

DAF: Distillation, Augmentation and Filtering based Framework for Efficient Smartphone Human Activity Recognition

Ujjal Kr Dutta
Dolby Laboratories
India
ukdacad@gmail.com

Guan-Ming Su
Dolby Laboratories
USA
guanmingsu@ieee.org

Abstract

Larger, sophisticated sequential models excel in Human Activity Recognition (HAR) using multivariate time-series data but may not suit compute-constrained smartphones due to latency issues. Knowledge distillation offers a solution by training smaller models based on larger teachers, but a single teacher often struggles to perform uniformly well across diverse activity classes. To address this limitation, we propose the Distillation, Augmentation, and Filtering (DAF) framework, leveraging Multiple-Architecture based Multi-Teacher Distillation (MAMTD). This approach identifies the best-performing teacher model for each activity class and uses Contrastive loss-based Distillation to align a smaller student model with the most effective teachers while distancing it from less effective ones. For challenging categories, a peer student model is employed with data augmentation to focus on areas where the first student struggles. Finally, a novel checkpoint ensemble via probability filtering combines the strengths of both student models, achieving a 21.4-24.6% increase in accuracy for certain confusing categories compared to typical distilled networks, while maintaining low latency.

1. Introduction

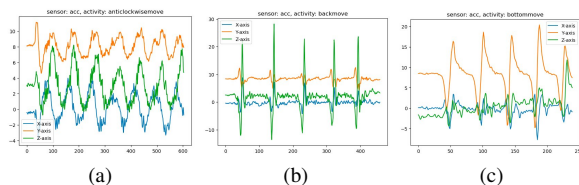


Figure 1. Triaxial (x-,y-,z-) accelerometer readings for moving a handheld smartphone in the following directions: (a) anticlockwise (b) backwards (c) towards ground.

Smartphones are ubiquitous in daily life, serving as primary communication and entertainment tools, as well as playing a significant role in scientific fields like health-

care research [15]. They contain various sensors (e.g., accelerometers, gyroscopes) that collect multivariate time-series signals, enabling the detection of human movements (Fig 1a-1c). Human Activity Recognition (HAR) involves analyzing these signals to identify activities (e.g., rotating the smartphone anticlockwise, swiftly swiping left, towards the ground, etc), often requiring a specified time window and robust logic to handle noisy data. While simple rule-based methods can be insufficient, Deep Learning (DL) models offer a powerful alternative by learning complex patterns in data, making them effective for detecting complex movements with higher Degrees of Freedom (DOF).

Smartphone-based HAR typically includes sensor fusion, data preprocessing, feature extraction, and activity classification. Supervised DL models, such as LSTM, ConvLSTM, and Transformer architectures, are commonly used for HAR due to their ability to capture temporal information. However, complex models can lead to high inference latency, which is mitigated using knowledge distillation techniques [9]. Our work builds upon these concepts to develop efficient HAR frameworks for mobile deployment. We propose a **Multiple-Architecture based Multi-Teacher Distillation (MAMTD)** framework, which we name as **Distillation, Augmentation and Filtering (DAF)**, which leverages multiple teacher models to improve the performance of smaller students suitable for mobile deployment. This framework involves (Fig 3):

1. Training multiple teacher models to allow us for identifying the best-performing architecture for each class, as a single model may not excel uniformly across all classes.
2. **Contrastive loss-based Distillation (CD)** to align the student distribution with the positive teacher model (the best-performing architecture for a class) while distancing it from negative models (all other architectures).
3. Training a peer student with CD and data **augmentation** to address confusing classes where first student struggles.
4. Performing a checkpoint ensemble via a novel **proba-**

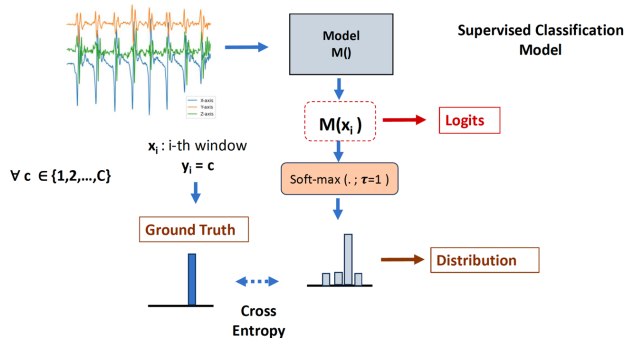


Figure 2. An illustration of training a supervised classification model to predict the class of a time-series window. The logits obtained via a model are converted to a probability distribution via the softmax function, and matched with the ground-truth.

bility filtering strategy to combine the strengths of both student models.

2. Related Work

Several existing methods explore knowledge distillation in various domains, offering insights into how our approach differs. For instance, federated learning setups like [7] focus on privacy preservation using a single teacher, which contrasts with our multi-teacher approach. In cross-modal learning, [13] utilizes two teachers with standard distillation, whereas our method extends beyond two teachers and incorporates contrastive learning. Another approach, [12], reduces multi-axis signals using a single teacher-student pair without contrastive learning or augmentation. Stage-wise distillation methods, such as [3], perform distillation on MobileNet, differing from our multi-architecture approach.

In video action recognition, [19] uses three teachers with similar backbones, unlike our diverse architectures for time-series data. For video human activity recognition, [17] proposes a simple distillation method using a single teacher, which is distinct from our multi-architecture setup. Ensemble learning methods like [6] use ensemble learning with a single teacher, differing from our approach that leverages multiple teachers and contrastive learning. Non-deep learning methods, as discussed in [14] and [22], focus on different aspects of HAR.

Other related works include the application of contrastive learning, such as [16], which applies inter-example contrastive learning on image datasets, whereas we use intra-example contrastive learning among logits. In computer vision, [21] uses knowledge distillation with image augmentation but lacks multiple teachers. Self-supervised learning approaches, like those in [8] and [5], relate to self-supervised methods using positive pairs, unlike our use of InfoNCE-based [4] contrastive loss. Additionally, data-

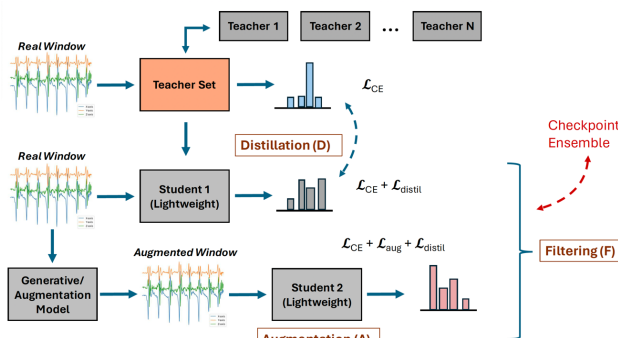


Figure 3. High level illustration of the proposed DAF framework. The approach involves leveraging multiple teacher models to distill a student via Contrastive Learning. To further complement the student in confusing scenarios, a second peer student is trained using data augmentation. The predictions of both students are combined using a filtering technique.

free distillation methods, such as [11], propose a data-free approach with unstable training, unlike our method using training data. Ensemble HAR methods, such as [2], use multiple models but do not address mobile deployment efficiency like our distillation-based approach.

Our framework is novel, combining a multi-architecture, multi-teacher setup with intra-example contrastive learning, focal loss, data augmentation, and probability filtering.

3. Proposed framework for efficient HAR

Training supervised DL models for smartphone-based HAR, involves segmenting multi-variate time-series signals into fixed-length windows and classifying them into one of the activity classes (Figure 2). Each window \mathbf{x}_i has a corresponding class label $y_i = c$, where $c \in \{1, 2, \dots, C\}$ represents one of C activities. The model, represented by $M(\cdot)$, predicts logits $M(\mathbf{x}_i)$, which are converted into a probability distribution using the SoftMax function. The predicted distribution is compared to the ground truth, and Cross-Entropy loss is optimized over several epochs to train the model for accurate recognition.

3.1. Distillation (D) in DAF

To enable low-latency deployment on smartphones, knowledge distillation [9] trains a smaller student model $s(\cdot)$ using a larger teacher model $M(\cdot)$, with a total loss:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{CE}} + (1 - \alpha) \tau^2 \mathcal{L}_{\text{distil}}^{\text{std}} \quad (1)$$

where, \mathcal{L}_{CE} is the cross-entropy loss, which aligns the model's predictions with the ground truth. α is a hyperparameter balancing the two loss terms. τ is the temperature parameter controlling the distribution smoothness. $\mathcal{L}_{\text{distil}}^{\text{std}}$ is the **standard distillation** loss, typically computed using the Kullback-Leibler divergence between the teacher and student distributions ($\mathcal{H}_{\text{distil}}^{\text{std}}$).

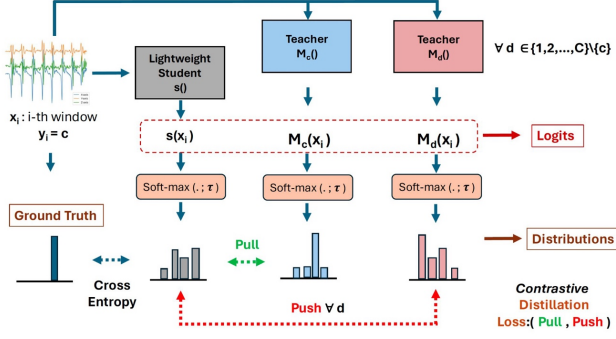


Figure 4. Illustration of the proposed Contrastive Distillation technique. For a training window, the student distribution is pulled closer to the distribution of the teacher which is the best performing for the class label of the window. The student distribution is simultaneously moved away from all other teacher distributions.

The cross-entropy loss \mathcal{L}_{CE} is expressed as:

$$\mathcal{L}_{CE} = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^C \mathbf{y}_i(j) \log\{\sigma(s(\mathbf{x}_i); \tau)(j)\} \quad (2)$$

Here, \mathbf{x}_i is a training example. \mathbf{y}_i is the one-hot encoded ground truth label, s.t., $\mathbf{y}_i = c \in 1, \dots, C$. C is the number of classes. $|\mathcal{B}|$ is the mini-batch size. $\sigma(s(\mathbf{x}_i); \tau)$ is the softmax function applied to the student model's logits with temperature τ .

However, a single teacher often performs inconsistently across diverse activity classes. To address this, we propose creating a *teacher set* (Fig 3) by training multiple large architectures independently and selecting the best-performing teacher for each class. For a multi-teacher setup, we can use the **Class-Specific Activation (CSA) strategy**. Each class c has a best-performing teacher $M_c(\cdot)$, identified from the pre-distillation stage. The CSA distillation loss is:

$$\mathcal{L}_{\text{distil}}^{\text{CSA}} = \mathcal{H}_{\text{distil}}^{\text{std}}(\sigma(M_c(\mathbf{x}_i); \tau), \sigma(s(\mathbf{x}_i); \tau)) \quad (3)$$

However, CSA only considers the positive teacher for each class and does not account for negative teachers. To address this limitation, we propose using **Contrastive Distillation (CD)**. The CD loss is (Fig 4):

$$\mathcal{L}_{\text{distil}}^{\text{CD}} = \frac{1}{|\mathcal{B}|} \sum_{x_i \in \mathcal{B}} \log \left[\frac{\exp(\mathcal{H}_i^c(x_i))}{\sum_{d \neq c} \exp(\mathcal{H}_i^d(x_i))} \right] \quad (4)$$

where

$$\mathcal{H}_i^c(x_i) = \mathcal{H}_{\text{distil}}(\sigma(M_c(\mathbf{x}_i); \tau), \sigma(s(\mathbf{x}_i); \tau)) \quad (5)$$

For each \mathbf{x}_i , CD aligns the student distribution with the positive teacher (M_c , s.t., $\mathbf{y}_i = c$) while pushing away from distributions of negative teachers ($M_d, d \neq c$).

We also incorporate **Focal Loss** to handle class imbalance by modifying $\mathcal{H}_{\text{distil}}$ as:

$$\mathcal{H}_{\text{distil}} = FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (6)$$

Here, α_t is a weighting factor, and γ is a focusing parameter. The total loss is then computed as:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{CE} + (1 - \alpha) \mathcal{L}_{\text{distil}}^{\text{CD}} \quad (7)$$

3.2. Augmentation (A) in DAF

For highly confusing classes (e.g., Fig 6, Fig 8), a single student trained using eq (7) may still struggle. To complement it, we propose training a peer student model with an additional augmented data based loss term \mathcal{L}_{aug} , while modifying eq (7) as:

$$\mathcal{L}_{\text{total}} = \alpha(\mathcal{L}_{CE} + \lambda_{\text{aug}} \mathcal{L}_{\text{aug}}) + (1 - \alpha) \mathcal{L}_{\text{distil}}^{\text{CD}} \quad (8)$$

where, $\lambda_{\text{aug}} > 0$ is the hyperparameter term for \mathcal{L}_{aug} , and

$$\mathcal{L}_{\text{aug}} = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^C \tilde{\mathbf{y}}_i(j) \log\{\sigma(s(\tilde{\mathbf{x}}_i); \tau)(j)\} \quad (9)$$

In our work, we suggest obtaining an *augmented data* instance $\tilde{\mathbf{x}}_i$ using two approaches:

1. **Synthetic Data Augmentation:** We use **Variational Auto-Encoders (VAEs)**, $\mathcal{V}(\cdot)$, to generate synthetic time-series data $\tilde{\mathbf{x}}_i = \mathcal{V}(\mathbf{x}_i)$ for underrepresented classes. The VAE loss function is given by:

$$\mathcal{L}_{\theta, \phi} = -\mathbb{E}_{q(z|x; \phi)}[\log p_\theta(x|z)] + KL(q(z|x; \phi) || p_\theta(z)) \quad (10)$$

VAEs are chosen for their stability and ability to produce realistic samples [10] (see Fig 5 for our IHS dataset).

2. **Mix-Up Augmentation:** This technique mixes both data and labels of confusing classes to increase data diversity and prevent overconfidence in model predictions [20]. For examples \mathbf{x}_i and \mathbf{x}_j with labels \mathbf{y}_i and \mathbf{y}_j , Mix-Up generates a new example $\tilde{\mathbf{x}}$ and label $\tilde{\mathbf{y}}$ as follows:

$$\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1 - \lambda) \mathbf{x}_j \quad (11)$$

$$\tilde{\mathbf{y}} = \lambda \mathbf{y}_i + (1 - \lambda) \mathbf{y}_j \quad (12)$$

where λ is sampled from a **Beta distribution**. This approach helps the model understand the percentage of belongingness of an example towards a certain category, improving predictive performance.

3.3. Filtering (F) in DAF

We then fuse the predictions of both students using a novel **probability filtering** technique. Unlike **average-based fusion**, which computes the final distribution as:

$$p_i^{(\text{final})}(c) = \text{mean}(p_i^{(1)}(c), p_i^{(2)}(c)) \quad (13)$$

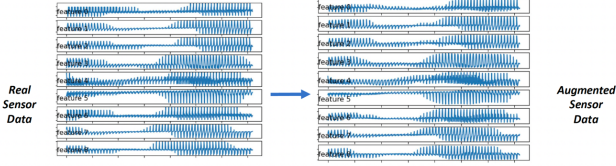


Figure 5. Comparison of real and synthetic time series data generated using a VAE on IHS dataset.

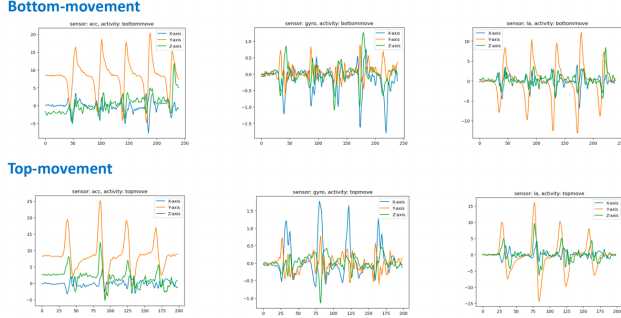


Figure 6. Side-by-side comparison of data acquired by different sensors for a pair of confusing classes in our IHS data set. The rows correspond to the two classes, and the columns correspond to the sensor types.

and **max-based fusion**, which uses:

$$p_i^{(final)}(c) = \max(p_i^{(1)}(c), p_i^{(2)}(c)) \quad (14)$$

our approach selects the better performing student for each class using a membership array \mathcal{A} (Fig 7):

$$p_i^{(final)}(c) = \mathbb{I}[\mathcal{A}(c) = 1] * p_i^{(1)}(c) + \mathbb{I}[\mathcal{A}(c) = 2] * p_i^{(2)}(c) \quad (15)$$

where $\mathbb{I}[\cdot]$ is an indicator function that takes the value of 1 if the condition within the brackets is met. $\mathcal{A} \in \{(a_1, \dots, a_C) | a_i \in \{1, 2\}, i = 1, \dots, C\}$ is obtained by maximizing validation performance using student 1, $s_1(\cdot)$ and student 2, $s_2(\cdot)$. This approach effectively leverages the strengths of both models, particularly in scenarios where one model excels in certain classes while the other does not.

3.4. Algorithmic Details and Speed-Memory Trade-offs of DAF

We provide the algorithmic details of the proposed DAF framework in Algorithm 1. The framework involves training two student models in parallel, one with contrastive distillation and another with additional augmentation, to improve performance on challenging classes. This approach allows for a better speed-memory trade-off compared to simply increasing the size of a base model like LSTM. While running both students in parallel maintains the same latency as a base LSTM, it doubles the memory requirement due to storing two checkpoints. However, this is more efficient than doubling the units in a base LSTM, which would

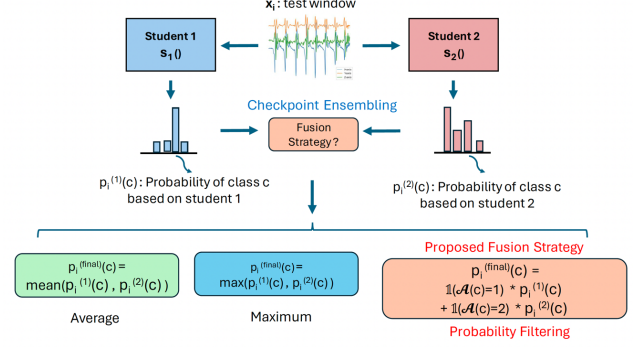


Figure 7. Illustration of the proposed probability filtering technique to combine predictions of both the students using a membership array.

Algorithm 1 DAF Framework Algorithm

- Require:** Labeled dataset $\{\mathbf{x}_i, \mathbf{y}_i\}$ of windows and activity class label.
Ensure: A pair of student models $s_1(\cdot)$ and $s_2(\cdot)$, and membership array $\mathcal{A}[1 : C]$.
- 1: For each class c in $1 : C$, identify $M_c(\cdot)$, a DL model architecture performing best for class c .
 - 2: Randomly initialize weights of $s_1(\cdot)$ and $s_2(\cdot)$.
 - 3: **Contrastive Distillation (CD):**
 - 4: **for** each epoch in $1 : \text{epochs}_{s1}$ **do**
 - 5: **for** each mini-batch \mathcal{B} **do**
 - 6: Obtain $\{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{B}$.
 - 7: Compute $\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{CE}} + (1 - \alpha) \mathcal{L}_{\text{distil}}^{\text{cd}}$ using $\{\mathbf{x}_i, \mathbf{y}_i\}$.
 - 8: Update $s_1(\cdot)$ using $\nabla \mathcal{L}_{\text{total}}$.
 - 9: Save model checkpoint for $s_1(\cdot)$ using validation performance.
 - 10: **end for**
 - 11: **end for**
 - 12: Identify set of classes $\{\tilde{c}\}$ where $s_1(\cdot)$ does not perform satisfactorily.
 - 13: **Contrastive Distillation with Augmentation (CDA) via Generative Model:**
 - 14: **for** each \tilde{c} **do**
 - 15: Synthesize $\{\mathcal{V}(\mathbf{x}_i), \mathbf{y}_i = \tilde{c}\}$ using generative model \mathcal{V} and obtain $\mathcal{X}_{\tilde{c}} = \{\mathcal{V}(\mathbf{x}_i), \mathbf{y}_i = \tilde{c}\} \cup \{\mathbf{x}_i, \mathbf{y}_i = \tilde{c}\}$.
 - 16: Train $s_2 * (\cdot)$ using steps similar to CD with $\mathcal{X}_{\tilde{c}}$.
 - 17: **end for**
 - 18: **Contrastive Distillation with Augmentation (CDA) via Mix-Up:**
 - 19: Form pairs of classes $\{(\tilde{c}_1, \tilde{c}_2)\}$ to obtain mix-up dataset $\{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}\}$.
 - 20: Train $s_2 * (\cdot)$ using original data and mix-up dataset.
 - 21: Using heuristics from validation data, assign the better performing of $s_2 *$ and $s_2 * *$ as $s_2(\cdot)$.
 - 22: **Filtering (F):**
 - 23: Find $\mathcal{A} \in \{(a_1, \dots, a_C) | a_i \in \{1, 2\}, i = 1, \dots, C\}$ that maximizes validation performance using $s_1(\cdot)$ and $s_2(\cdot)$.
 - 24: Return $s_1(\cdot)$, $s_2(\cdot)$, and \mathcal{A} .

increase both memory and latency. Our framework thus offers a promising balance between speed and memory usage.

4. Experimental Results

We conduct experiments on three dataset settings:

1. **In-House Sensor (IHS)** dataset
2. **IHS pseudo Few Shot 5 (FSS)** dataset, and
3. **Public UCI HAR** dataset [1].

The IHS dataset consists of 8 smartphone-based hand move-

ments with triaxial data from accelerometer, gyroscope, and linear acceleration sensors. We perform **sensor fusion** by stacking these attributes: `acc_x`, `acc_y`, `acc_z`, `gyro_x`, `gyro_y`, `gyro_z`, `la_x`, `la_y`, `la_z`.

For the IHS dataset (Table 1), we use a sliding window approach with a length of 50 and step size of 1. The training and testing matrices are of shapes (1394, 50, 9) and (3788, 50, 9), respectively. The FS5 dataset is derived from IHS to mimic labeled data-scarce settings by using only the first 5 training windows per class, resulting in a training matrix of shape (40, 50, 9). The UCI HAR dataset (Table 2) contains 6 human activities and is used for benchmarking, with training and testing matrices of shapes (7352, 128, 9) and (2947, 128, 9).

Our experimental protocol involves training various deep supervised time-series classification models, identifying best-performing teacher models for each class, and applying our **Distillation, Augmentation, and Filtering (DAF) framework**. We use a lightweight LSTM as the student architecture and evaluate performance using **Accuracy** as the metric, reporting both average class-wise and overall average accuracies.

Table 1. Details of the **IHS dataset**

Class	Total samples	Train samples	Test samples	Train windows	Test windows
anticlock	604	181	423	(132, 50, 9)	(374, 50, 9)
back	867	260	607	(211, 50, 9)	(558, 50, 9)
bottom	456	136	320	(87, 50, 9)	(271, 50, 9)
clock	711	213	498	(164, 50, 9)	(449, 50, 9)
front	656	196	460	(147, 50, 9)	(411, 50, 9)
left	1325	397	928	(348, 50, 9)	(879, 50, 9)
right	982	294	688	(245, 50, 9)	(639, 50, 9)
top	365	109	256	(60, 50, 9)	(207, 50, 9)

Table 2. Details of the **UCI HAR dataset**

Class	Train windows	Test windows
walk	(1226, 128, 9)	(496, 128, 9)
walk_up	(1073, 128, 9)	(471, 128, 9)
walk_down	(986, 128, 9)	(420, 128, 9)
sit	(1286, 128, 9)	(491, 128, 9)
stand	(1374, 128, 9)	(532, 128, 9)
lay	(1407, 128, 9)	(537, 128, 9)

4.1. Results on IHS Dataset

In Table 3, we compare the performance of various deep supervised models on the IHS dataset. The ConvLSTM method performs best overall in terms of average class-wise accuracy (**Avg cw Acc**) and average accuracy (**Avg Acc**), as shown in bold. **However, no single method excels across all classes, motivating the use of a multi-teacher setup.** We identify the best-performing model for each class to create a teacher list: $teacherlist = \{ 'transf', 'cnnblstm', 'convlstm', 'convlstm', 'transf', 'cnnlstm', 'convlstm', 'transf' \}$. This list guides the selection of positive teachers for each class.

Table 4 shows that model latency increases with complexity, making the lightweight LSTM suitable for mobile

deployment. We thus use it as the base student model for our DAF framework.

In Table 5, we compare various distilled models against the base LSTM:

- **LSTM**: Base LSTM architecture.
- **TF-LSTM**: Distilled LSTM using only the Transformer as a teacher.
- **CV-LSTM**: Distilled LSTM using only ConvLSTM as a teacher.
- **MT-CSA**: Class-Specific Activation (CSA) multi-teacher approach.
- **CD-KL**: Contrastive Distillation with KL divergence loss.
- **CD**: Contrastive Distillation with Focal loss.
- **CDAF and CDAFs**: CDAF denotes the complete DAF pipeline, encompassing both augmentation and filtering. In contrast, CDAFs is a variant of CDAF that further incorporates the slice and shuffle technique [18] for time-series data.

Values highlighted in green indicate improvements over the base LSTM (in orange), while those in red show significant improvements, particularly by our methods.

4.1.1. Insights on Distillation (D) alternatives

From Table 5, we derive several insights:

- **A single teacher may not be adequate in cases of confusing classes**: Using a single teacher like Transformer (TF-LSTM) or ConvLSTM (CV-LSTM) improves performance on one class but worsens it on another due to inherent confusion between classes, as shown in Fig 6 and Fig 8.
- **Multiple teachers are useful for distillation**: The MT-CSA method outperforms single-teacher variants in terms of average accuracies, highlighting the benefits of multiple teachers.
- **Contrastive Distillation outperforms simple class-specific activation**: Our proposed variants using Contrastive Distillation outperform MT-CSA and the base LSTM.
- **Focal loss over KL-divergence**: Our CD method with Focal loss outperforms CD-KL, which uses KL divergence, indicating Focal loss is beneficial for hard-to-classify examples.
- **Benefits of the Augmentation and Filtering stages**: The CDAF variant with the full DAF pipeline significantly improves performance, especially on confusing classes (bottom: 89.3 from 67.9, top: 59.9 from 35.27).

4.1.2. Insights on Augmentation (A) alternatives

In Table 6, we analyze different augmentation strategies within our DAF framework:

- **CDA-V**: Contrastive Multi-Teacher distilled LSTM with VAE-based augmentation for the top class, improving its performance from 35.27 to 54.59.

Table 3. Performance comparison among various supervised models with different architectures on the IHS dataset

Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc
LSTM	100	92.83	67.9	100	92.7	100	99.22	35.27	85.99	92.19
BLSTM	90.91	90.68	86.35	100	95.62	95.34	97.03	26.09	85.25	90.65
DLSTM	100	88.35	90.04	98.66	94.16	91.47	90.92	19.32	84.12	88.86
ConvLSTM	98.13	98.03	95.94	100	93.19	97.16	98.75	79.71	95.11	96.52
CNN-LSTM	97.86	99.1	94.46	100	96.59	100	94.99	44.93	90.99	95.04
CNN-BLSTM	91.18	99.46	92.62	100	96.11	99.77	97.34	55.07	91.44	95.14
CNN-DLSTM	98.66	91.76	78.97	100	92.46	99.89	86.85	17.39	83.25	89.57
CNN-DBLSTM	94.92	90.32	74.17	100	92.21	98.86	86.7	73.91	88.89	91.45
Transformer	100	71.15	76.75	100	97.81	95.45	87.32	90.82	89.91	90.15

Table 4. Latency comparison of different architectures on the IHS dataset

Method	LSTM	BLSTM	DLSTM	ConvLSTM	CNN-LSTM	CNN-BLSTM	CNN-DLSTM	CNN-DBLSTM	Transformer
Latency	9ms	13ms	18ms	26ms	13ms	19ms	15ms	45ms	280ms

Table 5. Performance comparison among various distilled models against the base student LSTM on the IHS dataset

Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc
LSTM	100	92.83	67.9	100	92.7	100	99.22	35.27	85.99	92.19
TF-LSTM	94.92	82.8	97.42	100	90.27	96.25	91.24	16.91	83.72	88.83
CV-LSTM	100	90.14	43.17	100	91.24	96.81	98.9	46.86	83.39	89.7
MT-CSA	99.47	91.76	90.41	100	90.75	93.52	93.58	23.67	85.39	90.29
CD-KL (Ours)	100	99.28	77.49	100	93.19	99.77	98.75	27.05	86.94	93.29
CD (Ours)	100	94.98	80.44	100	93.43	100	100	30.92	87.47	93.37
CDAF (Ours)	100	96.42	89.3	100	93.67	100	100	59.9	92.41	95.83
CDAFs (Ours)	100	94.98	94.83	100	93.92	100	100	71.01	94.34	96.65

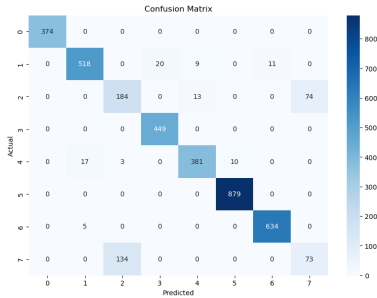


Figure 8. Illustration of the confusion matrix in the test data of IHS using base LSTM.

- **CDA-V-r**: Similar to CDA-V but with added L2 regularization. It slightly improves average performance and the top class, though regularization is unnecessary with VAE augmentation.
- **CDA-MU**: Contrastive Multi-Teacher distilled LSTM with Mix-Up augmentation for confusing classes (bottom, top). This achieves significant gains for both classes (89.3 from 67.9 for bottom, 59.9 from 35.27 for top) and improves average performance.
- **CDA-VMU**: Hybrid augmentation using VAE for the top class followed by Mix-Up for (bottom, top) and (top, bottom). It offers the highest overall average accuracy but is slightly lower than CDA-MU on confusing classes.

Most metrics are improved by our CD variants, as highlighted in green, compared to the base LSTM values in orange. While CD improves average performance (87.47 vs. 85.99), it drops the top class’s performance due to gains in the bottom class. This motivates the use of augmentation to focus on challenging classes.

4.1.3. Insights on Filtering (F)

In Table 7, we evaluate the performance of our **probabilistic filtering** component. We compare two student variants: **CD** (Contrastive Distillation) and **CDA-MU** (Contrastive Multi-Teacher distilled LSTM with Mix-Up augmentation), treated as model 2 and model 1, respectively. The **CDAF** variant, which uses probability filtering, outperforms simple **Averaging (Avg)** and **Maximum (Max)** methods by significant margins. This demonstrates the effectiveness of its decision fusion strategy. To obtain CDAF, we used the membership array $\mathcal{A} = [2, 2, 1, 1, 1, 2, 2, 1]$.

4.2. Results on FS5 Dataset setting

In Table 8, we compare the performance of various deep supervised models on the FS5 dataset setting. The Transformer model performs best, while BLSTM, ConvLSTM, and DBLSTM outperform the lightweight LSTM. Due to the limited training samples, many methods struggle to recognize most classes. The teacher list for this setting is: $teacherlist = \{‘transf’, ‘transf’, ‘convlstm’, ‘blstm’, ‘transf’, ‘transf’, ‘transf’, ‘transf’\}$.

In Table 9, we demonstrate that **our DAF framework boosts the base LSTM’s performance even in the few-shot (FS5) setting with few labeled examples**. The variants CDs, CDA-MUs, and CDAFs are used, with the latter referring to the final variant after probability filtering.

In Table 10, we compare CDAF with simple **Averaging (Avg)** and **Maximum (Max)** methods on the FS5 setting. CDAF outperforms these counterparts using the membership array $\mathcal{A} = [2, 2, 1, 2, 1, 1, 1, 1]$.

Table 6. Analysis with respect to augmentation on IHS dataset

Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc
LSTM	100	92.83	67.9	100	92.7	100	99.22	35.27	85.99	92.19
CD	100	94.98	80.44	100	93.43	100	100	30.92	87.47	93.37
CDA-V	100	91.94	63.84	100	92.94	100	97.34	54.59	87.58	92.53
CDA-V-r	99.47	95.16	69	100	91.97	100	99.22	39.13	86.74	92.69
CDA-MU	96.79	88.17	89.3	100	94.16	96.36	98.12	59.9	90.35	93.19
CDA-VMU	95.72	89.61	83.39	100	92.7	99.32	99.06	57.49	89.66	93.43

Table 7. Demonstration of probability Filtering (F) on IHS dataset

Model 1	Model 2	Method	anticlock	back	bottom	clock	front	left	Right	top	Avg cw Acc	Avg Acc
N/A	N/A	CDA-MU	96.79	88.17	89.3	100	94.16	96.36	98.12	59.9	90.35	93.19
N/A	N/A	CD	100	94.98	80.44	100	93.43	100	100	30.92	87.47	93.37
CDA-MU	CD	Avg	100	90.68	88.93	100	93.92	99.09	99.69	38.16	88.81	93.53
CDA-MU	CD	Max	100	90.5	89.3	100	93.67	99.09	99.69	38.65	88.86	93.53
CDA-MU	CD	Prob Filtering/CDAF (Ours)	100	96.42	89.3	100	93.67	100	100	59.9	92.41	95.83

Table 8. Performance comparison among various supervised models with different architectures on the FS5 dataset setting

Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc
LSTM	34.76	11.29	32.84	10.69	13.14	14.22	18.62	33.82	21.17	18.43
BLSTM	16.84	7.89	35.06	43.88	13.14	38	14.87	28.5	24.77	24.84
DLSTM	0	0	0.37	0	0	0	86.85	67.63	19.36	18.37
ConvLSTM	35.56	23.3	66.79	29.4	10.46	11.95	13.15	36.71	28.41	23.34
CNN-LSTM	0	0.18	7.38	0	0	79.98	30.83	0	14.8	24.31
CNN-BLSTM	0	0	5.9	0	0	0	100	8.7	14.32	17.77
CNN-DLSTM	0	0	0	0	0	0	67.61	58.94	15.82	14.63
CNN-DBLSTM	25.13	50.36	55.35	16.7	7.06	16.38	28.17	43.96	30.39	27.56
Transformer	47.33	62.37	35.42	26.06	69.34	21.5	60.41	91.3	51.72	47.18

Table 9. Performance comparison among various distilled models against the base student in FS5 setting

Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc
LSTM	34.76	11.29	32.84	10.69	13.14	14.22	18.62	33.82	21.17	18.43
CDs (Ours)	36.36	12.54	42.44	12.92	14.11	15.13	22.07	25.12	22.59	20.14
CDA-MUs (Ours)	44.12	7.17	53.51	5.79	13.38	17.18	18.31	22.22	22.71	19.67
CDAFs (Ours)	37.7	10.93	59.41	14.25	13.38	21.27	19.72	22.22	24.86	22.2

Table 10. Demonstration of probability Filtering (F) on FS5 Dataset setting

Model 1	Model 2	Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc
N/A	N/A	CDA-MUs	44.12	7.17	53.51	5.79	13.38	17.18	18.31	22.22	22.71	19.67
N/A	N/A	CDs	36.36	12.54	42.44	12.92	14.11	15.13	22.07	25.12	22.59	20.14
CDA-MUs	CDs	Avg	38.5	10.57	45.02	11.36	13.87	15.59	21.6	24.15	22.58	20.01
CDA-MUs	CDs	Max	38.24	10.93	42.44	12.47	13.87	14.79	21.13	25.12	22.37	19.77
CDA-MUs	CDs	Prob Filtering/CDAF (Ours)	37.7	10.93	59.41	14.25	13.38	21.27	19.72	22.22	24.86	22.2

4.3. Results on UCI HAR Dataset

In Table 11, we compare various deep supervised models on the UCI HAR dataset. The ConvLSTM model performs best in terms of average performance, while the Transformer model performs the worst. The teacher list for this setting is: $teacherlist = \{‘cnnlstm’, ‘blstm’, ‘convlstm’, ‘cnnblstm’, ‘cnnblstm’, ‘convlstm’\}$.

In Table 12, we demonstrate that **our DAF framework improves the base LSTM’s performance on the UCI HAR dataset**. The variants CDs, CDA-MUs, and CDAFs are used, with CDAFs being the final variant after probability filtering. Our approach enhances performance on classes like sitting and standing, where the base LSTM was weaker.

In Table 13, we compare CDAFs with simple **Averaging (Avg)** and **Maximum (Max)** methods on the UCI HAR dataset. CDAFs outperform these counterparts, albeit marginally, using the membership array $\mathcal{A} =$

$[2, 1, 1, 2, 2, 1]$.

4.4. Insights on Speed-Memory trade-off

To assess whether doubling the base LSTM’s size would be beneficial, we compare it with our DAF framework. We create two variants: **LSTM+** (doubling LSTM units) and **LSTM++** (doubling both LSTM and dense units). In Table 14, we report that while these variants increase memory requirements by 2.07x to 4x, they also increase latency by up to 2x, which is undesirable for on-device inference. In contrast, our **CDAFs** method, despite a 2x increase in memory, maintains the same inference latency as the base LSTM.

4.5. Additional Experiments

We report additional experiments:

- Table 15 showcases the performance differences arising out by varying α in the core CD method from eq (7), on the IHS dataset. We noticed an overall stability of our

Table 11. Performance comparison among various supervised models with different architectures on the UCI HAR dataset

Method	walk	walk_up	walk_down	sit	stand	lay	Avg cw Acc	Avg Acc
LSTM	93.55	92.14	98.1	78.62	87.41	100	91.63	91.55
BLSTM	95.77	96.82	97.86	79.02	86.47	100	92.65	92.53
DLSTM	94.15	95.75	99.05	76.99	90.41	100	92.73	92.64
ConvLSTM	92.14	92.57	99.52	81.26	93.42	100	93.15	93.11
CNN-LSTM	96.17	96.6	97.62	73.52	89.85	100	92.29	92.23
CNN-BLSTM	91.13	94.69	98.1	69.04	93.8	100	91.13	91.11
CNN-DLSTM	93.75	95.12	99.05	75.56	89.66	95.34	91.41	91.25
CNN-DBLSTM	94.35	94.48	99.29	84.11	82.14	100	92.4	92.2
Transformer	89.92	89.81	75.48	58.45	92.67	99.63	84.33	84.87

Table 12. Performance comparison among various distilled models against the base student in UCI HAR dataset

Method	walk	walk_up	walk_down	sit	stand	lay	Avg cw Acc	Avg Acc
LSTM	93.55	92.14	98.1	78.62	87.41	100	91.63	91.55
CDs (Ours)	94.76	90.66	96.19	81.67	89.29	100	92.09	92.09
CDA-MUs (Ours)	91.73	97.24	97.86	87.37	78.01	99.81	92	91.75
CDAFs (Ours)	94.56	95.54	97.38	81.67	89.29	99.81	93.04	92.98

Table 13. Demonstration of probability Filtering (F) on UCI HAR dataset

Model 1	Model 2	Method	walk	walk_up	walk_down	sit	stand	lay	Avg cw Acc	Avg Acc
N/A	N/A	CDA-MUs	91.73	97.24	97.86	87.37	78.01	99.81	92	91.75
N/A	N/A	CDs	94.76	90.66	96.19	81.67	89.29	100	92.09	92.09
CDA-MUs	CDs	Avg	94.56	92.78	97.62	84.73	84.59	100	92.38	92.26
CDA-MUs	CDs	Max	94.35	92.36	97.62	84.73	84.59	100	92.27	92.16
CDA-MUs	CDs	Prob Filtering/CDAFs (Ours)	94.56	95.54	97.38	81.67	89.29	99.81	93.04	92.98

Table 14. Effect of increasing base LSTM size (in terms of doubling the number of units in LSTM and Dense layers) vs using our method on the IHS dataset.

Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc	Latency	DiskStorage	Trainable Params
LSTM	100	92.83	67.9	100	92.7	100	99.22	35.27	85.99	92.19	9ms	1.3MB	108360
LSTM+	98.4	90.5	82.29	100	90.75	97.04	98.9	37.68	86.95	91.9	15ms (1.6x LSTM)	2.7MB (2.07x LSTM)	224328 (2.07x LSTM)
LSTM++	94.39	86.02	81.92	100	94.65	99.54	96.56	40.1	86.65	91.55	19ms (2.1x LSTM)	5.2MB (4x LSTM)	429704 (3.9x LSTM)
CDAFs (Ours)	100	94.98	94.83	100	93.92	100	100	71.01	94.34	96.65	9ms (= LSTM)	2.6MB (2x LSTM)	216720 (2x LSTM)

method across different values of α . However, a very low value emphasizes too much importance to the distillation term, and a very high value focuses too less. We noticed an optimal value of 0.1 resulting the best performance, similar to other works in distillation literature.

- In Table 16, we showcase the performances of different variations of our DAF method on IHS dataset, while also highlighting the impact of the shuffle technique [18]. We noticed that the resulting variant CDAFs performs the best, and thus use it for the other two datasets as well.
- Varying λ_{aug} in eq (8) for the augmentation loss term with Mix-Up on the IHS dataset (Table 17).

Table 15. Analysis wrt α in Contrastive Distillation on IHS Dataset

α	Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc
0.1	CD	100	94.98	80.44	100	93.43	100	100	30.92	87.47	93.37
0.01	CD	100	91.94	76.01	100	93.67	100	100	33.82	86.93	92.79
0.5	CD	100	91.76	78.97	100	92.7	97.61	99.37	28.02	86.05	91.9

Table 16. Effect of Shuffling on IHS Dataset

Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc	Mixup	Shuffle	Prob Filtering
CDAF	100	96.42	89.3	100	93.67	100	100	59.9	92.41	95.83	Yes	No	Yes
CDAFs	100	94.98	94.83	100	93.92	100	100	71.01	94.34	96.65	Yes	Yes	Yes
CDA-MU	96.79	88.17	89.3	100	94.16	96.36	98.12	59.9	90.35	93.19	Yes	No	No
CDA-MUs	98.4	74.55	94.83	100	93.43	99.54	61.5	71.01	86.66	86.83	Yes	Yes	No
CDA-VMUs	98.4	84.59	85.24	100	92.7	99.89	61.82	66.18	86.1	87.41	Yes	Yes	No
Base LSTMs	100	86.2	42.8	100	93.92	93.74	64.01	39.13	77.48	82.37	No	Yes	No

Table 17. Effect of Mixup/Aug Loss Hyperparameter on IHS Dataset

λ_{aug}	Method	anticlock	back	bottom	clock	front	left	right	top	Avg cw Acc	Avg Acc
1	CDA-MUs	98.4	74.55	94.83	100	93.43	99.54	61.5	71.01	86.66	86.83
5	CDA-MUs	91.44	88.71	61.99	100	87.83	99.09	89.98	55.56	84.33	89.12
0.2	CDA-MUs	99.73	87.28	49.82	100	93.43	95.79	74.02	34.3	79.3	84.85

5. Conclusions

Our framework offers a novel multi-architecture, multi-teacher setup for knowledge distillation in smartphone-based Human Activity Recognition (HAR), leveraging contrastive learning, focal loss, synthetic data augmentation, and Mix-Up augmentation. It also includes a probability filtering technique to fuse decisions from two student variants without increasing latency, validated across various datasets.

However, the framework has limitations. Consistent training across datasets is challenging due to the need for trial and error in selecting the teacher list and membership array \mathcal{A} , which relies on a good validation set. Augmentation strategies are not deterministic and may fail. The effectiveness of contrastive loss depends on the diversity of negative teachers, influenced by base teacher architectures and their initial performances.

References

- [1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, pages 3–4, 2013. 4
- [2] Debarshi Bhattacharya, Deepak Sharma, Wonjoon Kim, Muhammad Fazal Ijaz, and Pawan Kumar Singh. Ensemble: An ensemble deep learning model for smartphone sensor-based human activity recognition for measurement of elderly health monitoring. *Biosensors*, 12(6):393, 2022. 2
- [3] Runze Chen, Haiyong Luo, Fang Zhao, Xuechun Meng, Zhiqing Xie, and Yida Zhu. A light-weight deep human activity recognition algorithm using multi-knowledge distillation. *IEEE Sensors Journal*, 2024. 2
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020. 2
- [5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 2
- [6] Shizhuo Deng, Jiaqi Chen, Da Teng, Chuangui Yang, Dongyue Chen, Tong Jia, and Hao Wang. Lhar: Lightweight human activity recognition on knowledge distillation. *IEEE Journal of Biomedical and Health Informatics*, 2023. 2
- [7] Gad Gad and Zubair Fadlullah. Federated learning via augmented knowledge distillation for heterogenous deep human activity recognition systems. *Sensors*, 23(1):6, 2022. 2
- [8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2
- [9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2
- [10] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013. 3
- [11] Jingru Li, Sheng Zhou, Liangcheng Li, Haishuai Wang, Jiajun Bu, and Zhi Yu. Dynamic data-free knowledge distillation by easy-to-hard learning strategy. *Information Sciences*, 642:119202, 2023. 2
- [12] Malihe Mardanpour, Majid Sepahvand, Fardin Abdali-Mohammadi, Mahya Nikouei, and Homeyra Sarabi. Human activity recognition based on multiple inertial sensors through feature-based knowledge distillation paradigm. *Information Sciences*, 640:119073, 2023. 2
- [13] Jianyuan Ni, Anne HH Ngu, and Yan Yan. Progressive cross-modal knowledge distillation for human action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5903–5912, 2022. 2
- [14] L Anand Kumar Reddy and S Padmakala. Human activity recognition on smartphones using innovative logistic regression and comparing accuracy of extra gradient boost algorithm. In *E3S Web of Conferences*, page 03024. EDP Sciences, 2024. 2
- [15] Marcin Straczekiewicz, Peter James, and Jukka-Pekka Onnela. A systematic review of smartphone-based human activity recognition methods for health research. *NPJ Digital Medicine*, 4(1):148, 2021. 1
- [16] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 2
- [17] Hayat Ullah and Arslan Munir. A 3dcnn-based knowledge distillation framework for human activity recognition. *Journal of Imaging*, 9(4):82, 2023. 2
- [18] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*, pages 216–220, 2017. 5, 8
- [19] Meng-Chieh Wu, Ching-Te Chiu, and Kun-Hsuan Wu. Multi-teacher knowledge distillation for compressed video action recognition on deep neural networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2202–2206. IEEE, 2019. 2
- [20] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3
- [21] Jian Zhang, Ze Tao, Kehua Guo, Haowei Li, and Shichao Zhang. Hybrid mix-up contrastive knowledge distillation. *Information Sciences*, 660:120107, 2024. 2
- [22] Shibo Zhang, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng, and Nabil Alshurafa. Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors*, 22(4):1476, 2022. 2