleep	-EDF	Epilepsy	
Baseline	ACC	MF1	ACC	MF1	ACC	MFI
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	$90.31\pm2.24$	83.41±1.44	$74.78 \pm 0.86$	96.66±0.24	$94.52\pm0.43$
SSL-ECG [P. Sarkar, 2020]	65.34±1.63	$63.75\pm1.37$	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	$73.94 \pm 1.75$	96.61±0.43	94.44±0.69
SimCLR [Chen et al., 2020]	80.97±2.46	$80.19\pm2.64$	78.91±3.11	$68.60\pm2.71$	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

Figure 4: Comparison between our proposal TS-TCC model against baselines using linear classifier evaluation experiment

		TT IFBLE LOSS	: 0.2191   IPSIN ACCURSCY	8 9250
Dataset: HAR		Valid toss	: 0.4982   Valid Accuracy	
mothod: TS-1				
Mode: rand	iom_init	Epoch: 33		
			: 0.2233   Train Accuracy	
Data loaded .				
Training star	rted			
		Epoch : 34		
Epoch : 1		Train Loss	: 0.2252   Train Accuracy	
		ah Valid Loss	: 0.4684   Valid Accuracy	
		Epoch : 85		
Epoch : 2				
Train Loss		ni Valld tons		
Valid Loss	: 0.6936 Va			
		Epoch : 36		
		Train Loss	: 0.2274   Train Accuracy	
			1 0.4874   Valid Accuracy	
		Epoch : 37		
Epoch : 4		train tess Valid tess	: 0.2515   India Accuracy : 0.4710   Valid Accuracy	
Train Loss	: 0.4164 Tr	5.6	: 0.4/10   Valid Accuracy	
Valid Loss	1 0.6895 Va	11 Epoch : 38		
		Train Loss	: 0.2277   Train Accuracy	
			1 0.4724   Valid Accuracy	0.7966
			1 0.4724   Valla Accordey	
Valid toss		Li French : 39		
		train toss	: 0.2200   Train accuracy	
Epoch : 6			: 0.4836   Valid Accuracy	8 7985
Train Loss		81	- Table - Control	
Valid Loss	1 0.6646   Val	IIi Epoch   40		
		Train Loss	: 0.2281   Train Accuracy	
Epoch : 7	: 0.3569   Te	· Valid Loss	1 0.4727   Valid Accuracy	
Train Loss				
Valid Loss		Evaluate on t	the Test set:	
		fest loss	18.5881   Test Accuracy	0.7575

Dataset: HAR		Feoch : 23			
Hethod: TS-		Train Loss		Train Accuracy	1 0.9681
	ervised	wallid Loss		Valid Accuracy	0.9531
		V2110 1001		varia securacy	. 4
Data loaded		Epoch : 34			
Training sta	rted	Train Loss		Train Accuracy	0.9681
		Valid Loss		Valid Accuracy	0.9560
Epoch : 1					
Train Loss	0.6498	lipoch : 35			
valid Loss	0.6318	Train toss		Train Accuracy	8.9781
******		Walid toss		Valid Accuracy	0.9609
Fpoch : 2					
Train Loss		Epoch : 36			
Wallid Loss		Train Loss		Train Accuracy	0.9783
Willia Lines		Welid Loss		Valid Accuracy	0.9595
Cooch 1 3					
Train Loss					
Valid Loss		train toss		Iruin Accuracy	0.9641
Valid Loss		wallid toss		valid accuracy	01.59524
Epoch : 4		Epoch : 38			
Irain Loss		Train Loss		Train Accuracy	0.9670
Walid Loss		valid Loss		Valid Accuracy	0.9673
Epoch : 5					
Train Loss		Train Loss		Train Accuracy	0.9093
Valid Loss		Valid Loss		Valid Accuracy	0.9560
Epoch : 6		Epoch : 48			
Train Loss		Train Loss		Train Accuracy	0.9696
Walid Loss		Valid Loss		Valid Accuracy	0.9574
Epoch: 7		Evaluate on t			
Train Loss				Test Accuracy	1 0,9555
Valid Loss					
		**********	eesees Trainic	ng is Donel memeran	***********

These above two are for the random initialization and supervised learning approaches on the HAR dataset.

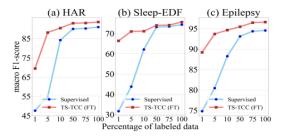
lataset: sle lethod: 15-1	ICC	train toss			
tode: ran	ion_init	valid toss		valid Accuracy	
ata loaded		Fpoch : 34			
fraining star					
		Valid Loss		Valid Accuracy	
Epoch : 1					
Frain Loss		η Epoch : 35			
Falid Loss		Train Loss		Train Accuracy	
		Walid Loss		Welid Accuracy	
Epoch : 2					
		Epoch : 36   Train toss		Train Accuracy	
talld toss		Valid Loss			
		Value Loss		Valid Accuracy	
Epoch : 3		- Epoch : 37			
Frain Loss		Train Loss		Train Accuracy	
Falid Loss		Valid Loss	2.6124	Valid Accuracy	0.5840
		WALLS LOSS	. 226.00	WELLS ACCUPACY	
ipenth : 4		, Epoch : 38			
		Train Loss		Train Accuracy	
falld toss		Valid Loss		Valid Accuracy	
		WALLES COSS		WELLIN PECCUPACY	
Epoch : 5		- Epoch : 39			
Train Loss		. Train Loss		Train Accuracy	0.7989
ralid Loss		Valid Loss		Valid Accuracy	0.5842
		*01.55 0.55		watto recovery	
ipoch : 6		, Epoch : 40			
Irain Loss		Train Loss	9,5579	Train Accuracy	: 0.7981
falid toss		valid Loss	2.6845	Valid Accuracy	
spoch : 7		. Evaluate on t	the Test set		
rrain Loss		rest loss		Test Accuracy	
ralid Loss					

Dataset: Epilepsy Method: TS-TCC Mode: random_init	Egoch   33   10.1168   Train Accuracy   10.5957   valid Loss   10.2992   Valid Accuracy   10.5958
Data loaded Training started	Fpoch : 34 Train Loss   1 0.1219   Train Accuracy   1 0.9555 Valid Loss   1 0.2962   Valid Accuracy   1 0.9588
Epoch : 1 Trein Loss : 8.2628   Walid Loss : 8.4274	Epoch   35 trada coss   1 0.1218   Trada Accuracy   1 0.9546 Walld coss   1 0.2979   Walld Accuracy   1 0.9266
Epoch : 2 Train Loss : 8.3719   Walid Loss : 8.3291	Epoch   46 Irain Loss : 0.1189   Irain Accuracy : 0.9564 Valid Loss : 0.2975   Valid Accuracy : 0.998
Epoch : 3 train toss : 8.3664   valid toss : 8.2783	Epoch   37 Train Loss   1 0.1200   Train Accuracy   1 0.9546 Walld Loss   1 0.2953   Walld Accuracy   1 0.9213
Fpoch : 4 Train toss : 0.3634   Valid toss : 0.2487	Epoch : 48 Train Luss : 0.1194   Train Accuracy : 0.9776 Valid Luss : 0.2918   Valid Accuracy : 0.9774
Epoch : 5 Train toss : 0.3559   Valid toss : 0.2975	Epoch : 39 Train Loss : 0.1205   Train Accuracy : 0.9559 Walid Loss : 0.3023   Walid Accuracy : 0.9202
Epoch : 6 Train Loss : 0.3492   Valid Loss : 0.2880	Epoch : 40 Train Luss : 0.1221   Train Accuracy : 0.9926 Valid Luss : 0.2995   Valid Accuracy : 0.9986
Epoch: 7 Train Loss : 0.1550   Valid Loss : 0.2374	Evaluate on the Test set: Test loss :0.2005   Test Accuracy : 0.0205

These two are for the Random Initialization on the sleepEDF and Epilepsy datasets, respectively.

## 7. Enlightening Observations

TS-TCC is highly effective in semi-supervised settings. Even with just 1% of labeled data, TS-TCC outperforms supervised training, showcasing its ability to leverage unlabeled data for improved performance. A comprehensive performance comparison of the proposed TS-TCC framework against several baseline methods, including Random Initialization, Supervised training, SSL-ECG, CPC, and SimCLR. It is observed that TS-TCC outperforms all the baseline methods, showcasing the effectiveness of the proposed framework for self-supervised timeseries representation learning. Temporal features are more important than general features in time-series data, as CPC, which focuses on temporal information, performs better than Sim-CLR, which was originally designed for images.



## 8. Ways to the future

To advance this field further and build on the findings of this study, there are several directions and opportunities for future research:

1. Enhanced Architectures: Future work can explore more advanced neural network architectures tailored to time-series data. Investigating modifications or novel architectures for temporal feature extraction and contrastive learning can lead to even

more powerful self-supervised models

2. Generalization to Diverse Domains: While the study evaluates the performance of TS-TCC in various domains, future research can extend this evaluation to an even broader range of applications, including finance, healthcare, and environmental monitoring. Generalization across diverse domains is an important step for real-world applicability.

Future research should continue to push the boundaries of self-supervised learning for time-series data, making it more accessible, interpretable, and applicable to a wide array of real-world problems.

## 9. Final Words

In conclusion, the TS-TCC framework proves to be a powerful and efficient approach for self-supervised time-series representation learning. It is effective in various training settings, including linear evaluation, semi-supervised training, and transfer learning. We have developed a framework useful not just in sleep data classification, but highly important to all of artificial intelligence, increasing accuracy working with time-series data. The research findings highlight the importance of leveraging both temporal features and discriminative learning in the context of time-series data, offering a promising approach for real-world applications involving time-series data.

## Acknowledgements

- 1. https://paperswithcode.com/task/automatic-sleep-stage-classification
- 2. https://www.ijcai.org/proceedings/2021/0324.pdf Special thanks to Prof. Kshitij Jadhav and teaching assistant Raja Kumar for assisting us throughout the project.