Learning from Stochastic Teacher Representations Using Student-Guided Knowledge Distillation

Muhammad Haseeb Aslam[1] (✉), Clara Martinez[2], Marco Pedersoli[1], Alessandro Lameiras Koerich[1], Ali Etemad[3], and Eric Granger[1]

[1] LIVIA, Dept. of Systems Engineering, ETS Montreal, Canada
muhammad-haseeb.aslam.1@ens.etsmtl.ca
{eric.granger, marco.pedersoli, alessandro.koerich}@etsmtl.ca
[2] CentraleSupélec, Universite Paris Saclay, Paris, France
[3] Aiim Lab, Queen's University, Canada.
ali.etemad@queensu.ca

Abstract. Advances in self-distillation have shown that when knowledge is distilled from a teacher to a student using the same deep learning (DL) model, student performance can surpass the teacher, particularly when the model is over-parameterized and the teacher is trained with early stopping. Alternatively, ensemble learning also improves performance, although training, storing, and deploying multiple DL models becomes impractical as the number of models grows. Even distilling a deep ensemble to a single student model or weight averaging methods first requires training of multiple teacher models and does not fully leverage the inherent stochasticity for generating and distilling diversity in DL models. These constraints are particularly prohibitive in resource-constrained or latency-sensitive applications on, e.g., wearable devices. This paper proposes to train only one model and generate multiple diverse teacher representations using *distillation-time dropout*. However, generating these representations stochastically leads to noisy representations that are misaligned with the learned task. To overcome this problem, a novel stochastic self-distillation (SSD) training strategy is introduced for filtering and weighting teacher representation to distill from task-relevant representations only, using student-guided knowledge distillation. The student representation at each distillation step is used to guide the distillation process. Experimental results[4] on real-world affective computing, wearable/biosignal (UCR Archive), HAR, and image classification datasets show that the proposed SSD method can outperform state-of-the-art methods without increasing the model size at both training and testing time. It incurs negligible computational complexity compared to ensemble learning and weight averaging methods.

Keywords: Deep Learning · Self Distillation · Dropout · Time-Series · Student-Guided Knowledge Distillation

[4] Code and supplementary available at: https://github.com/haseebaslam95/SSD

1 Introduction

Wearable technology has many applications, primarily in healthcare monitoring, such as activity and exercise tracking, sleep analysis, stress detection, and fall detection. It also includes applications like chronic disease management, personalized health insights, and human behavior and physiology research by continuously tracking metrics like heart rate, steps taken, body temperature, and movement patterns over time. Time-series signals such as electrocardiogram (ECG), respiration rate, and other biosignals are often multi-dimensional, noisy, and collected in real time from resource-constrained devices. These signals require efficient processing methods that balance accuracy with computational efficiency. Cumbersome methods for performance boosting are less effective for this application. Knowledge distillation (KD) is typically used for transferring knowledge from a large, well-trained teacher model to a more compact student model for deployment, thereby enhancing the latter's accuracy without incurring significant computational costs [12].

Self-distillation is a specialized case in KD, where the teacher and student have the same DL architecture, and the student typically surpasses the teacher's performance particularly where the model is over-parameterized i.e., has sufficient capacity and the teacher is trained with early-stopping. This increase in performance is typically associated with the fact that, with DL models, the teacher and the student have learned separate discriminative features, and self-distillation implicitly ensembles the two models [1]. Diversity in the feature space is a critical factor that enhances the robustness and accuracy of machine learning models. Diverse representations provide a comprehensive understanding of the input data, mitigating overfitting and improving generalization across various tasks [9] [17].

Approaches for ensemble learning leverage the independent training of multiple diverse models to learn more robust decision boundaries, leading to significant improvements in predictive accuracy. Despite these advantages, deploying deep ensembles introduces substantial computational and storage overhead, as each model in the ensemble requires independent training, parameter storage, and inference pipelines. These constraints are particularly prohibitive in resource-constrained embedded systems as employed in wearable applications.

State-of-the-art (SOTA) approaches [1] that distill diverse ensemble-based representations involve the cumbersome process of training the teacher model multiple times or utilizing complex ensemble learning methods to generate a pool of diverse teacher models for effective knowledge transfer. These methods are computationally intensive and may not fully leverage the potential of stochasticity inherent in DL models for generating diversity.

This paper introduces a KD training strategy called Stochastic Self-Distillation (SSD) to capitalize on *distillation-time* dropout, thereby inducing stochasticity in a single, pre-trained teacher model. SSD generates multiple stochastic feature representations, effectively simulating a diverse ensemble of DL models without requiring extensive teacher re-training. This technique aligns with the principles of Monte Carlo dropout [11]. Moreover, a Student-Guided Knowledge Distilla-

tion (SGKD) is introduced to distill the most relevant knowledge (or filter out noisy representations) to the student model using student-guided attention. This mechanism allows the student to selectively focus on the most informative representations within the teacher's output space, facilitating a more efficient and targeted knowledge transfer. Subsequently, feature-level KD is employed to align the student's feature representations with the filtered and attention-weighted teacher feature representation.

The main contributions of this paper are summarized as follows.
(1) We propose SSD, a novel distillation-time dropout strategy to generate diverse stochastic representations from a single, pre-trained teacher model.
(2) Within SSD, a novel SGKD mechanism enables the student model to selectively distill knowledge from the most informative teacher representations. Feature-based KD is used to align the student's internal feature space with the teacher, promoting a more granular knowledge transfer.
(3) Our extensive experiments on challenging affective computing benchmark datasets (Biovid Pain and StressID), biosignal/wearable datasets (from the UCR Archive), the HAR dataset, and benchmark image classification datasets (CIFAR-10 and CIFAR-100) show that our SSD training strategy allows training models that can achieve SOTA performance while maintaining computational efficiency.

2 Related Work

Knowledge Distillation. Originally introduced by [5] [12], the KD domain has evolved with several refinements in its application and architecture. Romero et al. [29] introduced the concept of distilling from feature representations instead of logits. The idea of transferring the attention maps from the teacher model to the student model was studied by Zagoruyko and Komodakis [41]. Relational KD proposed by Park et al. [27] studied the benefits of utilizing structural information for more fine-grained KD. The KD domain was extended to multi-task, semi-supervised, and unsupervised learning by Lopez et al. [23]. KD has also been studied in multimodal systems, particularly with applications like cross-modal KD [30] privileged KD [3], federated learning [21], are a few examples of the widespread application of KD in real-world systems.
Deep Ensembles and Model Soups. Ensembling methods improve predictive performance, generalization in neural networks, and uncertainty estimation. Deep ensemble is a simple yet effective technique where a simple aggregation of independently trained models harnesses the diversity, leading to better performance than each model. Lakshminarayanan et al. [17] demonstrated the effectiveness of deep ensembles for uncertainty estimation, showing that they outperform many Bayesian approaches in terms of both calibration and robustness. Deep theoretical insights on ensemble diversity were provided by Fort et al. [9]. Moreover, Ovadia et al. [26] highlight the advantages of deep ensembles in handling distributional shifts, reinforcing their utility in real-world scenarios. Despite their advantages, deep ensembles are computationally expensive, requiring the training and storage of multiple models, which motivates research

into alternative methods that capture similar benefits with reduced complexity. More recently, parameter-efficient fine-tuning techniques like low-rank adaptation (LoRA) [13] have enabled efficient fine-tuning of large models. For example, Li et al. [20] introduced Ensembles of Low-Rank Expert Adapters. However, these techniques still require i) careful adaptation of each model, and ii) storing all the models for inference.

Model soups [39], is a technique for improving model generalization by averaging the weights of multiple fine-tuned models. Instead of selecting a single best model, model soups combine the parameters of different models fine-tuned with different hyperparameters, datasets, or random seeds, resulting in a more robust model. Two variations of the model soups were proposed: (i) uniform soup averages the weights of all fine-tuned models equally, and (ii) greedy soup, where models are added iteratively using a greedy approach. Model soup does not increase the model size for inference/deployment, yet it incurs significant additional train-time computational cost by fine-tuning models multiple times.

Self-Distillation. This term has been used in the literature in two different contexts: distilling knowledge from deeper layers in a model to shallower layers of the same model's instance or through the use of an auxiliary network [19,42], and knowledge distilled from a model to another instance of the model with the same architecture [1,10,25]. In this work, 'self-distillation' refers to the latter. Furlanello et al. [10] proposed born-again neural networks, a seminal work exploring KD using the same model for teacher and student, showing that the student can outperform the teacher. Iterative distillation from the trained student, used as a teacher for the subsequent student model, also improved performance. Dong et al. [**?**] showed that early stopping is crucial in harnessing dark knowledge in self-distillation settings. Dark knowledge is the hidden class relationships encoded in the teacher model's soft probability outputs, which provide more information than hard labels. This nuanced information helps the student model learn better generalization and richer representations. A direct correlation between the diversity in the teacher predictions and student performance was studied in depth by Zhang et al. [43]. The authors enhanced the predictive diversity through a novel instance-specific label smoothing.

The concept of self-distillation in a regression setting was first studied by Mohabi et al. [25], in which the authors provided a theoretical analysis of self-distillation where only the soft labels from the teachers were used to train the student. Multi-round self-distillation settings limit the number of basic functions that must be learned. Borup et al. [4] build upon the previous analysis by including the weighted-ground truth targets in the self-distillation procedure. They show that for fixed distillation weights, the ground-truth targets lessen the sparsification and regularization effect of the self-distilled solution. Stanton et al. [32] studied the paradigm of KD through the lens of fidelity. Their key takeaway regarding KD and ensembles was that the highest-fidelity student is the best calibrated, even when it is not the most accurate. The closest work to SSD was proposed by Allen-Zhu and Li [1], who explored the concept of self-distillation in conjunction with the *multi-view* structure of the input data. In

this case, the student model was trained on the ground truth labels with additional supervision from the ensemble of multiple teachers' soft labels. Multiple teachers were trained with random seed initialization.

In contrast to these methods, our proposed SSD training strategy obviates the need for such data augmentations or random seed initialization to generate diversity in the teacher space. SSD operates in the feature space, using dropout as a tool to introduce diversity and student-guided attention to distill relevant information for the student. Consequently, SSD requires significantly less complexity for training when compared to traditional ensemble learning and weight-averaging methods, and without increasing model size at deployment time.

3 Proposed Method

Notation: Let $\mathcal{T}$ be a teacher model with model parameters $\theta^{\mathcal{T}}$ and $\mathcal{S}$ be a student model with parameters $\theta^{\mathcal{S}}$. For given inputs $\mathcal{X} = [x_1, x_2, \ldots, x_m]$, we obtain the feature vectors $f^{\mathcal{T}} = \mathcal{T}(\mathcal{X}; \theta^{\mathcal{T}}) \in \mathbb{R}^{d \times m}$ and $f^{\mathcal{S}} = \mathcal{S}(\mathcal{X}; \theta^{\mathcal{S}}) \in \mathbb{R}^{d \times m}$, where d is the dimension of the feature vector and m is the number of input samples. Let $\mathcal{F}^{\mathcal{T}}(x) = [f_1^{\mathcal{T}}(x), f_2^{\mathcal{T}}(x), \ldots, f_n^{\mathcal{T}}(x)]$ represents the multiple stochastic teacher representations generated through n forward passes through $\mathcal{T}(\mathcal{X}; \theta^{\mathcal{T}})$ for the same input sample $x \in \mathcal{X}$.
Problem Definition: Given a trained teacher model $\mathcal{T}$, the challenge is to effectively transfer its knowledge to a student model $\mathcal{S}$ such that its generalization performance is maximized. Standard self-distillation techniques often treat the teacher's outputs as deterministic, failing to exploit the inherent stochasticity that can provide richer and more diverse information. On the other hand, stochastically obtaining the teacher representations introduces diversity but at the cost of generating noisy representations. The problem, therefore, is to design a KD framework that leverages the variability in the teacher's representations, generated through stochastic mechanisms like dropout, while ensuring that the student learns task-relevant information in a computationally efficient manner. The main aim of our paper is to filter out the noisy representations from $\mathcal{F}^{\mathcal{T}}(x)$ and obtain weighted teacher representation $\hat{f}^{\mathcal{T}}(x)$ to selectively distill from the relevant teacher representations.

3.1 Stochastic Self-Distillation

The proposed SSD training strategy generates multiple diverse representations per sample and uses the current student representation as a reference to rank, select, and weigh (using student-guided attention) the teacher representations before distilling. Further, SSD enforces the attention weights of the meaningful representations to be spread out through temperature scaling, this implicitly models the feature ensemble to harness diversity. The student representation guides each distillation step because it is initialized with the same weights as the main trained teacher. Fig. 1 illustrates the proposed SSD method. The remainder of this section provides details on the SSD training strategy.

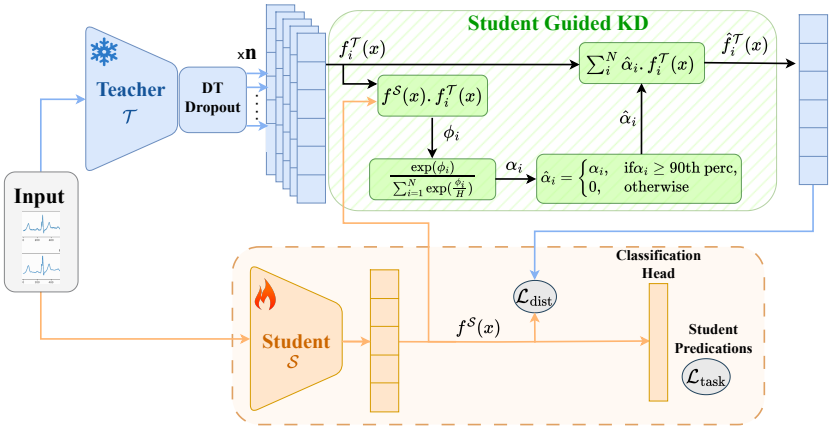


Fig. 1: Illustration of the proposed SSD training strategy. The teacher $\mathcal{T}$ is trained, and its parameters are frozen except for the dropout layers in the student training stage. In SGKD, for each input $x \in \mathcal{X}$, n stochastic teacher representations $f_i^{\mathcal{T}}(x)$, $i = 1, 2, ..., n$, and one student representation is obtained $f^{\mathcal{S}}$, which are fed to the SSD module that outputs $\hat{f}^{\mathcal{T}}(x)$. Feature-based KD is then applied on $\hat{f}^{\mathcal{T}}(x)$ and $f^{\mathcal{S}}(x)$, with addition of $\mathcal{L}_{\text{task}}$, and the student model parameters $\theta_{\mathcal{S}}$ are updated. The part inside the dashed orange block is kept at inference.

Teacher Training. The first step is to train the teacher model and get trained teacher parameters $\theta^{\mathcal{T}'}$. This step is needed for two purposes: i) because the trained teacher model is used to generate the stochastic teacher representations $\mathcal{F}^{\mathcal{T}}(x)$, and ii) because these weights are also used to initialize the student model parameters in the student training step.

Student Parameters Initialization. After the teacher model is trained, the student model parameters are initialized with the trained teacher weights. This initialization is also crucial in the proposed training strategy. The proposed method relies on student guidance to obtain the attended teacher feature vector $\hat{f}^{\mathcal{T}}(x)$. The initialization of the student parameters with trained teacher weights lets the student serve as authority to weigh the teacher representations. As mentioned earlier, the stochastic nature of $\theta^{\mathcal{T}'}$, necessitates that some of the generated representations would be misaligned with the learned task-specific representation and hence would act as noise for the student model. This phenomenon is studied in detail in Section 4.3

3.2 Student-Guided Knowledge Distillation

Traditionally, in KD methods, the teacher representation(s) are informative and serve as additional supervision for the student model. However, if the teacher representations are generated through a stochastic process, they are not aligned with the learned class boundary and can be noise for the student model. There-

fore, we use the current student representation as authority to select representations aligned with the learned class boundary. The student representation at each distillation step can be used as the guiding mechanism because the student model is initialized with the learned weights from the teacher network. This initialization allows the student to make use of its intermediate representation $f^{\mathcal{S}}(x)$ as an anchor to rank the teacher representation $f_i^{\mathcal{T}}(x) \in \mathcal{F}^{\mathcal{T}}$.

To guide the distillation process using the current student representation, we first calculate the dot product (ϕ_i) between the current student representation $f^{\mathcal{S}}(x)$ and each teacher representation $f_i^{\mathcal{T}}(x) \in \mathcal{F}^{\mathcal{T}}(x)$ and compute attention weights using:

$$\alpha_i = \frac{\exp(\phi_i/h)}{\sum_{i=1}^{N} \exp(\phi_i/h)} \tag{1}$$

where N is the number of teacher representations and h is a regularization factor used to smooth the attention weights to ensure that the attended teacher feature vector $\hat{f}^{\mathcal{T}}(x)$ is not heavily influenced only by a single teacher representation. This regularization step is crucial since it dictates the attention weights for teacher representations. Since the way these teacher representations are ranked is through dot product between current student representation $f^{\mathcal{S}}(x)$ and each of the stochastic teacher representation $f_i^{\mathcal{T}}(x)$, the value ϕ_i naturally would be the highest for the teacher representation that is the most similar to the student representation. This renders the entire framework ineffective because $\hat{f}^{\mathcal{T}}(x)$ becomes overly similar to $f^{\mathcal{S}}(x)$.

The SSD method relies on selecting teacher representations that would mimic an ensemble of independently trained teacher models. In other words, it masks out the representations that are too different from the current student representations. A direct way of selecting such representations would be to use the top-k strategy at each distillation step. Although this strategy can work, it relies on k a hyperparameter that is agnostic to the distribution of $f_i^{\mathcal{T}}(x) \in \mathcal{F}^{\mathcal{T}}$ at each distillation step. To avoid a manual selection of meaningful representations using a top-k strategy, α_i is masked for all indices falling outside of the ϵ-th percentile, denoted by $\hat{\alpha}_i$.

$$\hat{\alpha}_i = \begin{cases} \alpha_i, & \text{if } \alpha_i \geq \epsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where $\epsilon \in [0, 100]$ is the threshold value for masking. Section 4.3 provides a more detailed discussion. After obtaining the regularized attention weights $\hat{\alpha}_i$, the original teacher feature representations $f_i^{\mathcal{T}}(x)$ are weighed using $\hat{\alpha}_i$ to obtain the attended teacher feature vector $\hat{f}^{\mathcal{T}}(x)$ as:

$$\hat{f}^{\mathcal{T}}(x) = \sum_{i=1}^{N} \hat{\alpha}_i \cdot f_i^{\mathcal{T}}(x) \tag{3}$$

The attended feature vector $\hat{f}^{\mathcal{T}}(x)$ is used to distill the information using the mean squared error loss:

$$\mathcal{L}_{\text{dist}} = \frac{1}{d}\sum_{j=1}^{d}\left(f^{\mathcal{S}}(x_j) - \hat{f}^{\mathcal{T}}(x_j)\right)^2 \tag{4}$$

The total loss for the student is shown in Eq. 5,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda\mathcal{L}_{\text{dist}} \tag{5}$$

where λ is the weighting parameter for the distillation loss. The method also allows for the additional constraint for logit-level distillation and can be added to $\mathcal{L}_{\text{total}}$. For the unsupervised/contrastive loss-based methods, we use the intermediate teacher representations to obtain the $\hat{f}^{\mathcal{T}}(x)$. In case of augmented views of the input sample, we average the distillation loss from both views. The pseudo-code for the SSD training procedure is shown in Algorithm 1.

Algorithm 1 - SSD Training Procedure.

Require: Teacher model $\mathcal{T}$ with parameters $\theta^{\mathcal{T}}$, student model $\mathcal{S}$ with parameters $\theta^{\mathcal{S}}$, input samples $\mathcal{X} = [x_1, x_2, \ldots, x_m]$, weighting parameter λ.
Ensure: Trained student model $\mathcal{S}$.
Extract teacher feature vectors $f^{\mathcal{T}}{=}\mathcal{T}(\mathcal{X};\theta^{\mathcal{T}}) \in \mathbb{R}^{d\times m}$.
Extract student feature vectors $f^{\mathcal{S}}{=}\mathcal{S}(\mathcal{X};\theta^{\mathcal{S}}) \in \mathbb{R}^{d\times m}$.
for each input sample $x \in \mathcal{X}$ **do**
    Generate n stochastic teacher representations $\mathcal{F}_i^{\mathcal{T}}(x) = [f_1^{\mathcal{T}}(x), \ldots, f_n^{\mathcal{T}}(x)]$.
    Compute **dot products** $\phi_i = f^{\mathcal{S}}(x) \cdot f_i^{\mathcal{T}}(x)$ for $i = 1, 2, \ldots, n$.
    Compute **attention weights**: α_i for $i = 1, 2, \ldots, n$ using Eq. (1)
    Mask α_i for all indices outside the ϵ-th percentile using Eq. (2)
    Compute **attended teacher feature vector**: $\hat{f}^{\mathcal{T}}(x)$ using Eq. (3)
    Compute distillation loss $\mathcal{L}_{\text{dist}}$ using Eq. (4)
end for
Compute total loss $\mathcal{L}_{\text{total}}$ using Eq. (5)
Update parameters $\theta^{\mathcal{S}}$ using backpropagation

4 Results and Discussion

4.1 Experiment Setup

SSD is validated on: (i) real-world affective computing datasets: the Biovid Heat Pain Database [35] and StressID dataset [6], (ii) wearable and biosignal datasets from the UCR Archive [8], (iii) Human Activity Recognition (HAR) dataset set from the UCI Archive [2], and (iv) on benchmark image classification datasets. Appendix A.1 provides details on these datasets, while Appendix A.3 provides implementation details and results on the CIFAR-10 and CIFAR-100 datasets.

Biovid Heat Pain Database. For Biovid, we use the EDA modality and LOSO cross-validation. The proposed method is tested with a SOTA method on the dataset using the Pain Attention Net [24], which is a transformer-based physiological signal classification network comprised of a multi-scale convolutional network, as SE residual network, and a transformer encoder block. The batch size used for teacher training was 128, and the network was optimized using the Adam optimizer with a learning rate of 0.001. The network was trained over 100 epochs with early stopping. The total number of folds was 87, corresponding to the total number of subjects. For student training, we keep the same setting as the teacher. We manually activate dropout layers with a p-value of 0.2 while keeping the teacher model in inference mode. The total number of repetitions for generating diverse teacher representations was 30.
StressID Dataset. For the StressID dataset, use the EDA and RR modalities and apply feature concatenation to fuse the backbone representations. The EDA backbone was Pain Attention Net [24], and the RR backbone was a 1D CNN with three 1D conv layers with 16, 32, and 64 channels, respectively, with a kernel size of 5, and stride equal to 1, followed by three batch normalization layers. Following the original dataset authors [6], we apply an $80-20$ split for the train and test set and further divide the train data and keep 20% of that for model selection. The batch size used for both teacher and student training was 128, with the learning rate of 0.001 using the Adam optimizer. The total number of repetitions was 30. The dropout layer was activated before the feature concatenation module with a p-value of 0.2.
UCR Archive. For the datasets in the UCR Archive, the proposed method was applied to two SOTA techniques – TS2Vec and SoftCLT – for unsupervised time-series representation learning. We follow the same experimental methodology proposed in TS2Vec [40] and SoftCLT [18]. For the TS2Vec method, the number of stochastic teacher representations was 15, with a teacher dropout rate p of 0.2, and the student dropout rate was set to 0.1. The value of H was 5. For loss weighting, λ was set to 0.2. For softCLT, all the parameters were kept the same as those for TS2Vec except for the value of H, which was set to 15.
Computing infrastructure. All experiments were performed on the NVIDIA A100-SXM4-40GB GPUs with the ϵ value of 90.

4.2 Comparison Against State-of-the-Art Methods

Affective Computing Datasets. Table 1 reports the results of SSD and SOTA methods on Biovid. SSD improves accuracy by 2.5% over the selected baseline (PAN without SSD) and 1.7% over the current SOTA. Specifically, accuracy on Biovid increases from 84.59% (using the EDA modality without SSD) to 86.90% with the proposed SSD method.

Table 2 compares the performance of the proposed method with SOTA on the StressID dataset. We achieve 0.74 ± 0.02 for the F1-score and 0.74 ± 0.03 accuracy without applying SSD. The proposed method improves 3% for the F1-score and 4% in accuracy for the binary classification task over the SOTA. This increase in predictive performance shows that the proposed method can perform well

Table 1: Accuracy of SSD against state-of-the-art methods on Biovid data. Baseline results were obtained using the network architecture without applying the proposed SSD method (without distillation).

Method		Modality	CV Scheme	Accuracy
Werner et al. [38]	ICPR 2014	Physio + Vision	5-fold	0.8060
Werner et al. [37]	IEEE TAC 2016	Video	LOSO	0.7240
Kachele et al. [16]	IEEE IJSTSP 2016	EDA, ECG, EMG	LOSO	0.8273
Lopez et al. [23]	ACII 2017	EDA, ECG	10-fold	0.8275
Lopez et al. [22]	EMBC 2018	EDA	LOSO	0.7421
Thiam et al. [33]	Sensors 2019	EDA	LOSO	0.8457
Wang et al. [36]	EMBC 2020	EDA, ECG, EMG	LOSO	0.8330
Pouromran et al. [28]	PLoSONE 2021	EDA	LOSO	0.8330
Thiam et al [34]	Frontiers 2021	EDA, ECG, EMG	LOSO	0.8425
Shi et al [31]	ICOST 2022	EDA	LOSO	0.8523
Ji et al [14]	ACM SAC 2023	EDA	LOSO	0.8040
Jiang et al [15]	ESWA 2024	EDA	LOSO	0.8458
Baseline (w/o SSD)	–	EDA	LOSO	0.8459
SSD (ours)	–	**EDA**	**LOSO**	**0.8690**

on real-world time-series dataset tasks by effectively harnessing task-relevant diversity from stochastic teacher representations.

Table 2: Performance of the SSD against state-of-the-art methods on the StressID dataset. (MM: Multimodal, NR: Not Reported.)

Method	Modality	2-class problem		3-class problem	
		F1-score	Accuracy	F1-score	Accuracy
HC + RF	Physio	0.73 ± 0.02	0.72 ± 0.03	0.55 ± 0.04	0.56 ± 0.03
HC + SVM	Physio	0.71 ± 0.02	0.71 ± 0.02	0.59 ± 0.04	0.59 ± 0.03
HC + MLP	Physio	0.70 ± 0.03	0.70 ± 0.03	0.54 ± 0.04	0.53 ± 0.04
AUs + kNN	Vision	0.70 ± 0.04	0.69 ± 0.04	0.54 ± 0.05	0.53 ± 0.05
AUs + SVM	Vision	0.69 ± 0.04	0.69 ± 0.04	0.55 ± 0.05	0.54 ± 0.04
AUs + MLP	Vision	0.70 ± 0.03	0.70 ± 0.03	0.55 ± 0.03	0.55 ± 0.03
HC + kNN	Audio	0.67 ± 0.06	0.60 ± 0.05	0.53 ± 0.04	0.52 ± 0.04
HC + SVM	Audio	0.61 ± 0.06	0.54 ± 0.03	0.53 ± 0.08	0.48 ± 0.04
wav2vec 2.0	Audio	0.70 ± 0.02	0.66 ± 0.03	0.56 ± 0.04	0.52 ± 0.04
Mordacq et al.	MM	0.69	0.76	NR	NR
Baseline (w/o SSD)	Physio	0.74 ± 0.02	0.74 ± 0.03	0.61 ± 0.01	0.59 ± 0.01
SSD (ours)	**Physio**	**0.77 ± 0.03**	**0.77 ± 0.03**	**0.63 ± 0.02**	**0.60 ± 0.02**

Time-Series Datasets (UCR Archive). Table 3 shows the performance of SSD against the TS2Vec baseline on 12 wearable/biosignal datasets from the UCR Archive. The student model achieves an average accuracy score of 0.8441.

Table 3: Accuracy of SSD applied to TS2Vec and SoftCLT baselines for the wearable biosignal datasets in the UCR Archive.

Dataset	TS2Vec [40] AAAI '22	**TS2Vec + SSD (Ours)**	SoftCLT [18] ICLR '24	**SoftCLT + SSD (Ours)**
ECG200	0.9000	**0.9100**	0.8800	**0.9300**
ECG5000	0.9348	**0.9411**	0.9400	**0.9413**
TwoLeadECG	0.9789	**0.9877**	0.9762	**0.9798**
NIFetalECGThorax1	0.9277	**0.9318**	0.9201	**0.9394**
NIFetalECGThorax2	**0.9389**	0.9343	0.9435	**0.9480**
Chinatown	**0.9737**	0.9708	**0.9737**	0.9708
UWaveGestureLibraryX	0.7995	**0.8079**	0.8001	**0.8143**
UWaveGestureLibraryY	0.7152	**0.7317**	0.7169	**0.7266**
UWaveGestureLibraryZ	0.7624	**0.7660**	0.7674	**0.7682**
MedicalImages	**0.8092**	0.8078	**0.8171**	0.7710
DodgerLoopDay	0.5125	**0.5250**	0.5500	0.5500
DodgerLoopGame	0.7826	**0.8405**	0.8260	**0.8695**
Total	0.8350	**0.8441**	0.8426	**0.8508**

Comparison with Traditional Ensembles and Model Soups Table 4 compares the performance of SSD against traditional ensembles and weight-averaging methods. For simplicity in experimentation and to make sure the results are not biased by the internal mechanisms of architecture, this comparison is performed on bare-bones 1D CNN architecture, which serves as the teacher network for SSD and is also used in the ensembles as well as weight-averaging results.

Table 4: Accuracy of the SSD against state-of-the-art methods with a 1D CNN and traditional ensembles on the HAR dataset.

Model	**Accuracy**
1D CNN (Baseline)	0.9002
Ensemble Majority Vote (25 Models)	0.9135
Ensemble Average (25 Models)	0.9128
Model Soup (10 Models)	0.9101
Model Soup (25 Models)	0.9183
SSD (Student)	0.9182

SSD aims to minimize the space and computational complexity both during training and testing. Fig. 2 compares the performance gain in terms of model size at inference. We compare traditional ensembles (majority voting and averaging), stochastic weight averaging, uniform soup (uniform weight averaging), and greedy soup (a greedy approach for weight averaging). The marker size in Fig. 2 denotes the model size at inference; since traditional ensembles require storing all of the trained models for inference, it becomes impractical to deploy for inference on wearable devices. It can be observed from Fig. 2 that both ma-

jority vote and averaging-based ensembles increase the model performance, but the model size proportionally increases; essentially, for an ensemble model with 25 models, it would become **25**× the size of the baseline model. On the other hand, model soups do not increase the model size at inference but are still computationally expensive in terms of train-time FLOPs as shown in Appendix B. The total number of FLOPs increases from ≈ **0.87 G-FLOPs** to ≈ **21.8 G-FLOPs**. In contrast, the proposed method achieves comparable performance to the traditional approaches, i.e., 1.8% increase in the accuracy over the baseline, while keeping the model size the same as the baseline model, and the train-time computation is significantly less since SSD requires the model to be trained twice, once in the teacher training process and second for student training.

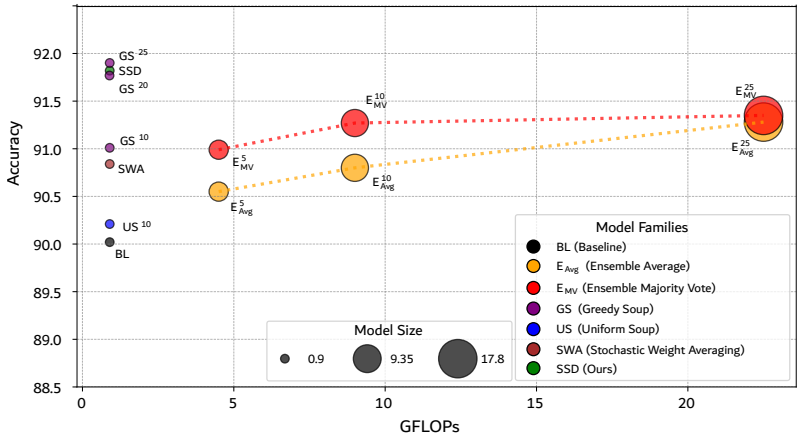


Fig. 2: Comparison of the SSD method with baseline (BL), traditional ensembles with majority vote (E_{MV}) and average (E_{Avg}), uniform soup (US) and greed soups (GS) on HAR dataset in terms of accuracy and model size at inference.

4.3 Ablations

Number of Stochastic Representations. As discussed before, meaningful diversity in the teacher space is crucial for the superior performance of an ensemble and, by extension, also crucial in SSD, since it also implicitly ensembles teacher representations. The number of repetitions dictates how diverse $f_i^{\mathcal{T}}(x)$ is. We evaluated with different settings and reported the results in Table 5. Distill All - the first two rows show the results without applying SSD and learning from all stochastic representations. This could also be seen as an alternative way of teaching students to drop out. Rows 3-6 show results with an increasing number of total repetitions and selected representations. As the total number of representations becomes too large, the performance drops even when applying

Table 5: Comparison of schemes for selecting teacher representations on the StressID dataset. (DS: Dynamic selection. NA: Not available.)

Scheme	**# Representations**	**# Selected**	**F1-Score**	**Accuracy**
Distill All	10	All	0.74 ± 0.02	0.74 ± 0.02
Distill All	30	All	↓ 0.72 ± 0.02	0.72 ± 0.02
top-k	10	3	↑ 0.75 ± 0.02	0.74 ± 0.02
top-k	20	10	↑ 0.76 ± 0.03	0.76 ± 0.02
top-k	30	15	↑ 0.76 ± 0.04	0.76 ± 0.04
top-k	50	30	↓ 0.72 ± 0.02	0.73 ± 0.02
DS	**30**	**NA**	**↑ 0.77 ± 0.03**	**0.77 ± 0.03**
DS	**50**	**NA**	**↑ 0.77 ± 0.03**	**0.77 ± 0.04**

SSD. This leads us to believe that when you select a more significant number of representations, the noisy representations bypass through the filtering mechanism and become part of the distillation process. This problem also indicates a simple top-k selection is not the best strategy in this case. Hence, dynamic selection is applied based on ϵ-th percentile thresholding.

Dropout Rate The extent of diversity in the teacher space is directly related to the distillation-time dropout rate. To study how the probability value of each neuron to be deactivated affects the teacher representation space, we plot and compare the t-SNE plots with different dropout rates. Figs. 3(a)-(d) are plotted with dropout rates 0.1, 0.2, 0.5 and 0.9 respectively. It is observed that for smaller dropout rates (Fig. 3(a)), the teacher can maintain its discriminative ability; however, the three teacher representations $f_1^{\mathcal{T}}(x)$ (triangle), $f_2^{\mathcal{T}}(x)$ (circle), $f_3^{\mathcal{T}}(x)$ (square) mostly overlap each other, effectively meaning there is not enough diversity in the teacher space. For dropout rate 0.2 (Fig. 3(b)), the three representations are diverse while maintaining the original structure, which shows that the teacher space has become diverse while maintaining the discriminative ability. In Fig. 3(c), the three representations are adequately spaced, but the model loses its discriminative ability.

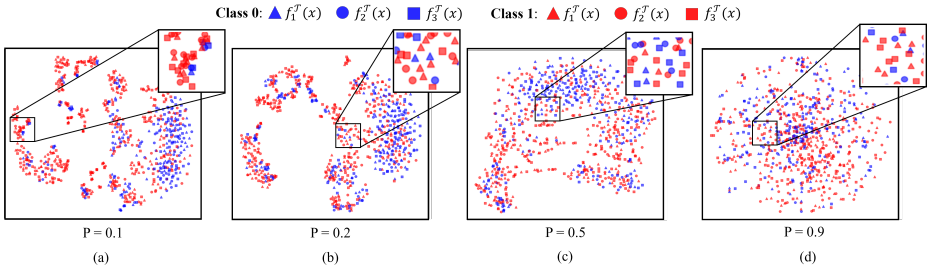


Fig. 3: t-SNE plots of three teacher representations $f_1^{\mathcal{T}}(x)$ [triangle], $f_2^{\mathcal{T}}(x)$ [circle], $f_3^{\mathcal{T}}(x)$ [square] over various dropout rates P showing the effect of the stochastic representations on the learned feature space.

To further analyze the impact dropout rate on both variance and overall performance, we conducted an ablation study on the HAR dataset. Fig. 4 shows the effect of dropout rates on the student performance (dashed/black line) and the variance across teacher representations (colored/solid lines). Variance increases with the teacher dropout rate. For each dropout rate, the highest variance is observed for the lowest number of repetitions. Conversely, the lowest variance for each dropout rate is observed with the highest number of reps. Results suggest that when the number of forward passes is greater, the overall variance decreases. It may also indicate a more accurate approximation of the true variance because it is calculated with more samples. In the latter case, it can be observed that for lower dropout rates, the variances across different number of repetitions are more accurate approximation of the true variance. In terms of performance, the model peaks at a dropout rate of 0.2, and the performance starts deteriorating beyond a dropout rate of 0.5. This phenomenon can be further explained from Fig. 3 where the t-SNE plots show the model starting to lose its discriminative ability at higher dropout rates.

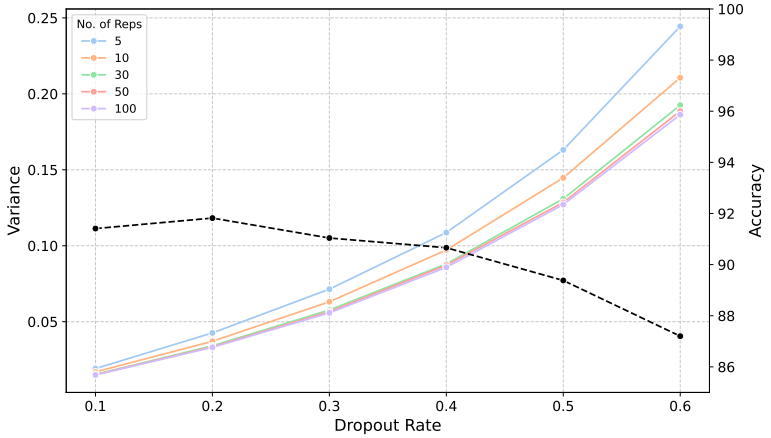
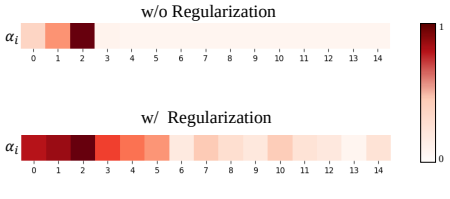
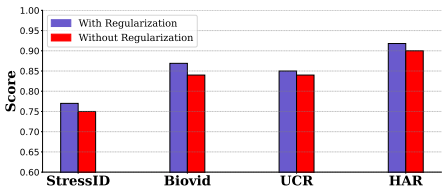


Fig. 4: Effect of various dropout rates on the performance and variance quantification among teacher representations. Accuracy (black/dashed line), and variance for various no. of reps (colored/solid lines) are plotted against dropout rates on the HAR dataset

Effect of Attention Weights Regularization. SSD heavily relies on the diversity in the $\hat{f}^{\mathcal{T}}(x)$, implicitly mimicking the ensemble of task-relevant embeddings from the teacher space. If the attention weights are not regularized, the $\hat{f}^{\mathcal{T}}(x)$ would be highly influenced by one of the teacher embeddings, which is closest to the current $f^{\mathcal{S}}(x)$, essentially rendering the proposed methodology ineffective. Fig. 5a shows the attention weights of an input sample $x \in \mathcal{X}$. It



(a) Visualization of attention weights α_i for two input samples $x_1 \in \mathcal{X}$ and $x_2 \in \mathcal{X}$ with and without regularization.



(b) Performance comparison with and without attention weights α_i regularization on various datasets.

Fig. 5: Effect of attention weights α_i regularization on student performance

can be observed from Fig. 5a that, without regularization, the attention weight is extremely high for α_2. On the other hand, the weights are more spread out with regularization, showing that the student model trained with SSD leverages the diversity in the meaningful teacher representations. Fig. 5b shows the results obtained with and without regularization of α_i. In all instances, regularization improves accuracy.

Effect of Student Parameters Initialization with Teacher Weights. The current student representation guides the distillation process at each step. This section investigates the impact of student parameter initialization with the trained teacher weights $\theta^{\mathcal{T}'}$. Fig. 6 shows the results obtained by the student model with the baseline, random initialization, and student parameter initialization with trained teacher weights. It can be observed from the figure that

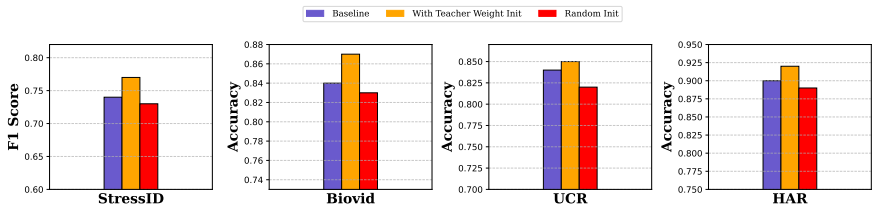


Fig. 6: Performance of SSD with and without student parameters initialization with trained teacher weights $\theta^{\mathcal{T}'}$ on various datasets.

in each instance, the random initialization performs even worse than the baseline. This shows the effectiveness of the proposed method, because if the student model is not initialized with $\theta^{\mathcal{T}'}$, the ϕ_i calculated would be ineffective since the current $f^{\mathcal{S}}(x)$ is not task aligned. Hence adding the additional constraint in the $\mathcal{L}_{total}$ term breaks the performance. See Appendix C.1 for detailed results.

4.4 Discussion

Analysis of Performance on Teacher Network Architecture. SSD can improve performance quite significantly on real-world datasets, particularly with models that have multiple dropout layers built into the architecture. Methods like PAN [24] and the fusion-based architecture used for Biovid and StressID datasets, respectively, have multiple dropout layers throughout backbones and various modules; in contrast, models for HAR and CIFAR datasets are 1D and 2D-Conv ResNet-based architectures with only one dropout layer. One direct correlation between the performance and the capacity to generate stochastic representations can be drawn. The models with a higher number of dropout layers naturally have a greater capacity to generate stochastic representations with more diversity, which allows the proposed method to form a more informative $\hat{f}^{\mathcal{T}}(x)$, consequently leading to a student model with a significant performance boost over the teacher model.

Supervised vs. Unsupervised Setting. The proposed method can enhance performance both in supervised and unsupervised settings. The performance boost observed in the supervised setting is slightly higher than in the unsupervised setting; this could be because the unsupervised contrastive loss-based methods are already equipped with the ability to learn generalized representation. The augmented views of the input data in both TS2Vec and SoftCLT methods lead to more generalized representations.

Why is SSD Different from just Teaching *Dropout* to the Student Model? Intuitively, it seems that the proposed method might be an alternative way of teaching the student model to drop out. However, during both the teacher training step and student training, the standard *training-time* dropout is activated, but the performance boost is not observed. The performance boost can be explained by the filtering of the teacher representations, where *teaching the student to drop out* would be equivalent to distilling from all stochastic representations $f_i^{\mathcal{T}}(x)$ instead of the attended teacher representation $\hat{f}^{\mathcal{T}}(x)$, which is then filtered through the proposed SSD method (see Appendix D for a detailed discussion). This explanation is also supported by the results in Table 5 where we first distill from all $f_i^{\mathcal{T}}(x)$, and the student performance does not improve.

5 Conclusion

In this work, we introduced SSD, a novel approach to enhancing diversity in the teacher space by leveraging the stochastic nature of DL models using *distillation-time* dropout and applying SGKD to learn meaningful representations. It employs the student's current representation as a guide to select meaningful representations, implicitly mimicking an ensemble of task-relevant representations. Extensive experiments on real-world time-series data, complemented by validation on a benchmark vision dataset, show the effectiveness of SSD in improving representation learning. While SSD outperforms SOTA methods, its current evaluation is limited to architectures that already incorporate dropout. Given that SSD operates in the latent space, we hypothesize that the proposed SGKD

framework could be extended to other settings where diversity is introduced through perturbations in the feature space or noise injection. This presents an compelling avenue for future research and a promising alternative to deep ensembles. It also provides an alternative for learning generalized representations for time-series data, paving the way for more efficient and robust representation learning in a wide range of applications.

Acknowledgments. This research endeavor was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Fonds de recherche du Québec – Santé (FRQS), Canada Foundation for Innovation (CFI), and the Digital Research Alliance of Canada.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Allen-Zhu, Z., Li, Y.: Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. ICLR (2020)
2. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: A public domain dataset for human activity recognition using smartphones. In: ESANN (2013)
3. Aslam, M.H., Osama Zeeshan, M., Pedersoli, M., Koerich, A.L., Bacon, S., Granger, E.: Privileged knowledge distillation for dimensional emotion recognition in the wild. In: IEEE/CVF CVPRw (2023)
4. Borup, K., Andersen, L.N.: Even your teacher needs guidance: Ground-truth targets dampen regularization imposed by self-distillation. arXiv 2102.13088 (2021)
5. Bucilua, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD. Association for Computing Machinery (2006)
6. Chaptoukaev, H., Strizhkova, V., et al., M.P.: StressID: a multimodal dataset for stress identification. In: Neural IPS Datasets and Benchmarks Track (2023)
7. Chen, Y., Wang, N., Zhang, Z.: Darkrank: Accelerating deep metric learning via cross sample similarities transfer (2017)
8. Dau et al.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica (2019)
9. Fort, S., Hu, H., Lakshminarayanan, B.: Deep ensembles: A loss landscape perspective. arXiv 1912.02757 (2020)
10. Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. In: ICML (2018)
11. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning (ICML 2016)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv 1503.02531 (2015)
13. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models (2021)
14. Ji, X., Zhao, T., Li, W., Zomaya, A.: Automatic pain assessment with ultra-short electrodermal activity signal. SAC '23, ACM (2023)
15. Jiang, M., Rosio, R., et al., S.S.: Personalized and adaptive neural networks for pain detection from multi-modal physiological features. ESWA **235** (2024)

16. Kächele, M., Thiam, P., Amirian, M.e.a.: Methods for person-centered continuous pain intensity assessment from bio-physiological channels. IEEE JSTSP (2016)
17. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: NeurIPS 2017
18. Lee, S., Park, T., Lee, K.: Soft contrastive learning for time series. In: ICLR 2024
19. Li, S., Lin, M., Wang, Y., Wu, Y., Tian, Y., Shao, L., Ji, R.: Distilling a powerful student model via online knowledge distillation. IEEE TNNLS (2023)
20. Li, Y. et al.: Ensembles of low-rank expert adapters (2025),
21. Lin, T., et al: Ensemble distillation for robust model fusion in fed. learning (2021)
22. Lopez-Martinez, D., Picard, R.: Continuous pain intensity estimation from autonomic signals with recurrent neural networks. In: IEEE EMBC 2018
23. Lopez-Martinez, D., Picard, R.: Multi-task neural networks for personalized pain recognition from physiological signals. In: IEEE ACIIw 2017
24. Lu, Z., Ozek, B., Kamarthi, S.: Transformer encoder with multiscale deep learning for pain classification using physiological signals. Frontiers in Physiology **14** (2023)
25. Mobahi, H., Farajtabar, M., Bartlett, P.L.: Self-distillation amplifies regularization in hilbert space. arXiv 2002.05715 (2020)
26. Ovadia, Y., Fertig, E., et al., J.J.R.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In: NeurIPS 2019
27. Park, W., Kim, D., Lu, Y., Cho: Relational knowledge distillation. In: CVPR '19
28. Pouromran, F., Radhakrishnan, S., Kamarthi, S.: Exploration of physiological sensors, features, and ml models for pain intensity estimation. PlusOne Journal 2021
29. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: ICLR 2015
30. Sarkar, P., Etemad, A.: Xkd: Cross-modal knowledge distillation with domain alignment for video representation learning. AAAI **38**(13), 14875–14885 (Mar 2024)
31. Shi, H., Chikhaoui, B., Wang, S.: Tree-based models for pain detection from biomedical signals. Springer International Publishing, ICOST (2022)
32. Stanton, S., Izmailov, P., Kirichenko, P., Alemi, A.A., Wilson, A.G.: Does knowledge distillation really work? arXiv 2106.05945 (2021)
33. Thiam, P., Bellmann, P., Kestler, H.A., Schwenker, F.: Exploring deep physiological models for nociceptive pain recognition. Sensors **19**, 4503 (2019)
34. Thiam, P., Hihn, H., Braun, D.A., Kestler, H.A., Schwenker, F.: Multi-modal pain intensity assessment based on physiological signals. Frontiers in Physiology 2022
35. Walter, S., Werner, e.a.: The biovid heat pain database: Data for the advancement and systematic validation of an automated pain recognition. In: IEEE ICC 2013
36. Wang, R., Xu, K., Feng, H., Chen, W.: Hybrid rnn-ann based deep physiological network for pain recognition. In: 42nd IEEE EMBC (2020)
37. Werner, P., Al-Hamadi, A., Limbrecht-Ecklundt, K., Walter, S., Gruss, S., Traue, H.C.: Automatic pain assessment with facial activity descriptors. IEEE TAC 2016
38. Werner, P., Al-Hamadi, A., Niese, R., Walter, S., Gruss, S., Traue, H.C.: Automatic pain recognition from video and biomedical signals. In: ICPR 2014
39. Wortsman, M., Ilharco, G., Gadre, S.Y., et al., R.R.: Model soups. arXiv 2203.05482 (2022)
40. Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., Xu, B.: Ts2vec: Towards universal representation of time series. In: AAAI (2021)
41. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of cnns via attention transfer. arXiv 1612.03928 (2017)
42. Zhang, L., Bao, C., Ma, K.: Self-distillation: Towards efficient and compact neural networks. IEEE T-PAMI **44**(8), 4388–4403 (2021)
43. Zhang, Z., Sabuncu, M.R.: Self-distillation as instance-specific label smoothing. arXiv 2006.05065 (2020)