Project Laboratory Report

Department of Telecommunications and Artifical Intelligence

Author:	**András Palásti**
Neptun:	**IDNGIS**
Specialization:	**Data Science and Artifical Intelligence**
E-mail address:	**andras.palasti@edu.bme.hu**
Supervisor:	**Gábor Szűcs, PhD**
E-mail addres:	**szucs@tmit.bme.hu**
Co-Supervisor:	**Marcell Németh**
E-mail addres:	**nemethm@tmit.bme.hu**

Classification and segmentation of time series data with supervised and unsupervised AI

Time-series analysis is a powerful tool for analyzing and forecasting multivariate time series data. TimeDRL [1], a deep learning model for time series analysis, is based on a dual-embedding architecture that simultaneously learns two distinct representation types: timestamp-level embeddings capturing temporal characteristics at individual points, and instance-level embeddings encoding holistic patterns across entire series. This report investigates the performance and representational properties of its learned representations. Specifically, we investigate the utility of timestamp- and instances-level representations with simple classifiers, the scenarios when positive-only training is beneficial, the redundancy between timestamps and instances, and the localization of instance information within timestamp-level embeddings.

2024/25 II. semester

1 Theory and previous works

1.1 Introduction

This report investigates TimeDRL [1], a versatile deep learning model designed for time-series analysis, including both classification and forecasting tasks. We begin with a theoretical overview of the TimeDRL model (Section 1.2), which forms the basis for our research questions and project objectives, presented in Section 1.3. Section 2 then outlines our implementation approach and provides detailed answers to the formulated questions.

1.2 Theoretical summary

The TimeDRL (Time-series Disentangled Representation Learning) framework [1], proposed by researchers in the field of time series analysis, represents a significant advancement in extracting meaningful representations from multivariate time series data. This section provides a comprehensive examination of their approach, which addresses the inherent challenges of analyzing complex, high-dimensional temporal datasets.

At its core, TimeDRL introduces a novel dual-embedding architecture that simultaneously learns two distinct representation types: timestamp-level embeddings capturing temporal characteristics at individual points, and instance-level embeddings encoding holistic patterns across entire series. This disentangled representation paradigm facilitates more nuanced analytical capabilities applicable to diverse downstream tasks including classification and forecasting.

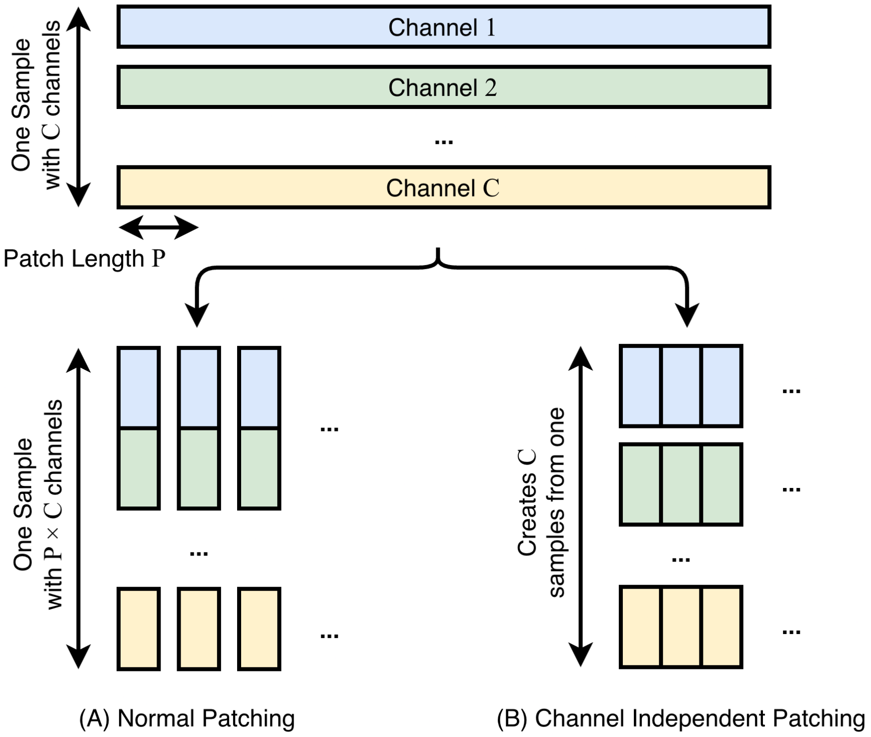

Figure 1: Illustration of patching mechanisms employed in TimeDRL: (A) Conventional patching applied uniformly across input data, and (B) Channel-independent patching with separate extraction protocols for individual channel dimensions.

The authors formulate the input multivariate time series as a matrix $\mathbf{X} \in \mathbb{R}^{C \times T}$, where T represents the temporal dimension and C denotes the number of channels or variables. A key methodological contribution lies in their strategic approach to data segmentation, employing multiple patching techniques as illustrated in Figure 1:

- **Timestep Aggregation**: Building upon principles established in PatchTST [2], this approach aggregates consecutive timesteps into unified token representations, effectively reducing the contextual window requirements for transformer processing. The authors demonstrate this technique's particular efficacy for classification tasks.

- **Channel-Independent Processing**: This methodology conceptualizes multivariate series as collections of univariate sequences processed within a unified model architecture. In contrast to channel-mixing approaches that directly model cross-variable interactions, this technique demonstrates superior performance in forecasting applications through effective isolation of individual channel patterns.

The authors note that for certain datasets, overlapping patch implementations enhance representation continuity—particularly valuable for phenomena characterized by smooth temporal transitions.

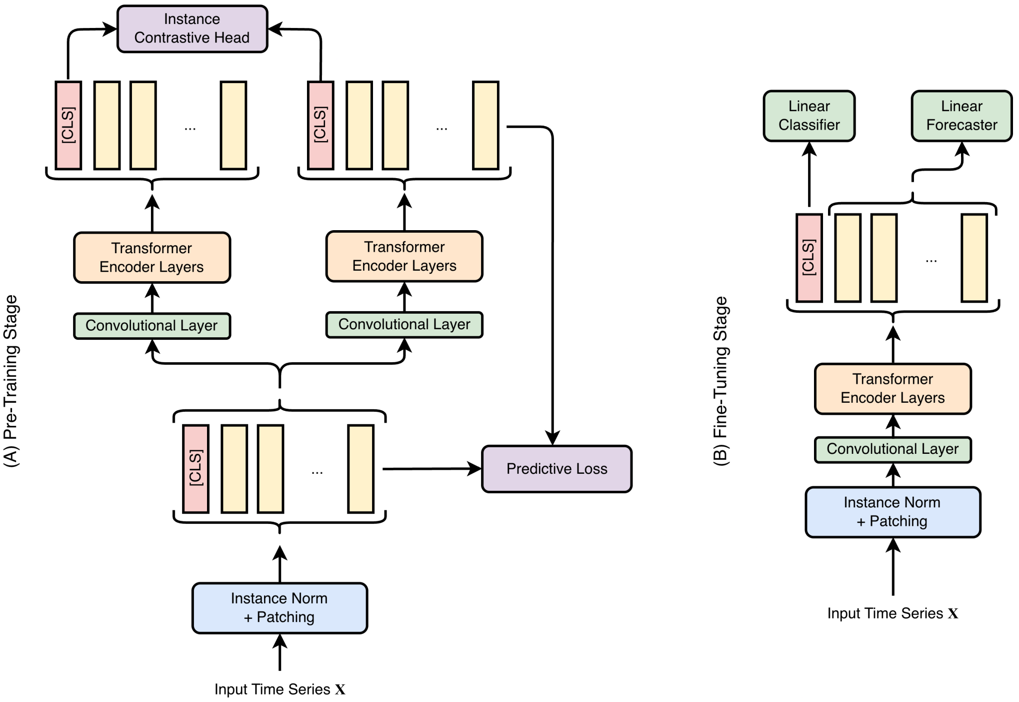

Figure 2: The dual-stage transformer-based TimeDRL framework for time series modeling: (A) Pre-training utilizes self-supervised contrastive learning with normalized and patched time series inputs processed through convolutional and transformer encoder layers. (B) Fine-tuning adapts the pre-trained representations for downstream classification or forecasting tasks via specialized linear output layers while maintaining the core processing pipeline.

The architectural implementation, as depicted in Figure 2, employs a transformer-based encoding mechanism through the following sequential process:

- **Initial Embedding Generation**: Each patch undergoes transformation into an embedding vector e_i, yielding a sequence of N patch embeddings. Following established practices in representation learning [9], a learnable classification token $e_{\text{CLS}} \in \mathbb{R}^d$ is prepended to this sequence, creating the input array $[e_{\text{CLS}}, e_1, \ldots, e_N]$. Positional encodings are subsequently incorporated to preserve temporal ordering information.

- **Transformer-Based Processing**: This sequence is processed through a multi-layered transformer encoder comprising self-attention mechanisms and feed-forward networks. The self-attention operations enable comprehensive interaction between patch embeddings, effectively capturing temporal dependencies across the series. Concurrently, the classification token attends to all patch embeddings, aggregating global information. This process yields refined representation vectors $[h_{\text{CLS}}, h_1, \ldots, h_N]$, where each $h_i \in \mathbb{R}^d$.

- **Global Representation Extraction**: The output corresponding to the classification token, h_{CLS}, functions as the instance-level embedding, encapsulating holistic characteristics of the entire time series.

- **Local Representation Extraction**: The remaining outputs $[h_1, \ldots, h_N]$ constitute the timestamp-level embeddings, with each h_i corresponding to a specific temporal segment. These localized representations enable reconstruction of the original patch data.

TimeDRL's training utilizes a multi-objective optimization approach to learn both timestamp-level and instance-level representations effectively. The model processes each patched input $\mathbf{X}_{\text{patched}}$ twice, generating $[h_{\text{CLS}}^{(1)}, h_1^{(1)}, \ldots, h_N^{(1)}]$ and $[h_{\text{CLS}}^{(2)}, h_1^{(2)}, \ldots, h_N^{(2)}]$ as dual views of the input time-series data. These representations are optimized using a combined loss function balancing two objectives: a timestamp-predictive loss and an instance-contrastive loss.

The timestamp-predictive loss, $\mathcal{L}_P$, uses Mean Squared Error (MSE) to reconstruct the patched time-series data, ensuring accurate timestamp-level embeddings:

$$\mathcal{L}_P = \frac{\text{MSE}(\mathbf{X}_{\text{patched}}, f_\theta([h_1^{(1)}, \ldots, h_N^{(1)})]) + \text{MSE}(\mathbf{X}_{\text{patched}}, f_\theta([h_1^{(2)}, \ldots, h_N^{(2)}]))}{2} \tag{1}$$

where $f_\theta([h_1^{(1)}, \ldots, h_N^{(1)})])$, $f_\theta([h_1^{(2)}, \ldots, h_N^{(2)}])$ are predictions from the timestamp-level embeddings of both views. The instance-contrastive loss, $\mathcal{L}_C$, employs a contrastive approach with cosine similarity, using dropout to introduce variation instead of negative samples:

$$\mathcal{L}_C = \frac{-\text{cosine}(\hat{h}_{\text{CLS}}^{(1)}, \text{stopgrad}(h_{\text{CLS}}^{(2)})) - \text{cosine}(\hat{h}_{\text{CLS}}^{(2)}, \text{stopgrad}(h_{\text{CLS}}^{(1)}))}{2} \tag{2}$$

where $\hat{h}_{\text{CLS}}^{(1)}$ and $\hat{h}_{\text{CLS}}^{(2)}$ are projected instance-level embeddings derived from $h_{\text{CLS}}^{(1)}$ and $h_{\text{CLS}}^{(2)}$. The authors argue that the use of negative samples—commonly obtained through augmentations or other strategies in prior work [7, 6, 8]—introduces inductive biases, and therefore adopt a positive-sample-only approach. The total loss combines both components:

$$\mathcal{L} = \mathcal{L}_P + \lambda \cdot \mathcal{L}_C \tag{3}$$

with hyperparameter λ used to adjust between the two losses. This dual-loss strategy enables TimeDRL to disentangle fine-grained temporal dynamics from holistic instance characteristics, enhancing its effectiveness for multivariate time-series analysis.

1.3 Objectives

Building on its foundational architecture, we undertake a comprehensive experimental investigation to evaluate the performance and representational capabilities of TimeDRL. Our goal is not only to replicate the results reported in the original work but also to extend the analysis by examining how different components of the model behave under varied training regimes and tasks.

To guide our study, we formulate the following key research questions:

- **RQ1**: How effective are the learned instance-level embeddings for downstream classification tasks when using simple classifiers (e.g. k-NN) without fine-tuning with actual class labels?
- **RQ2**: In which scenarios does training a model using only positive samples enhance performance, and in what cases, might this approach be less effective?
- **RQ3**: To what extent do instance-level and timestamp-level embeddings capture redundant information? Is it possible to accurately infer instance-level representations from timestamp-level embeddings?
- **RQ4**: Building on the previous question, if timestamp-level embeddings indeed encode instance-level information, what is the specific mechanism or location within these embeddings where such information is stored or represented?

2 Own work on project

2.1 Experiments

To evaluate the effectiveness of TimeDRL across a diverse set of time-series classification tasks, we conducted a series of controlled experiments using multiple publicly available datasets. This section outlines the experimental protocols, including dataset characteristics, implementation details, and the evaluation metrics used to assess model performance.

2.1.1 Experimental Setup

Dataset	Series Length	# of Channels	# of Classes	Imbalance Ratio	# of Samples Train	Test
HAR	128	9	6	1.444872	5881 (67%)	2947 (33%)
WISDM	256	3	6	8.418033	2617 (76%)	819 (24%)
PenDigits	8	2	10	1.084840	7494 (68%)	3498 (32%)
Epilepsy	178	1	2	4.054945	7360 (76%)	2300 (24%)
FingerMovements	50	28	2	1.012739	316 (76%)	100 (24%)

Table 1: Overview of the datasets used in the experiments, detailing the series length, number of input channels, number of classes, class imbalance ratio, and the number of training and test samples (with their respective proportions).

These datasets in Table 1 represent a variety of time-series classification tasks: HAR and WISDM are focused on human activity recognition; Epilepsy and FingerMovements involve EEG recordings for seizure detection and finger movement recognition, respectively; and PenDigits captures pen trajectory data for handwritten digit recognition. For each dataset, we report an imbalance ratio, defined as the number of samples in the most frequent class divided by the number of samples in the least frequent class. This provides a simple yet informative measure of class distribution skew. An imbalance ratio close to 1 indicates a balanced dataset, whereas higher values reflect significant class imbalance—as seen in WISDM and Epilepsy.

The model architectures and training configurations used for all datasets were kept identical to those reported in the TimeDRL paper[1]. This includes both the model-specific parameters—such as the number of layers and hidden units—and training hyperparameters like learning rate, weight decay, and batch size. Maintaining these settings was essential to ensure a faithful replication of the original results.

2.1.2 Metrics

To ensure consistency with the original TimeDRL study and to facilitate meaningful comparison, we evaluate performance using the same three metrics: accuracy (ACC), macro-averaged F1 score (MF1), and Cohen's Kappa score (Kappa).

- **Accuracy** measures the proportion of correctly classified samples over the total number of samples:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(y_i = \hat{y}_i)$$

  where N is the total number of samples, y_i is the true label, $\hat{y}_i$ is the predicted label, and $\mathbf{1}$ is the indicator function.

- **Macro-averaged F1 score** calculates the F1 score independently for each class and then averages the results:

$$\text{F1}_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C}\frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$$

[1] For the FingerMovements dataset, we identified a likely error in the published parameters, where the weight decay exceeded the learning rate. To address this, we reduced the weight decay by an order of magnitude.

where C is the number of classes, and for each class c:

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}, \quad \text{Recall}_c = \frac{TP_c}{TP_c + FN_c}$$

with TP_c, FP_c, and FN_c denoting the true positives, false positives, and false negatives for class c, respectively.

- **Cohen's Kappa score** [3] assesses the level of agreement between predicted and true labels, while accounting for the agreement that could occur by chance:

  $$\kappa = \frac{p_o - p_e}{1 - p_e}$$

  where the observed agreement p_o and the expected agreement by chance p_e are given by:

  $$p_o = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(y_i = \hat{y}_i), \quad p_e = \sum_{k=1}^{C} p_k^{(y)} \cdot p_k^{(\hat{y})}$$

  Here, $p_k^{(y)}$ and $p_k^{(\hat{y})}$ represent the empirical class distributions of the true and predicted labels, respectively.

These metrics provide a comprehensive evaluation of model performance. Notably, macro F1 and Cohen's Kappa are particularly informative in the presence of class imbalance, which is significant in datasets such as WISDM and Epilepsy. All evaluations are performed using the designated fixed test sets for each dataset.

2.2 Results

We initially sought to replicate the findings presented in the TimeDRL paper concerning classification datasets. Our experimental outcomes closely aligned with the original results[2] for most datasets, with the notable exception of the FingerMovements dataset[3]. For this particular dataset, we observed a significant discrepancy in performance; even with the use of the source code accompanying the TimeDRL paper, we were unable to replicate the accuracy levels reported by the authors.

The detailed outcomes of our replication experiments are presented in Table 2. In addition to replication results, the table presents classification accuracy from models fine-tuned on various types of embeddings—where a pretrained model was first trained on the dataset without class labels, and then a linear classifier was trained on frozen embeddings—along with the performance of a k-Nearest Neighbors (k-NN) classifier [4] trained directly on the frozen instance-level embeddings.

2.2.1 Effectiveness of Instance-Level Embeddings (RQ1)

To address the first research question regarding the effectiveness of the learned instance-level embeddings when used with simple classifiers, we analyze the results shown in Table 2. A k-NN classifier applied to frozen embeddings achieves performance that is remarkably close to that of a fine-tuned linear classifier. This observation indicates that the model's embeddings are inherently expressive and contain sufficient discriminative information to support accurate classification without additional model training.

2.2.2 Scenarios Where Positive-Only Training Helps or Hinders Performance (RQ2)

To understand when training with only positive samples enhances or limits performance, we analyzed k-NN classification accuracy across training epochs (Figure 3) and analyzed embedding structures via t-SNE [5] visualizations after a single training step without contrastive loss (Figure 4). The results in Figure 3 show that the effectiveness of positive-only training is highly dependent on the intrinsic structure of the dataset.

[2] It is worth noting that a different training strategy was used in the original implementation: a linear model was fine-tuned after each pretraining epoch, and the best metric achieved in this manner was reported.

[3] This discrepancy is not attributable to the previously mentioned reduction in weight decay, as we also tested the specified parameter configuration, which yielded even lower performance.

Dataset	Metric	TimeDRL (our implementation)			Original
		Using Instance-level	Using Timestamp-level	Using k-NN	
HAR	ACC	0.8901	0.8744	0.8392	0.8901
	MF1	0.8950	0.8792	0.8417	0.8941
	Kappa	0.8679	0.8492	0.8068	0.8679
WISDM	ACC	0.8787	0.8575	0.8339	0.9145
	MF1	0.7748	0.7472	0.6952	0.8237
	Kappa	0.8273	0.7970	0.7550	0.8785
PenDigits	ACC	0.9727	0.9732	0.9703	0.9800
	MF1	0.9730	0.9735	0.9704	0.9801
	Kappa	0.9697	0.9702	0.9670	0.9778
Epilepsy	ACC	0.9696	0.9778	0.9617	0.9778
	MF1	0.9514	0.9653	0.9373	0.9653
	Kappa	0.9029	0.9306	0.8748	0.9307
FingerMovements	ACC	0.5033	0.4667	0.5500	0.6400
	MF1	0.5012	0.4647	0.5445	0.6377
	Kappa	0.0084	-0.0640	0.1042	0.2826

Table 2: Comparison of Classification Performance Metrics for TimeDRL: This table presents the performance metrics across the five datasets using our implementation of TimeDRL with instance-level embeddings, timestamp-level embeddings, and a k-NN classifier, alongside the original TimeDRL results reported in the paper.

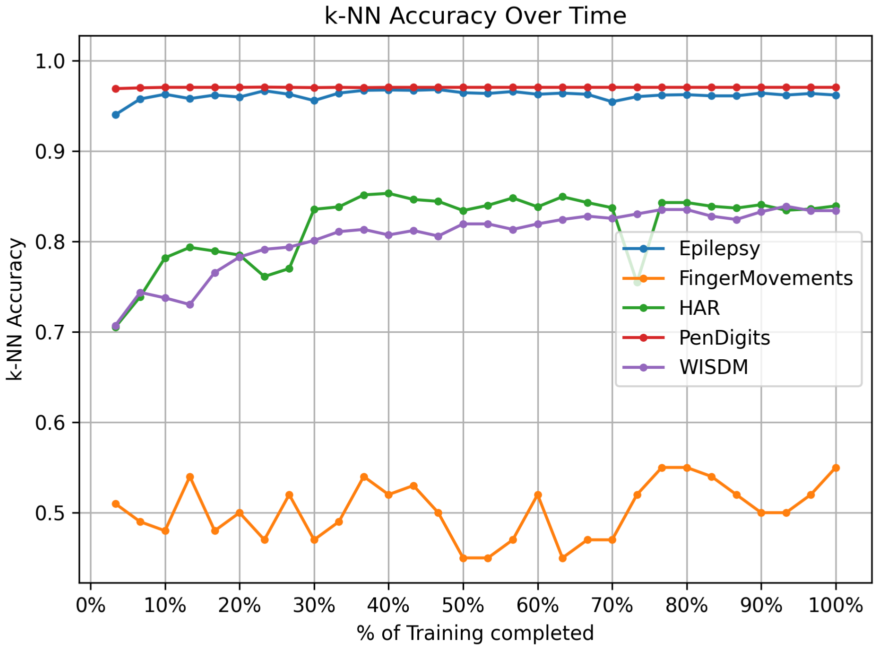

Figure 3: Accuracy of k-NN classification on test datasets across percentage of completed training epochs

Positive-only training proves most effective when there is natural class separability in the data. For instance, PenDigits and Epilepsy showed high and stable k-NN accuracy right from the start of training, indicating that class-specific patterns are easily captured even without negative examples. This is reinforced by the t-SNE visualizations in Figure 4, where these datasets exhibit well-formed, clearly separated clusters even after a single training step without contrastive loss.

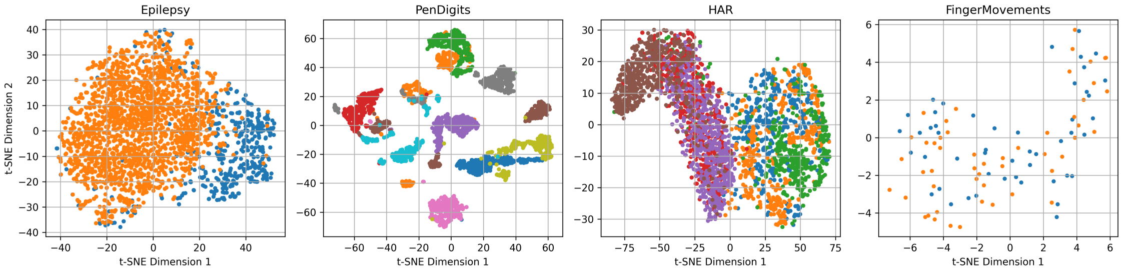

Figure 4: t-SNE visualization of the embedding space after a single training step without contrastive loss.

HAR and WISDM show gradual accuracy improvements, eventually stabilizing at approximately 84% and 83%, respectively. The t-SNE plot for HAR indicates more scattered embeddings, yet the results imply that positive contrastive loss training still supports meaningful representation learning in such cases.

In contrast, FingerMovements demonstrates the weakest performance among all datasets, with k-NN accuracy remaining consistently low and exhibiting high variability throughout training. As shown in Figure 4, the t-SNE projection reveals no real clusters and no real class separation. This suggests that when class differences are subtle or heavily overlapping, positive-only training struggles to learn informative embeddings. In summary, the effectiveness of positive-only training depends on the inherent structure of the data—specifically, the degree of intra-class cohesion and inter-class separability—highlighting that this approach works best when classes are naturally distinct and internally consistent.

2.2.3 Redundancy Between Instance-Level and Timestamp-Level Embeddings (RQ3)

In relation to the third research question, we investigate the extent to which instance-level and timestamp-level embeddings capture overlapping information. Returning to Table 2, we observe that classification performance based on instance-level and timestamp-level embeddings is generally similar, suggesting that both embedding types encode overlapping information. The largest observed gap is a 5% increase in accuracy on the FingerMovements dataset when using instance-level embeddings. These observations point to a substantial degree of shared informational content between the two embedding types.

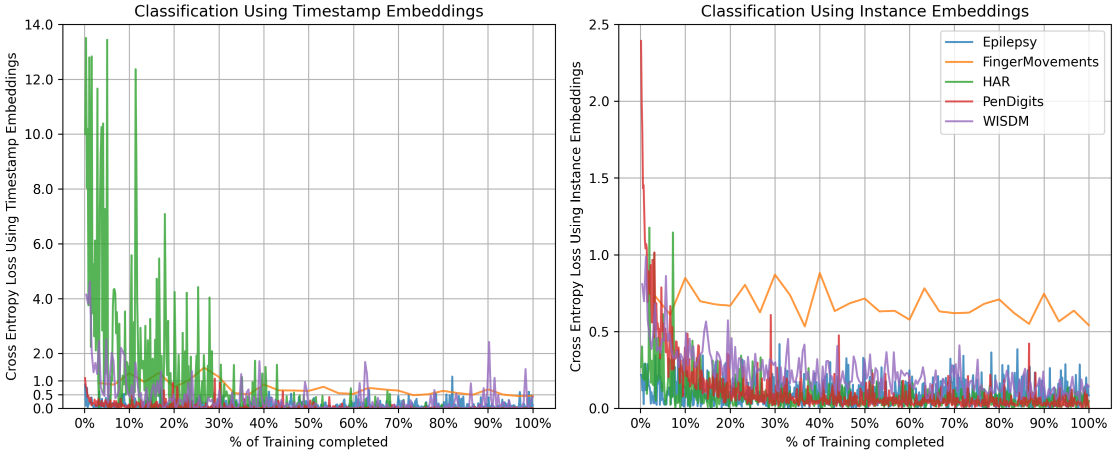

Figure 5: Comparison of Training Loss Trajectories for Instance-Level and Timestamp-Level Embeddings

Despite similar classification performance, indicating overlap in the information captured by both embedding types, their training dynamics differ significantly. Figure 5 visually captures this distinction by plotting the training loss trajectories for both instance-level and timestamp-level embeddings throughout training. The plot illustrates that instance-level embeddings achieve faster, smoother loss reduction, reflecting more stable and efficient optimization. In contrast, timestamp-level embeddings show slower convergence and greater variability.

This demonstrates that while both embeddings ultimately achieve similar classification performance, the instance-level embeddings offer substantial advantages in terms of training stability and efficiency, emphasizing their potential for practical applications where training dynamics are critical.

2.2.4 Localization of Instance Information Within Timestamp Embeddings (RQ4)

Building on the previous point, the fourth research question seeks to understand where within the timestamp-level embeddings the instance-level information is encoded. To this end, we trained a linear model to reconstruct the instance-level embedding from the individual timestep embeddings, optimizing with a mean squared error loss. We evaluated the quality of reconstruction using cosine similarity, while systematically varying the proportion of randomly masked timestep embeddings to assess robustness.

The results shown in Figure 6 demonstrate that reconstruction quality remains largely stable (>0.95) at low missing rates (0%-50%) across all datasets, with a gradual decline as the missing rate increases. Notably, FingerMovements experiences the most significant drop, falling to 0.70 at a 90% missing rate, while other datasets maintain a cosine similarity above 0.83 even when 90% of the timestep-level embeddings are missing.

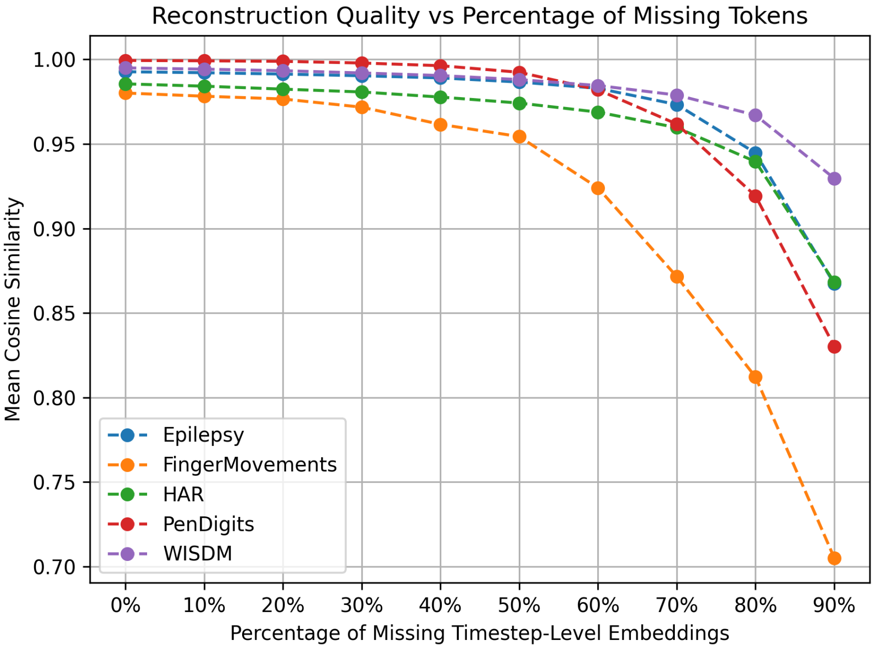

Figure 6: Average cosine similarity between original and reconstructed instance embeddings versus the percentage of missing timestep-level embeddings across datasets.

Dataset	Reconstructed Instance Embedding (0% Missing)			Reconstructed Instance Embedding (50% Missing)		
	ACC	MF1	Kappa	ACC	MF1	Kappa
HAR	0.8718	0.8772	0.8460	0.8604	0.8658	0.8323
WISDM	0.8718	0.7609	0.8171	0.8592	0.7521	0.7997
PenDigits	0.9723	0.9724	0.9692	0.9508	0.9516	0.9454
Epilepsy	0.9687	0.9502	0.9004	0.9678	0.9488	0.8977
FingerMovements	0.4900	0.4890	-0.01782	0.5033	0.5024	0.0085

Table 3: Classification performance using reconstructed instance-level embeddings under two conditions: input reconstruction without and with 50% of timestep embeddings missing.

Although a high reconstruction quality (e.g., 0.98 cosine similarity) indicates strong alignment between the original and reconstructed embeddings, it remains essential to assess its impact on downstream performance. Table 3 provides this evaluation by reporting classification accuracy when the classifier—introduced during fine-tuning—is applied to instance embeddings reconstructed from timestep-level representations.

When comparing the results using the reconstructed instance-level embeddings with those presented in Table 2, we observe similar performance despite the classifier having never encountered the vectors to which it was applied. This highlights the fact that instance-level representations can effectively be derived from timestep-level embeddings.

Additionally, it is noteworthy that comparable performance can be achieved even when the instance embeddings are constructed from only half of the timestep embeddings. This suggests that the instance-level embedding does not rely on specific locations within the timestep embeddings, but rather integrates information from across the entire sequence of timestep embeddings.

2.3 Summary

This study builds upon the TimeDRL framework to evaluate its effectiveness across various time-series classification tasks and to investigate the representational properties of its learned embeddings. We formulated four key research questions addressing (1) the utility of instance-level embeddings with simple classifiers, (2) the scenarios where training with only positive samples enhances or limits performance, (3) redundancy between instance- and timestamp-level embeddings, and (4) the localization of instance information within timestamp embeddings.

Our replication of TimeDRL's classification results showed close alignment with the original work on most datasets, except for FingerMovements, where significant performance discrepancies persisted. We demonstrated that frozen instance-level embeddings support strong performance with simple classifiers such as k-NN, confirming their expressiveness without the need for fine-tuning. Additionally, our experiments with positive-only training showed that embedding quality improves progressively during training, and distinct class structures can emerge even without negative samples. However, the effectiveness of this approach varies significantly by dataset, performing best when classes exhibit strong intra-class cohesion and clear inter-class separability, as observed in datasets like PenDigits and Epilepsy, but struggling with datasets like FingerMovements where class differences are subtle or overlapping.

Investigating redundancy, we found that timestamp-level embeddings contain largely overlapping information with instance-level embeddings, though the latter offer modest accuracy gains. Reconstruction experiments showed that instance-level embeddings can be reliably inferred from timestamp-level embeddings—even with up to 50% of the inputs missing—indicating a distributed and robust encoding of instance-level information across the full sequence of timestep representations.

3 References

[1] Chang, C., Chan, C.T., Wang, W.Y., Peng, W.C. and Chen, T.F., 2024, May. *TimeDRL: Disentangled Representation Learning for Multivariate Time-Series.* In 2024 IEEE 40th International Conference on Data Engineering (ICDE) (pp. 625-638). IEEE.

[2] Nie, Y., Nguyen, N.H., Sinthong, P. and Kalagnanam, J., 2022. *A time series is worth 64 words: Long-term forecasting with transformers.* arXiv preprint arXiv:2211.14730.

[3] Cohen, Jacob. 1960. *A Coefficient of Agreement for Nominal Scales.* Educational and Psychological Measurement 20 (1): 37–46. https://doi.org/10.1177/001316446002000104

[4] T. Cover and P. Hart, 1967, January. *Nearest neighbor pattern classification.* In IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, doi: 10.1109/TIT.1967.1053964.

[5] Hinton, Geoffrey E and Roweis, Sam, 2002. *Stochastic Neighbor Embedding* In MIT Press, vol 15, https://proceedings.neurips.cc/paper_files/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf

[6] S. Tonekaboni, D. Eytan, and A. Goldenberg, 2021. *Unsupervised representation learning for time series with temporal neighborhood coding.* arXiv preprint arXiv:2106.00750.

[7] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, 2019. *Unsupervised scalable representation learning for multivariate time series.* In Advances in neural information processing systems, vol. 32.

[8] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, 2021 *Time-series representation learning via temporal and contextual contrasting.* arXiv preprint arXiv:2106.14112.

[9] J. Devlin, M. Chang, K. Lee and K. Toutanova, 2019 *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv preprint arXiv:1810.04805.