ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10094960

CLASS-INCREMENTAL LEARNING ON MULTIVARIATE TIME SERIES VIA SHAPE-ALIGNED TEMPORAL DISTILLATION

Zhongzheng Qiao[1,2,3,4] *Minghui Hu*[1] *Xudong Jiang* [1]
Ponnuthurai Nagaratnam Suganthan [5,1] *Ramasamy Savitha*[3,4]

[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[2]ERI@N, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore
[3]Institute for Infocomm Research, A*STAR, Singapore [4]CNRS@CREATE LTD, Singapore
[5]KINDI Centre for Computing Research, College of Engineering, Qatar University, Doha, Qatar

ABSTRACT

Class-incremental learning (CIL) on multivariate time series (MTS) is an important yet understudied problem. Based on practical privacy-sensitive circumstances, we propose a novel distillation-based strategy using a single-headed classifier without saving historical samples. We propose to exploit Soft-Dynamic Time Warping (Soft-DTW) for knowledge distillation, which aligns the feature maps along the temporal dimension before calculating the discrepancy. Compared with Euclidean distance, Soft-DTW shows its advantages in overcoming catastrophic forgetting and balancing the stability-plasticity dilemma. We construct two novel MTS-CIL benchmarks for comprehensive experiments. Combined with a prototype augmentation strategy, our framework demonstrates significant superiority over other prominent exemplar-free algorithms.

Index Terms— Continual Learning, Multivariate time series classification, Knowledge Distillation, Dynamic Time Warping

1. INTRODUCTION

Multivariate Time Series (MTS) are temporal sequences with ordered, discrete, and multi-dimensional attributes [1]. Deep learning methods have witnessed great success in MTS classification problems under the assumption that samples are drawn from an independent and identical distribution (i.i.d.). However, this ideal assumption is often violated in real-world MTS data, e.g. due to changes in sensors or the occurrence of new classes. In this case, it is more desirable to continuously update the model to accommodate these changes, while alleviating catastrophic forgetting (CF) of previously learned knowledge [2].

Incremental learning (IL), also known as continual learning (CL), aims at alleviating CF in neural networks. Existing methods are mainly based on memory replay, adaptation of network architecture or regularization of parameters/representations, or a combination of these strategies [3]. It must be noted that replay-based methods require a subset of samples to be saved in a memory buffer. This may violate data privacy, especially in applications involving MTS data such as healthcare or manufacturing. Furthermore, most architecture-based methods require the task id of samples during inference and hence are not task-agnostic. These methods are inapplicable for privacy-sensitive circumstances or task-agnostic scenarios, e.g. *class-incremental learning* (CIL) scenario.

A representative solution is to adopt Knowledge Distillation (KD) [4] for regularization. Concretely, by imitating the pseudo targets (e.g. feature map) generated by a saved teacher, i.e. the last state of the model, historical knowledge can be distilled and preserved in the current student model. The imitation is conducted by minimizing a discrepancy metric, which mostly is Euclidean distance in methods for image data [5, 6]. However, it is not suitable for MTS-CIL where the feature maps are temporal sequences. Euclidean distance assumes the ith timestep in one sequence is aligned with the ith timestep in the other and penalizes any discrepancy along the temporal axis (e.g. shift or dilation). That may impose a large loss for similar feature maps with minor temporal shifts (see the right part of Figure 1) and lead to the over-rigidity problem or stability-plasticity dilemma.

To alleviate the dilemma, we propose to exploit a shape-aligned discrepancy metric, namely Soft-DTW [7], for knowledge distillation. It leads to a differentiable loss function that computes the discrepancy of two sequences after matching their shapes along the temporal dimension. By distilling the feature maps with Soft-DTW, the knowledge of the feature extractor can be preserved in a more flexible manner, leading to a better balance between stability and plasticity. Together with a global KD term [8] and a prototype augment (protoAug) strategy [9], we construct a task-agnostic CIL framework for MTS without saving any historical samples. We name it as DT^2W, which represents Knowledge **D**itllation along **T**emporal dimension with soft-**DTW**.

Compared with image data, the problem of CIL for MTS is quite under-explored. The proposed methods mainly focus on RNN [10, 11, 12] or violate the exemplar-free setting [13]. Furthermore, most benchmarks are either simulated from image data or get limited to short sequence lengths [14], making them far from practical requirements. In this work, we find two real-world MTS datasets with balanced long-term sequences and construct two standard MTS-CIL benchmarks. Comprehensive experiments are conducted with CNN. To the best of our knowledge, our work is the first attempt to exploit Soft-DTW for KD. Our framework demonstrates superiority over its Euclidean-variant and other non-exemplar state-of-the-arts.

2. APPROACH

2.1. Problem Definition

In CIL, a model f parameterized by θ is trained to learn a task sequence $\mathcal{T} = [T^1, ..., T^M]$ in a sequential manner, where each task $T^t = (D^t, C^t)$ is represented by a class set $C^t = \{c_1^t, \ldots, c_{K_t}^t\}$ and training data $D^t = \{(\mathbf{x}_i^t, y_i^t) \,|y_i^t \in C^t\}_i^{N_t}$. Each sample $\mathbf{x}_i \in \mathbb{R}^{k\times l}$ is a MTS with k variables and a fixed sequence length l. Non-overlapping classes in different tasks are assumed, i.e. $C^i \cap C^j = \varnothing$

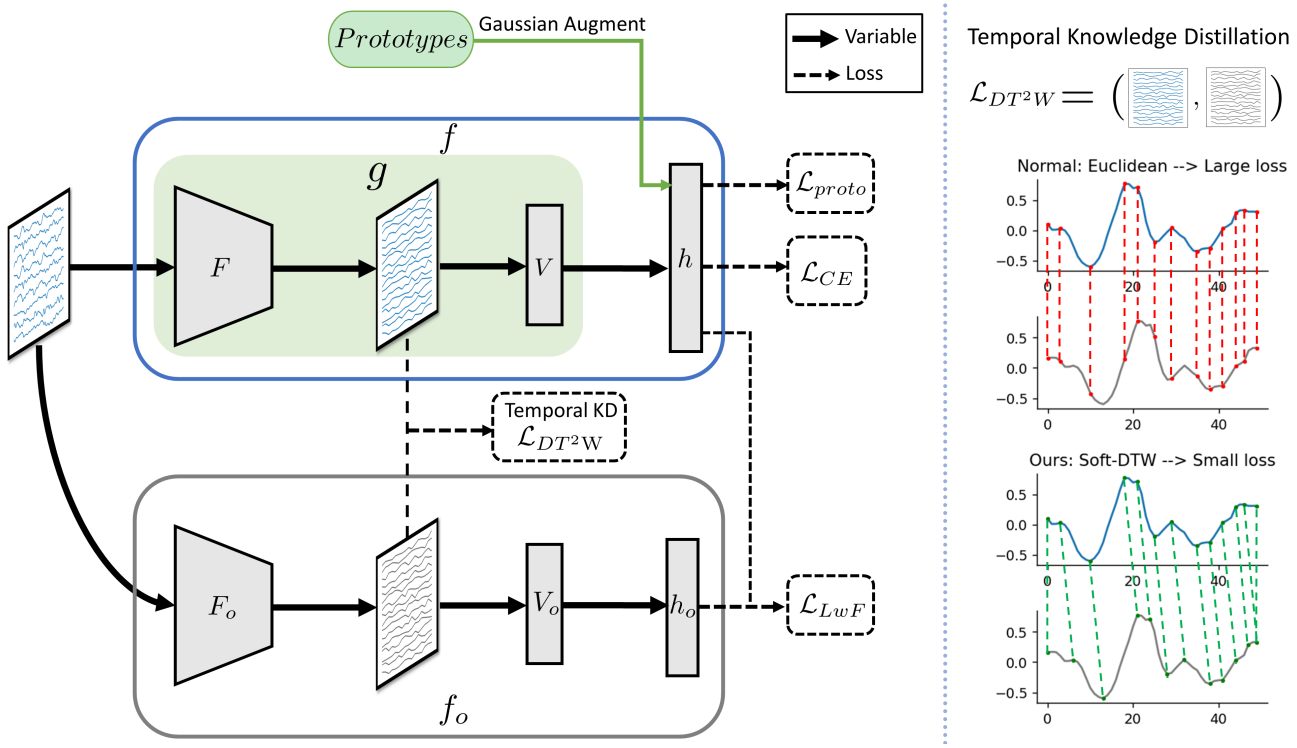


Fig. 1: Illustration of our DT^2W framework (left), where F is a 1D-CNN and V is a Global Average Pooling layer. Using Soft-DTW for knowledge distillation (right) imposes less penalty for feature maps with temporal shifts but in a similar shape.

if $i \neq j$.

In general, model $f = h \circ g$ is composed of a encoder g and a classification head h. The encoder g can be further decomposed as $V \circ F$, where F is a feature extractor to produce a *temporal feature map* and V is a vectorization operation (e.g. global average pooling layer). The feature map is converted into a feature vector by V and sent to h for classification. The head h, which is a fully-connected layer, follows the *single-head* configuration, i.e. dynamically adding a new node for each novel class.

We use a subscript t to denote the state of the model/parameter during task t. After learning tasks $\{1, \ldots, t-1\}$, f^t is continuously trained on D^t to learn task t, without access to the previous/future datasets. Saving historical samples is not allowed to fulfill the exemplar-free setting. After training, f^t with optimal θ^t gets evaluated on the test sets from all the learned tasks $\{1, \ldots, t\}$, without providing the task id of the sample (task agnostic).

2.2. Overall framework

Our DT^2W framework is depicted in Figure 1. The objective is three-fold: (a) Learn the temporal feature maps and classification decision boundary for task t; (b) Preserve knowledge of task 1 to task $t-1$ through Knowledge Distillation, and (c) Calibrate the classification head. These objectives are achieved by minimizing the following loss function:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{DT^2\mathrm{W}} + \lambda_2 \mathcal{L}_{LwF} + \lambda_{proto} \mathcal{L}_{proto} \quad (1)$$

where $\mathcal{L}_{CE}$ is the cross-entropy for efficient classification, $\mathcal{L}_{DT^2\mathrm{W}}$ aims at retaining the knowledge of feature extractor F through preserving the feature maps. $\mathcal{L}_{LwF}$ distills the global knowledge of the model f [8]. $\mathcal{L}_{proto}$ is for the calibration and knowledge preservation of the head h [9]. $\lambda_1, \lambda_2, \lambda_{proto}$ are the hyperparameters to control the strength of each penalty.

2.3. Temporal distillation with Soft-DTW

A desirable similarity metric should consider shifts or dilation when calculating the discrepancy. Dynamic Time Warping [15] is such a time-series-specific metric that reflects the discrepancy of two sequences after temporal shape alignment. By solving a constrained dynamic programming problem, it finds the optimal temporal alignment to produce the minimum discrepancy value. However, DTW is non-differential, so we employ a differentiable variant named Soft-DTW [7] in our framework. Soft-DTW considers the soft-minimum of the cost spanned by all possible alignments and can therefore be used as a loss function in the backpropagation pipeline. For the first time, we utilize Soft-DTW as the information preserving penalty for knowledge distillation in the field of incremental learning.

After task $t-1$, a copy of model f^{t-1} that has learned tasks $\{1, \ldots, t-1\}$ is saved as the teacher f_o. When the model is adapted to represent task t, the feature map output by the feature extractor F is trained to imitate the distillation target generated by the teacher's extractor F_o. The knowledge of F is distilled by minimizing the Soft-DTW-based loss as follows:

$$\mathcal{L}_{DT^2\mathrm{W}} = \mathrm{dtw}_\gamma(F(\mathbf{x}), F_o(\mathbf{x})) \quad (2)$$

where $\mathrm{dtw}_\gamma(\cdot,\cdot)$ is the Soft-DTW discrepancy and F_o is the feature extractor of the teacher. We refer readers to [7] for further details. After training task t, the teacher model f_o is also upgraded to f^t.

To achieve a global knowledge distillation of f^t, we include

LwF-MC loss [16], which implements a temperature-scaled cross-entropy to distill the output probabilities on the old classes. We denote $\mathbf{y}_o = f_o(x)$ as teacher's output (target of KD), $\hat{\mathbf{y}}_o$ as the vector of probabilities on old classes in the student's output, $\hat{\mathbf{y}}$, $\hat{\mathbf{y}} = f(x)$, the loss is shown as

$$\mathcal{L}_{LwF} = \mathcal{L}_{CE}(\mathbf{y}'_o, \hat{\mathbf{y}}'_o) = -\sum_{i}^{|C_o|} y'^{(i)}_o \log \hat{y}'^{(i)}_o \tag{3}$$

where $|C_o|$ is the number of classes in f_o. $y'^{(i)}_o = \frac{(y^{(i)}_o)^{1/\tau}}{\sum_j (y^{(j)}_o)^{1/\tau}}$ and $\hat{y}'^{(i)}_o = \frac{(\hat{y}^{(i)}_o)^{1/\tau}}{\sum_j (\hat{y}^{(j)}_o)^{1/\tau}}$ are normalized recorded and current probabilities $y^{(i)}_o$, $\hat{y}^{(i)}_o$, respectively. Temperature τ is set to 2 to emphasize the information of smaller logits values.

2.4. Knowledge retention and calibration for classifier

Another challenge in CIL is that the single-headed classifier is biased toward new classes, that is, the magnitudes of weights of new classes are larger than those of old classes [17, 18]. Eq 3 alone is not sufficient to solve the bias problem or to retain the head's knowledge. To address the issues, we adopt a prototype argumentation strategy proposed by [9], which saves a single prototype for each learned class. Concretely, after finishing each task, the prototype of new class c is computed as $\mu_c = \frac{1}{N_c}\sum_i g(\mathbf{x}_i | y_i = c)$, which equals the mean of feature vectors of that class. Then the prototypes are saved into a set $\mathcal{P}$. When a new task is trained, pseudo features of the encountered classes are generated by augmenting the prototypes in $\mathcal{P}$ with Gaussian noise:

$$\mathbf{f}_c = \mu_c + r * \mathbf{z}, \quad \mu_c \in \mathcal{P} \tag{4}$$

where $\mathbf{f}_c$ is the pseudo feature for class c, $\mathbf{z} \sim \mathcal{N}(0, 1)$ is the Gaussian noise. r is the radius for augmentation and is calculated as the average variance of the class features in the first task [9]. Pseudo features are sent into classifier, updating the model with an additional mini-batch. The loss for protoAug is:

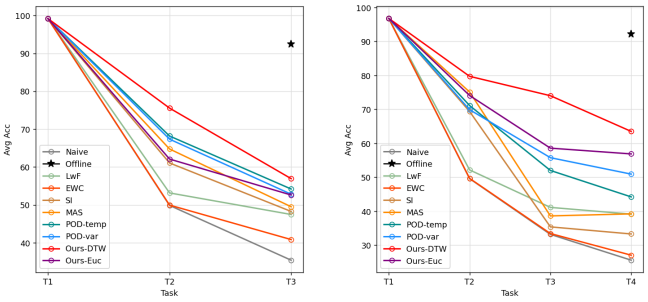
$$\mathcal{L}_{proto} = \mathcal{L}_{CE}(h(\mathbf{f}_c), \mathbf{y}_c) \tag{5}$$

where $\mathbf{y}_c$ is the one-hot vector of class c. For hyperparameter λ_{proto}, we set its value adaptively [17] in a saturated form: $\lambda_{proto} = (1 - \frac{\pi}{e^t})\lambda^{max}_{proto}$, where π is data specific to control the changing speed. $\lambda^{max}_{proto} > 1$ is the saturated value after training a number of tasks.

3. EXPERIMENTS SETUP

We develop two new benchmarks for MTS-CIL problems with public domain real-world datasets. *HAR* [19] is a representative human activity recognition dataset, which contains 9-dimensional inertial signals of smartphones when 6 different daily activities are performed. *Uwave* [20] collects the 3-axis coordinates data from accelerometers while generating 8 simple gestures. For each individual dataset, the sequence length of the sample is fixed, with 128 and 315 timesteps, respectively. To form CIL benchmarks, HAR and Uwave are divided into 3 and 4 tasks, respectively, with 2 different classes for each task. Like [16], classes in the dataset are arranged in a fixed random order for each run.

We experiment with a 1D-CNN-based architecture. Feature extractor F is a stack of three CNN blocks, each of which consists of 2 convolution layers with filter size of 3 and stride of 1. Each convolution layer is followed by a BatchNorm layer. A max-pooling layer (stride=2, kernel size=2) is added after the second BatchNorm layer in the block. The channels of convolutional layers are 64, 64, 128, 128, 256, and 128, respectively. A Global Average Pooling (GAP) layer [21] is used as the vectorization function, which pools the feature map along the temporal dimension and generates a feature vector of dimensionality 128.



(a) HAR (b) UWave

Fig. 2: Evolution of average accuracy.

Following the recommendations from [22], we use an SGD optimizer (learning rate of 1e-4, momentum of 0.8) with a small batch size of 16. All experiments are run 5 times with different random seeds. Each task is trained with 50 epochs with early stopping. The agent-specific hyperparameters are selected with grid search [17].

We compare our method with two baselines and multiple exemplar-free IL algorithms [23]. The baselines are (1). *Naive*: the lower bound without using any IL method and (2). *Offline*: the upper bound training with all the samples in the entire data stream. Compared methods include *EWC* [24], *SI* [25], *MAS* [26], *LwF* [8] and two variants of *PODNet* [5], namely *POD-temporal* and *POD-variate*.[1] We also make a comparison with a variant of our method by replacing Soft-DTW with Euclidean distance.

We adopt two standard metrics [18] for evaluation. **Average Accuracy** after learning task i is defined as $\text{ACC}_i = \frac{1}{i}\sum_{j=1}^{i} a_{i,j}$, where $a_{i,j}$ is the accuracy on the test set of task j after mastering task i. **Average Forgetting** reflects the task-level forgetting on the learned tasks, defined as $\text{FGT}_i = \frac{1}{i-1}\sum_{j=1}^{i-1} f_{i,j}$, where $f_{k,j} = \max_{l\in\{1,\ldots,k-1\}}(a_{l,j}) - a_{k,j}, \forall j < k$ represents how much knowledge about task j is forgotten after learning task k. The end values of the metrics with 0.95 confidence interval of the Student's t distribution are reported to summarize an overall performance after learning the entire task sequence.

4. RESULTS AND DISCUSSION

4.1. Evaluation of compared methods

The evolution of average accuracy is depicted in Figure 2. Our DTW-based algorithm beats all the competitors by a large margin in every single step. In general, the algorithms using protoAug (our methods and two PODNet variants) yield better results, showing the broad effectiveness of this augment strategy in the CIL setting. EWC is the worst one among all the compared methods, resulting in performance even close to the lower bound in UWave, which has also

[1] Use a single-headed classifier with protoAug and revise the POD loss by pooling along either the temporal dimension or the variate dimension.

Table 1: Overall performance of different methods

	HAR		UWave	
Method	**ACC(%)**	**FGT(%)**	**ACC(%)**	**FGT(%)**
Naive	35.41(±3.87)	64.23(±3.92)	25.57(±0.53)	72.50(±2.11)
LwF	47.48(±1.08)	51.26(±1.04)	39.13(±3.25)	58.22(±4.22)
EWC	40.86(±8.18)	58.71(±8.13)	27.04(±11.72)	50.02(±13.06)
SI	48.28(±4.7)	50.46(±4.89)	33.29(±4.12)	48.41(±5.87)
MAS	49.52(±3.54)	48.84(±3.64)	39.21(±6.39)	31.06(± 8.24)
POD-temporal	54.2(±3.57)	44.44(±3.59)	44.21(±3.55)	51.25(±4.48)
POD-variate	52.83(±6.69)	46.21(±6.52)	50.92(±4.3)	37.35(±4.16)
Ours-Euclidean	52.62(±4.87)	46.16(±5.05)	56.87(±8.07)	**17.59**(±4.28)
Ours-DTW	**56.94** (±4.96)	**41.63**(±4.69)	**63.50**(±5.34)	21.66(±1.67)

been found in [18, 14]. SI and MAS, show a better performance than EWC, while MAS outperforms SI steadily. As a basic KD-based method, LwF shows a similar performance to importance-based baselines. For the two PODNet variants, POD-variate outperforms the POD-temporal in UWave while POD-temporal works better in HAR, reflecting that there is no universally better pooling strategy for PODNet frameworks in MTS classification problems.

The end average accuracy and average forgetting after learning the final task are summarized in Table 1. Our DTW-based method dominates the end average accuracy, surpassing the best compared methods by 2.74 and 12.58 percent points (p.p.) on HAR and UWave, respectively. The final average forgetting of our DTW-based method is also competitive, respectively decreasing by 2.81 and 9.4 p.p compared to the best existing methods in the two datasets. However, the Euclidean-based variant of our framework shows the least forgetting on UWave, better than our Soft-DTW-based version. To find out the causes, we look into more details of Ours-DTW and Ours-Euclidean on UWave in the next subsection.

4.2. Soft-DTW vs Euclidean Distance

Figure 3 shows the final confusion matrices of Naive, Ours-Euclidean, and Ours-DTW from one run on UWave. Clearly, Naive has a severe bias towards new classes in the classification results. Although Ours-Euclidean alleviates this imbalance, many classes (4 to 7) invoke a high possibility of being misclassified to the previously learned classes. This is consistent with our analysis that using Euclidean distance for temporal knowledge distillation imposes an excessively strong penalty for retaining the old knowledge and tends to overemphasize stability during learning new tasks. In contrast, distillation with Soft-DTW reduces the bias problem without introducing new bias toward old classes (no activations below the diagonal of the confusion matrix). This implies that the shape alignment introduces beneficial plasticity for classification calibration. It is worth noting that the accuracy of Ours-Euclidean on the classes from new tasks

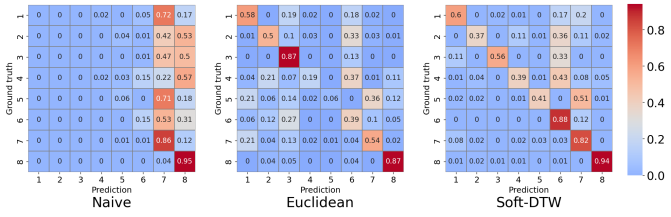


Fig. 3: Confusion matrices of Naive, Ours-Euclidean and Ours-DTW on UWave. The value in each cell shows the classification result, which is represented by the color level of the cell shown in the bar.

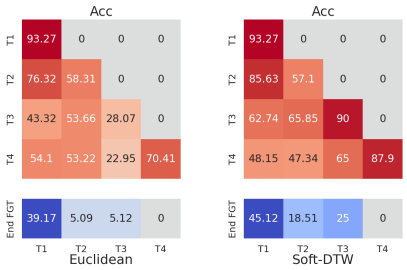


Fig. 4: Accuracy matrices and End Forgetting of Ours-Euclidean and Ours-DTW on UWave.

(4-8) is significantly lower than Ours-DTW. For further illustration of the results, Figure 4 depicts the accuracy matrices $[a_{i,j}]$ and end forgetting $f_{4,j}$ of Ours-Euclidean and Ours-DTW. It can be clearly noticed that the remarkably small forgetting of the Euclidean variant is achieved with the sacrifice of learning new knowledge (tasks 3-4).

4.3. Ablation study

We evaluate the effectiveness of each component in Eq 1. From the results in Table 2, we can observe that: (1). Distillation on temporal feature maps retains more information from the encoder, increasing by 13.76% than only using LwF and protoAug. (2) Introducing LwF term further improves the performance of KD by 7.3% on UWave. (3) Although protoAug only works on the classifier, it plays a vital role in maintaining accuracy and reducing forgetting, with gaps of 19.97 and 30.33, respectively. (4) Combining all the components, the performance reaches a significant higher level compared to using them individually.

Table 2: Ablation study on Uwave. Comparison of the performance of the model when disabling parts of the loss.

Components			UWave	
L^t_{DTW}	L^t_{lwf}	L^t_{proto}	**Acc(%)**	**FGT(%)**
	✓	✓	49.74(±2.33)	44.91(±3.28)
✓		✓	56.2(±5.4)	32.46(±5.62)
✓	✓		43.53(±5.46)	51.99(± 7.27)
✓	✓	✓	**63.50**(±5.34)	**21.66**(±1.67)

5. CONCLUSION

This paper proposes a novel privacy-preserving CIL algorithm for MTS data. By distilling the temporal feature maps with Soft-DTW, knowledge of the encoder is properly retained. Compared with Euclidean distance, using Soft-DTW for KD better balances the accommodation of new knowledge and the preservation of the old one. Our future work will investigate this KD strategy on the architecture with parameterized vectorization function (transformer) and MTS in variable sequence length.

6. ACKNOWLEDGMENT

This research is part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

7. REFERENCES

[1] A. Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 35, no. 2, pp. 401–449, 2021.

[2] Michael McCloskey and Neal J Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, pp. 109–165. Elsevier, 1989.

[3] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[4] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al., "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[5] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *European Conference on Computer Vision*. Springer, 2020, pp. 86–102.

[6] Minsoo Kang, Jaeyoo Park, and Bohyung Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16071–16080.

[7] Marco Cuturi and Mathieu Blondel, "Soft-dtw: a differentiable loss function for time-series," in *International conference on machine learning*. PMLR, 2017, pp. 894–903.

[8] Zhizhong Li and Derek Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[9] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu, "Prototype augmentation and self-supervision for incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5871–5880.

[10] Andrea Cossu, Antonio Carta, and Davide Bacciu, "Continual learning with gated incremental memories for sequential data processing," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[11] Lea Duncker, Laura Driscoll, Krishna V Shenoy, Maneesh Sahani, and David Sussillo, "Organizing recurrent network dynamics by task-computation to enable continual learning," *Advances in neural information processing systems*, vol. 33, pp. 14387–14397, 2020.

[12] Shagun Sodhani, Sarath Chandar, and Yoshua Bengio, "Toward training recurrent neural networks for lifelong learning," *Neural computation*, vol. 32, no. 1, pp. 1–35, 2020.

[13] Dani Kiyasseh, Tingting Zhu, and David A Clifton, "Clops: Continual learning of physiological signals," *arXiv preprint arXiv:2004.09578*, 2020.

[14] Andrea Cossu, Antonio Carta, Vincenzo Lomonaco, and Davide Bacciu, "Continual learning for recurrent neural networks: an empirical evaluation," *Neural Networks*, vol. 143, pp. 607–627, 2021.

[15] Donald J Berndt and James Clifford, "Using dynamic time warping to find patterns in time series.," in *KDD workshop*. Seattle, WA, USA:, 1994, vol. 10, pp. 359–370.

[16] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[17] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin, "Learning a unified classifier incrementally via rebalancing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[18] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.

[19] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Ambient Assisted Living and Home Care*, José Bravo, Ramón Hervás, and Marcela Rodríguez, Eds., Berlin, Heidelberg, 2012, pp. 216–223, Springer Berlin Heidelberg.

[20] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009, PerCom 2009.

[21] Zhiguang Wang, Weizhong Yan, and Tim Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[22] S. Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh, "Understanding the role of training regimes in continual learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7308–7320, 2020.

[23] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido M Van de Ven, et al., "Avalanche: an end-to-end library for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3600–3610.

[24] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[25] Friedemann Zenke, Ben Poole, and Surya Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.

[26] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.