Cross-Domain HAR: Self-supervised Learning and Enhanced Finetuning Approaches

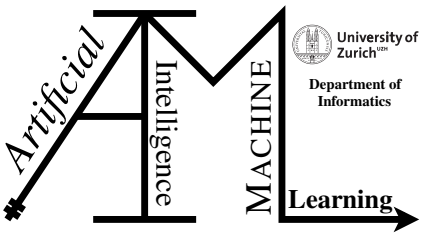
Master's Project

Hyeongkyun Kim, Orestis Oikonomou
21-732-797, 21-715-420

Submitted on
February 09 2024

Project Supervisor
Prof. Dr.Manuel Günther

Co-Supervisor
Prof. Dr.Christian Holz
Max Möbus



Master's Project

Author: Hyeongkyun Kim, Orestis Oikonomou
Email: hyeongkyun.kim@uzh.ch, orestis.oikonomou@uzh.ch
Project period: April 01 2023 - February 09 2024

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

We would like to thank Prof. Dr. Manuel Günther and Max Möbus for their support, and guidance during the course of the project.

Abstract

Wrist-worn accelerometers are increasingly prominent in Human Activity Recognition (HAR) within diverse sectors, encompassing sports and healthcare. Despite their potential, existing HAR models encounter challenges in generalizing across varied, unseen datasets, especially those originating from diverse devices and spanning distinct environmental contexts. This study delves into assessing the performance of Multi-Task Self-Supervised Learning (MTSSL) models in HAR across publicly available test datasets with unique domains, distinct from their training datasets. Furthermore, we explore various additional techniques in a fine-tuning process to generalize the model's performance across different domains. The approach involves integrating classic feature extraction methods, incorporating unknown samples into the training dataset, utilizing a loss function for improved activity distinction, and employing diverse data augmentation techniques. Our study demonstrates that advanced fine-tuning techniques significantly enhance cross-domain generalization and model adaptability. By integrating the three aforementioned enhancements, on the cross-domain target dataset, the model showed a balanced Accuracy improvement of 29% in the closed-set classification and 53% in performance improvement in the open-set classification task. Further, our analysis of different augmentation strategies revealed their varied impact on model performance across testing scenarios. The semantic analysis also sheds light on classification patterns and misclassification trends, emphasizing the value of customized augmentation approaches. These results underline the importance of tailored fine-tuning processes in addressing the challenges posed by dataset diversity and environmental variability, paving the way for more robust and accurate HAR models.

Contents

Chapter 1

Introduction

Over the last decade, large-scale time-series human movement data collection has been revolutionized due to the advancement and widespread adoption of wearable sensors such as smartphones, fitness trackers, and smartwatches. This made continuous, real-time monitoring and analysis of human activities feasible. By using this data, understanding human behavior through activity recognition known as *Human Activity Recognition (HAR)* has become of utmost importance with the trend of digital transformation (Straczkiewicz et al., 2021). It can lead to personalized user experiences, improved health outcomes, and enhanced performance metrics across various fields from healthcare to sports analytics. The numerous researchers aim to contribute to this endeavor, striving for a more transferable representation of HAR Task (Banos et al., 2014; Bin Morshed et al., 2020; Morshed et al., 2019).

With the advancements in deep learning, various deep learning techniques have led to the development of accurate and reliable HAR systems. Especially, the self-supervised learning approach has emerged as a revolutionary technique in the field of HAR tasks by reducing the annotation size and effort (Rani et al., 2023; Yuan et al., 2023). However, they typically assume that the domain of the training and test dataset are identical. Unfortunately, it is often not realistic in applications as the model aims to achieve *"Train once, deploy everywhere"* (Qin et al., 2022). In practical scenarios, a significant challenge still arises when the distributions of training and deployment conditions diverge substantially. This discrepancy can be attributed to various factors, including distinct sensor configurations, variations in device size, and disparities in data collection environments. Such differences pose a threat to the model's generalization performance (Figure 1.1). Therefore we chose the following research question for this project.

Research Question

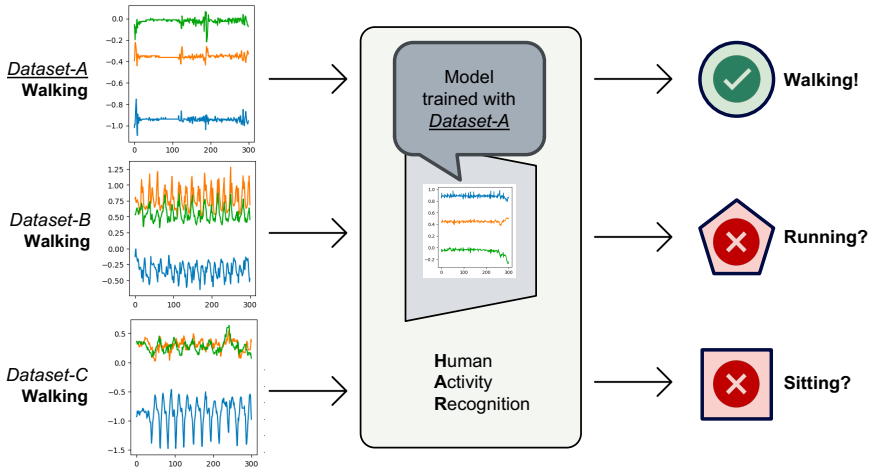
- How to tackle the model generalization issue of the HAR task by enhancing the fine-tuning process while preserving the pre-trained model?

To tackle this research question, we investigate the model generalization performance of the *Multi-Task Self-Supervised Learning (MTSSL)* pipeline introduced by Saeed et al. (2019) and extend the fine-tuning process of this model by integrating different approaches. The first key approach involves adding a classic HAR feature extractor into the MTSSL model architecture. This addition aims to leverage the strength of traditional feature engineering in HAR, proven effective in various studies such as Gjoreski et al. (2016), to enhance the model's capability to understand nuanced human activities. Secondly, one of the problems that may arise as the dataset domain expands is that new types of activity signals, namely unknown activities, occur that existing models cannot predict. Therefore, we incorporated unknown samples into the fine-tuning dataset and applied the objectosphere loss function, a novel approach inspired by Dhamija et al.

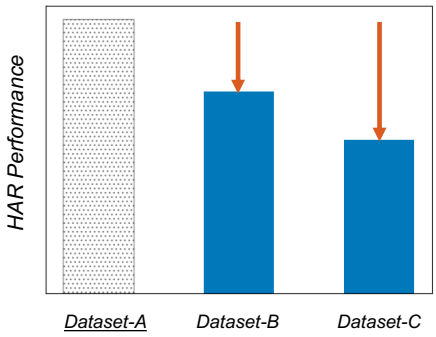
(2018). This technique is designed to expand the model's classification boundaries from the closed-set to the open-set classification in order to classify known classes correctly while also rejecting unknown classes. Lastly, even with the same activity, the intensity or characteristics of the activity signal are slightly different depending on the domain. Hence, we explored several augmentation techniques to artificially increase the domain capability of the training dataset. Inspired by Xu et al. (2023), these augmentations introduce synthetic variability, enabling the model to generalize better across diverse environments and sensor configurations. The whole extended pipeline, incorporating these advancements, is presented in Figure 1.2.

Report Outline This report is outlined as follows.

- **Related Work:** In this chapter we provide a non-technical overview of relevant approaches, such as Multi-Task Self-Supervised Learning, Cross-Domain Evaluation, and open-set Classification.
- **Dataset Preparation:** This chapter describes the process of dataset preparation. It includes the contents of the definition of the HAR dataset domain, types of various datasets, and preprocessing steps for fine-tuning and evaluation steps.
- **Methodology:** In this chapter, we describe the MTSSL model as the base model and introduce other variants. Evaluation matrices are also defined here.
- **Evaluation and Results:** This chapter explains the experimental setup for the model evaluation and its results.
- **Discussion:** In this chapter we interpret the results and answer our research question. Furthermore, we highlight the limitations of our work.
- **Conclusion:** In this chapter we conclude the project with our contribution and suggestions for future work.



(a) Scenario of deploying the trained model into different domains



(b) Performance drops when deploying the trained model into another domain

Figure 1.1: PERFORMANCE DEGRADE WHEN TESTING THE MODEL INTO ANOTHER DATASET. *This figure shows the challenge that the performance of the Human Activity Recognition model is dropped if the domain of the deployment is different from the domain of the training. Figure 1.1a describes the deployment scenario in the real world. The model was trained with a dataset from Dataset-A which is considered as a source domain dataset. However, it is usually deployed into another domain environment, such as Dataset-B and Dataset-C. These different datasets are named as target domain datasets. Even though the MTSSL model is based on transferring the general HAR knowledge into the specific downstream task, the HAR signal domains trained through a fine-tuning process cannot be infinite. As a result, the model trained using a specific domain dataset performs badly when it is deployed into other cross-domain environments (Figure 1.1b).*

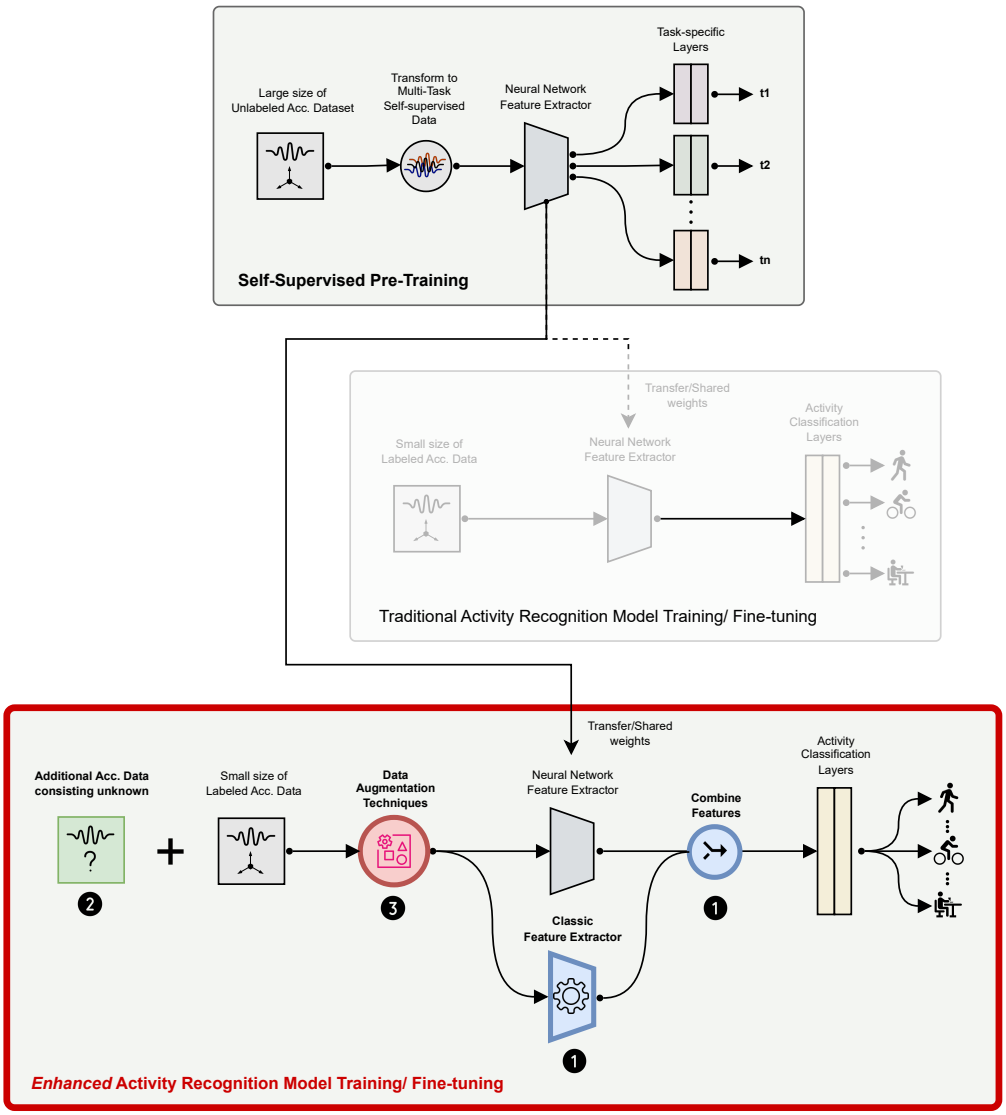


Figure 1.2: ENHANCED MTSSL FINE-TUNING PROCESS. *The figure highlights enhanced fine-tuning steps investigated in this project. It preserves the pre-training process of the Multi-Task Self-Supervised Learning(MTSSL) model. However, compared to the middle one which describes the traditional fine-tuning process of the MTSSL model, the bottom enhanced fine-tuning process marked as a red box, includes 3 additional function blocks to improve the model's generalization performance across different domains. 1. Blue item depicts aggregating classic HAR features on CNN-based features. These additional features are extracted by using statistical and mathematical ways. 2. Green item shows including unknown samples in the fine-tuning dataset and sets the threshold to filter out uncertain predictions. 3. Red item describes the augmentation techniques to increase the domain coverage of fine-tuning datasets.*

Chapter 2

Milestones and Work Distribution

Assuming 30 hours of work per week and a total of 15 ECTS credits with 30 hours of workload per ECTS on average, this Master's project is expected to take 15 weeks. We are a group of 2 students with experience in working online and distributing workload appropriately to work in parallel.

Milestone 1

Week 1 to 2: Literature review regarding the following topics:

- Self-Supervised learning (Multi-Task and Contrastive Predictive Coding).
- Cross-Domain evaluation in the human activity recognition.
- Open-set classification.

Milestone 2

Week 3 to 6: Data preparation & pipeline development

- Review benchmark datasets for the task of human activity recognition.
- Development of fetching and preprocessing code for each of 16 datasets.
- Development of the training pipeline for the base model (MTSSL).

Milestone 3

Week 6 to 8: Explore methodologies

- Explore classic features for human activity recognition and implement them into the pipeline.
- Discuss threshold techniques for the Open-Set classification task and implement the chosen method into the pipeline.
- Examine various augmentations and implement them into the pipeline.

Milestone 4

Week 9 to 11: Experimental setup & Evaluation

- Explore multiple evaluation metrics for Cross-Domain evaluation from closed-set and open-set perspectives.
- Set up the experiment plan and run fine-tuning for 10 different methodologies.
- Discuss the evaluation results and conduct additional evaluations based on the feedback.

Milestone 5

Week 12 to 15: Summarizing and wrap-up the project

- Review the project outcomes and organize the contents for the final report-out.
- Clean up the code repository to be deliverable.
- Write the final report and prepare the presentation.

Tasks		***Practical Jobs***	***Writing Report***
Introduction and Related work		-	O. Oikonomou
Dataset	*Fetching*	O. Oikonomou	O. Oikonomou
	Preprocessing	H. Kim, O. Oikonomou	O. Oikonomou
Methodology	*Base*	H. Kim, O. Oikonomou	H. Kim
	var-C	H. Kim	H. Kim
	var-CU	H. Kim, O. Oikonomou	H. Kim
	var-CUA	H. Kim	H. Kim
Evaluation and Results	*Experimental Setup*	O. Oikonomou	H. Kim, O. Oikonomou
	Processing Results	H. Kim, O. Oikonomou	H. Kim, O. Oikonomou
Discussion and Conclusions		-	O. Oikonomou

Table 2.1: DISTRIBUTION OF PROJECT TASKS AMONG GROUP MEMBERS.

Chapter 3

Related Work

In this chapter, we introduce a brief overview of the related works of multi-task self-supervised learning, the cross-domain evaluation, and the open-set classification for the Human Activity Recognition (HAR) task. The detailed implementation of each technique is introduced in more detail in Chapter 5.

3.1 Multi-Task Self-Supervised Learning

Human Activity Recognition (HAR) has been a focal point in the wearables domain, particularly using body-worn inertial sensors. Initially relying on tree-based methods with handcrafted features, traditional HAR models saw a paradigm shift toward deep learning (Yang et al., 2015). Nevertheless, a significant impediment to the training of deep-learning network models was the challenge of gathering extensive, high-quality labeled datasets.

Multi-Task Self-Supervised Learning (MTSSL), a subset of unsupervised learning, plays a crucial role in addressing the limitations posed by small training datasets. As discussed in studies by Saeed et al. (2019), the fundamental concept of MTSSL involves the transfer of the knowledge pre-trained with unlabeled large-scale datasets into the specific downstream task.

Essentially, this methodology comprises two distinct stages: pre-training and fine-tuning. In the pre-training stage, models extract general features from extensive, unlabeled datasets. This is achieved through the implementation of self-devised pretexts or auxiliary tasks, thereby laying a solid foundation for comprehensive data comprehension. After this, the fine-tuning stage involves applying these pre-trained parameters to a specific downstream task. Importantly, this stage can efficiently utilize a relatively small amount of explicitly labeled data.

A recent study by Yuan et al. (2023) corroborates the notion that an augmented volume of pre-training data correlates with enhanced performance in the downstream task. This underscores the significance of leveraging larger pre-training datasets to augment the efficacy of the subsequent fine-tuning phase.

3.2 Cross-Domain Evaluation

Cross-domain evaluation is an important concept in the field of Human Activity Recognition (HAR), which addresses the challenges of adapting models to operate effectively across diverse contexts. This approach is essential for understanding how well a model can adapt and perform when exposed to new, unseen data environments.

In this context, Lu et al. (2021) has made a significant contribution in this regard. They focused on Substructure-level Matching for Domain Adaptation (SSDA), encapsulated in their Substructural Optimal Transport (SOT) methodology. They employ clustering techniques to better utilize the locality information in activity data, optimizing the coupling of weighted substructures between different domains. This method is particularly effective in improving classification accuracy and efficiency, making it a valuable addition to the field of HAR. In a parallel advancement, Tang et al. (2022) contributed to the accuracy of cross-domain evaluations with their innovative triple attention mechanism. This mechanism is designed to enhance the processing of sensor data by addressing the intricate cross-interaction between sensor dimensions, temporal dimensions, and channel dimensions. This approach results in more accurate activity recognition, leveraging the complex interplays within the sensor data.

Finally, Thukral et al. (2023) addressed the challenge of adapting HAR models to diverse contexts with their 'Cross-Domain HAR' framework. This method utilizes a teacher-student self-training paradigm for effective transfer learning. The teacher model, trained on a labeled source dataset, is used to generate soft pseudo-labels for target data. These pseudo-labels, along with the labeled source data, are then employed to train a student model. This approach stands out for its integration of few-shot learning, where the student model is fine-tuned with a minimal set of labeled target data, significantly improving performance in varied HAR scenarios. Our project takes inspiration from these innovative approaches, aiming to enhance the fine-tuning process of HAR models that are robust and versatile across real-world applications.

3.3 Open-Set Classification

In machine learning, open-set classification introduces specific terminology for understanding and applying its concepts effectively. To lay the groundwork for discussing this approach, we first define key terms as follows, inspired by the framework of Dhamija et al. (2018):

- **Known Classes:** These are pre-defined classes included in the original recognition task. Training in closed-set models is confined to these classes.
- **Unknown Classes:** Representing the classes not encountered by the model during training or validation, these are often termed as 'unknown unknowns'.
- **Negative Classes:** These are a subset of classes added during training or validation, designed to teach the model to reject certain non-target classes.

With these definitions in place, we delve into the concept of open-set classification, which marks a paradigm shift in machine learning. This approach challenges classifiers to not only accurately identify known classes but also to detect when a sample belongs to none of the known categories. It's a crucial capability for preventing false classifications and involves a careful balance between specialization in known categories and generalization towards unknown or open space (Scheirer et al., 2013).

The relevance of open-set classification becomes particularly pronounced in Human Activity Recognition (HAR). Due to the diversity and unpredictability inherent in human activities, correctly identifying and categorizing these activities becomes a complex task. For instance, the study by Yang et al. (2019) introduces a model that enhances HAR systems by generating synthetic samples for unknown activities, thus improving the system's adaptability to novel activities not included in its training dataset. Drawing upon this model and the principles, our project incorporates open-set classification to develop a HAR system capable of efficiently navigating the vast and varied landscape of human activities.

Chapter 4

Dataset Preparation

This chapter details the dataset preparation process for our cross-domain Human Activity Recognition (HAR) study. Central to our approach is the strategic use of diverse datasets for source and target domains, facilitating a nuanced cross-domain evaluation.

The definition of the domain in this context is the unique combination of the type of used device and the setting of the measurement environment. The Human Activity dataset from *the source domain* is mainly used for training the model. In contrast, *the target domain* dataset is utilized to be not seen during training, but used for the testing. In other words, the purpose of Cross-Domain evaluation is to evaluate the model's performance on the target domain under the condition that this model has no chance to learn knowledge about this domain. It is much closer to the real-world scenario.

The Capture24 dataset, our primary source domain dataset, provides a comprehensive set of activity label samples, augmented by 8 additional datasets that will provide extra activity labels for negative and unknown samples. In parallel, our study incorporates 8 other datasets from the target domain which has different domain attributes from the source domain. They offer diverse scenarios essential for evaluating the generalization capability of our methods.

In Section 4.1, we introduce the definition of the domain and its attributes used for this project. After this, in Section 4.2, we lay the groundwork for understanding the specific components that define the source and target domain datasets in the context of different domains. Lastly, in Section 4.3, we delve into the data preparation steps, including windowing, resampling, relabeling, and splitting, which are vital to harmonizing the data across these varied domains for effective fine-tuning and evaluation of our models.

Group	*Device*	*Alias*
Wearable Device	*Axivity AX3*[1]	WD-AX
	GENEActiv[2]	WD-GEN
	Shimmer[3]	WD-SH
	Empatica[4]	WD-EMP
Sensor	*MPU-9250*[5]	S-MPU
	ADXL345[6]	S-ADX
	No info	S-NaN

Table 4.1: THE DOMAIN DEFINITION BY DEVICE TYPE. *This table shows different types of wearable devices that are used in this project. It is largely divided into the Wearable Device Type and Sensor Type. Then, grouped by the commonly used wearable device, and the alias for each device is defined. The Device Type defined for the source domain in this project is 'WD-AX' and it is underlined in the table.*

4.1 Definition of Domain

At the heart of our study lies the innovative use of an unprecedented number of diverse HAR datasets strategically categorized into source and target domain groups to facilitate a comprehensive cross-domain evaluation. The datasets consisting of the source domain serve as the training bedrock, providing our models with a diverse array of scenarios and participant experiences. In contrast, the datasets, that provide the target domain for evaluation, are crucial for testing these models in uncharted territories — environments, activities, sensor conditions, and other aspects representing the domain not covered during training. This approach ensures not only deep foundational learning from the source domain datasets but also a rigorous validation of our models' adaptability and accuracy in new, real-world situations.

The first attribute of the domain is the category of devices employed for the acquisition of Human Activity Signals. Datasets designed for HAR tasks are derived from a spectrum of devices, each characterized by its unique specifications rather than adhering to standardized ones. In real-world situations, fine-tuning must be carried out whenever the device in use undergoes a change or there is an update in the sensor configuration, as such modifications can alter the sensor signal. Consequently, it is imperative to scrutinize the model's performance and its generalizability across a diverse array of devices. Table 4.1 shows commonly used device types in the field of HAR datasets and the reference links to check their diverse specifications. In this project, we curated the dataset to encompass all types of devices, designating the dataset collected by 'Axivity AX3 (WD-AX)' for the source domain.

Apart from the category of device employed for the acquisition of Human Activity Signals, our study focuses on the diversity in the environmental setup for data collection (Table 4.2). This domain attribute is another critical aspect of HAR research, as it profoundly affects the na-

[1]https://axivity.com/product/ax3
[2]https://activinsights.com/technology/geneactiv
[3]https://shimmersensing.com/product/shimmer3-gsr-unit
[4]https://www.empatica.com/en-int
[5]https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250
[6]https://www.analog.com/en/products/adxl345.html

Environment	*Signal Consistency*	*Signal Naturality*	*Alias*
Unconstrained	Low	High	XCON
Semi-Constrained	Medium	Medium	SEMI-CON
Constrained	High	Low	CON
No info	-	-	NaN

Table 4.2: THE DOMAIN DEFINITION BY DATA COLLECTION ENVIRONMENTAL SETUP. *This table shows different types of Data Collection Environmental Setup that are used in this project. Each environment can be divided by the level of consistency and naturality of the measured signal. The environmental setup defined as the source domain in this project is 'XCON' and it is underlined in the table.*

ture and quality of the data collected. In our analysis, we worked with settings that range from unconstrained (natural free-living) environments to controlled constrained (in the laboratory) environments. Collecting data in natural, everyday situations in unconstrained environments (XCON) provides valuable insights into real-world scenarios. This is exemplified in a study comparing diaries with a pair of wearable cameras and accelerometers from Gershuny et al. (2020). In this study, participants wore an accelerometer that tracked their physical activity continuously throughout the 24 hours covered by the diary without any constrained. After that, by using a self-report time-use diary and a camera recording, each participant's activity was mapped into the corresponding human activity label. In contrast, controlled constrained environments (CON) offer more precision and consistency in data collection. This is highlighted in Jarchi (2017). In this study, participants were asked to perform walking, jogging, and bike riding in the lab environment. Specific activity scripts including speed and duration were given to participants for signal consistency. Semi-constrained environmental setting (SEMI-CON) tries to gather activity signals that are consistent but also diverse in the same activities. This setting is described in Roggen et al. (2012). Participants were instructed to follow the sequence of activities in the preset room. However, to measure the signal realistically, there was no specific guidance, such as speed or duration of walking activity, leaving them free interpretation from the participant.

4.2 Description of Dataset

For our Cross-Domain experiment, we selected 17 HAR task datasets for the project among a total of 36 investigated. One of them, known as the Capture24 dataset, was used for the source domain dataset and made the performance baseline in this project. In contrast, 8 other datasets represented the target domain datasets having diverse and different domain attributes with the source domain. The remaining 8 datasets will be used to provide the knowledge of unknown samples to extend the model into the open-set classifier task.

To better understand the dataset organization and a clear explanation for further process, let us first depict the dataset by using symbols $D = \{SD, TD\}$, where SD is the source domain dataset and TD is the target domain dataset. Furthermore, it has two more categories of parameters: train/test and known(kn)/negative(neg)/unknown($unkn$). Consequently, we denote the source domain train dataset of known samples as SD_{kn}^{train} and the corresponding test set as SD_{kn}^{test}. Let SD_{neg}^{train} be the train set of negative samples and SD_{neg}^{test} be the corresponding

Id	*HAR Dataset*	*Used for*	*Reference*
1	Capture24	SD_{kn}^{train} and SD_{kn}^{test}	Walmsley et al. (2022)
2	Gotov	SD_{neg}^{train} and SD_{neg}^{test}	Paraschiakos et al. (2021)
3	Harvardleo		Leotta et al. (2021)
4	Householdhu		Hu et al. (2022)
5	Pamap2		Reiss (2012)
6	Realworld		Sztyler (2016)
7	Wisdm		Weiss (2019)
8	Commuting	SD_{unkn}^{test}	Garcia (2014)
9	Paal		Climent i Pérez et al. (2022)
10	Adl	TD_{kn}^{test} and TD_{unkn}^{test}	Bruno et al. (2014)
11	Forthtrace		Karagiannaki et al. (2016)
12	Ichi14		Borazio et al. (2014)
13	Mendeleydaily		Ruzzon et al. (2020)
14	Newcastle		van Hees et al. (2018)
15	Oppo		Roggen et al. (2012)
16	Selfback		Sani et al. (2016)
17	Wristppg		Jarchi (2017)

Table 4.3: TOTAL 17 DATASETS USED FOR DIVERSE PURPOSES. *This table shows the overall HAR dataset used for this project and its purposes. There are 9 datasets for the source domain and 8 datasets for the target domain. Under the same purpose, the dataset names are listed in alphabetical order.*

test set. Similarly, let SD_{unkn}^{test} be the test set of unknown samples. Accordingly, we denote the source domain full train set as $SD^{train} = SD_{kn}^{train} \cup SD_{neg}^{train}$, and the source domain full test set as $SD^{test} = SD_{kn}^{test} \cup SD_{neg}^{test} \cup SD_{unkn}^{test}$. For the target domain dataset TD, only the test set including known and unknown samples are utilized: $TD^{test} = TD_{kn}^{test} \cup TD_{unkn}^{test}$. Table 4.3 shows all HAR datasets for the use of various purposes.

4.2.1 Datasets for the Source Domain

In this section, we describe the datasets representing the source domain, Capture24 which has the source domain attribute: Collected using Axivity AX3 (WD-AX) under the unconstrained environment (XCON). The other 6 datasets for negative samples and 2 datasets for unknown samples are explained additionally.

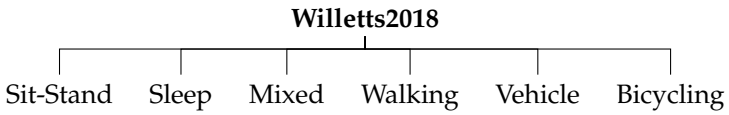


Figure 4.1: WILLETTS2018 HUMAN ACTIVITY LABEL TYPE. *Following the labeling policy in Willetts2018, all samples in Capture24 dataset can be labeled as one of six. Only these six labels are considered to be known labels that can be classified from the model in this project.*

Known Samples

In our study, we use the Capture24 dataset as the primary dataset in the source domain, a benchmark in the field of human activity recognition. Renowned for its extensive application across numerous research studies, this dataset is instrumental in advancing our understanding of human behaviors in naturalistic settings (Walmsley et al., 2022; Gershuny et al., 2020; Doherty et al., 2018; Willetts et al., 2018). In particular, the dataset involves data from WD-AX(Activity AX3) that is collected from participants in Oxfordshire between 2014 and 2016. The dataset encapsulates a wide range of daily activities, recorded at a 100Hz sampling rate over approximately 4,000 hours. Out of these, more than 2,500 hours of activities have been labeled, and our study exclusively utilizes this labeled portion of the dataset.

The extensive volume of data and the capacity for measurements in unconstrained environments facilitate the diverse labeling of samples within the Capture24 dataset, accommodating various research objectives. Notably, for this project, we intend to predominantly employ the *Willetts2018* labeling scheme (Willetts et al., 2018). This methodology prioritizes the categorization of human activities into six primary classes: *Sit-Stand*, *Sleep*, *Mixed*, *Walking*, *Vehicle* and *Bicycling*, as illustrated in Figure 4.1. These labels consist of the main known labels during our classification processes. Also, it is worth mentioning that the labels are distilled from a larger set of more detailed annotations (Chan Chang et al., 2021) and are specifically chosen for their relevance to unconstrained behavior patterns. Concentrating on these six labels allows us to leverage the rich diversity of the dataset, ensuring both the manageability and the clarity of our analysis.

Negative Samples

Following the principles of open-set classification, one of our proposed methodologies, outlined in Section 5.3, involves training the network to discern known samples from unknown samples by incorporating negative samples into the training process. Therefore, we need to prepare another source that can provide negative samples to the training dataset, for that we include six additional datasets to gather those samples specifically. In Table 4.3, the datasets used to provide negative samples are specified between Id 2 and Id 7. These datasets were chosen based on their ability to introduce unknown labels in contrast to the existing known labels in Capture24, specifically those related to activities such as *Kicking*, *Jumping* and *Vacuuming*. This approach enables the integration of novel negative labels not previously addressed in Willetts2018 label types, aiming to identify samples with low certainty, thereby enhancing the robustness and accuracy of our model.

Known Classes	*Negative Classes*	*Unknown Classes*
Bicycling	Catching	Blow Nose
Mixed	Clapping	Commuting
Sit-Stand	Cutting Vegetable	Open Bottle
Sleep	Dribbling	Put on a Jacket
Vehicle	House Cleaning	Put on a Shoes
Walking	Jumping	Put on Glasses
	Kicking	Salute
	Open/Close Drawer	Sneeze Cough
	Relaxing	Take off a Jacket
	Stir-Frying Vegetable	Take off a Shoes
	Using Mouse	Take off Glasses
	Vacuuming	

Table 4.4: ACTIVITY CLASSES FOR EACH GROUP IN SOURCE DOMAIN DATA. *This table informs the set of activity classes for known, negative, and unknown samples in the source domain dataset. Classes for each group are listed alphabetically. HAR Datasets to be used for each group are selected to prevent overlapping activity classes between groups. Activity labels in known classes are used as it is, but other labels in negative classes and unknown classes are renamed as 'unknown' by preprocessing.*

Unknown Samples

To evaluate the performance of open-set classification, we prepare another 2 datasets that can produce unknown samples to the source domain test dataset. These are listed on Table 4.3 in the ID 8 and 9. Compared to negative samples, these samples are never seen during the training. Therefore, the activity class of the unknown sample should not overlap with the known class or the negative class. Datasets used to provide unknown samples in the source domain were selected under these conditions. As a result, activity classes for each group of samples can be grouped like Table 4.4

4.2.2 Datasets for the Target Domain

We employ 8 different HAR datasets as target domains for cross-domain evaluation. It can encompass combinations of various domain characteristics without overlapping with the source domain - Axivity AX3(WD-AX), Unconstrained(XCON). The domain for each target dataset is depicted in Table 4.5. This evaluation approach is integral to ensuring that our models are robust and effective in various real-world applications, mirroring the complexity and diversity of human behaviors. Table 4.3 provides an overview of the diverse range of datasets used in our study, from ID 10 to 17 correspond to this.

To offer a comprehensive understanding, we provide below detailed descriptions of each of the 8 target domain datasets employed in our study. These descriptions highlight key characteristics, data collection methods, and the types of activities recorded in each dataset:

HAR Dataset	*Domain*		*Has*	*Has*
	Device	*Env.*	*Known?*	*Unknown?*
Adl	S-NaN	CON	✓	✓
Forthtrace	WD-SH	CON	✓	✓
Mendeleydaily	S-MPU	SEMI-CON	✓	✓
Oppo	S-NaN	SEMI-CON	✓	✗
Selfback	WD-AX	CON	✓	✗
Wristppg	WD-SH	CON	✓	✗
Ichi14	S-ADX	CON	✓	✗
Newcastle	WD-GEN	SEMI-CON	✓	✗

Table 4.5: 8 DIFFERENT TARGET DOMAIN DATASETS FOR CROSS-DOMAIN EVALUATION. *This table shows domain specification and whether it contains known or unknown samples. It can be seen that it is configured not to overlap with the characteristics of the source domain 'WD-AX & XCON'. All target domain datasets provide at least one known label. However, unknown samples are only sourced by Adl, Forthtrace, and Mendeleydaily.*

- **ADL** This dataset records 16 volunteers performing 14 activities of daily living as Human Motion Primitives (HMP). They provide volunteers with a wrist-mounted tri-axial accelerometer and ask them to perform each motion primitive multiple times.
- **FORTHTRACE** This dataset is collected from 15 volunteers wearing 5 wearable devices provided by Shimmer, on different body positions including the wrist. They are asked to perform 16 different activities related to sitting, standing, and walking.
- **MENDELEYDAILY** This dataset is organized with 9 activities of daily living from 10 volunteers with IMU sensors. There are sequences of activities to ask and these are recorded using an RGB Camera for data labeling purposes.
- **OPPO** This dataset collects human activities from 4 users with a customized motion jacket equipped with sensors at various points across the body, including the wrist area. Users are asked to follow a high-level script but leave them free interpretation to achieve natural execution.
- **SELFBACK** This dataset has 9 activity classes recorded with Axivity AX3 from 33 participants. Each activity is performed by each user for approximately 3 minutes.
- **WRISTPPG** This dataset records mainly photoplethysmography (PPG) signals, but also other sensor signals including the accelerometer on the wrist. There are 8 participants to ask for specific 4 activities, walking, running, and easy/hard bike riding.
- **ICHI14** This dataset is created for *Sleep* detection with an wrist accelerometer sensor. 42 users participated in the sleeping lab session which is monitored for at least one night under the given circumstance.

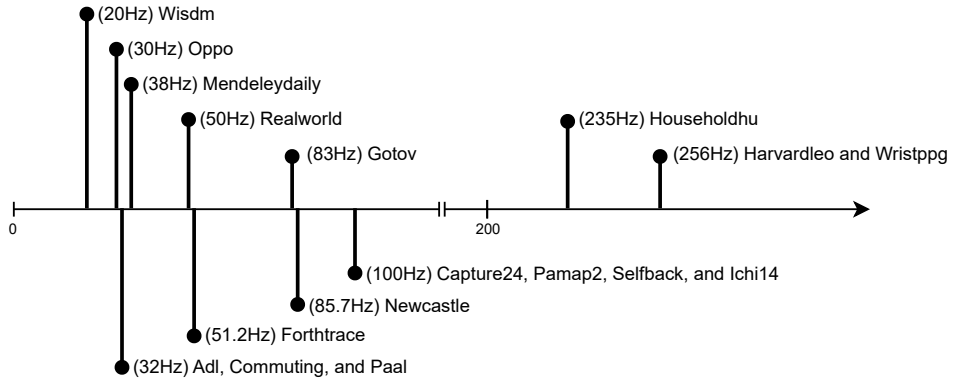


Figure 4.2: DIFFERENT SAMPLING RATES BY EACH HAR DATASET. *This figure shows how different the sampling rates between the datasets are. For consistent training and evaluations, we re-sampled all dataset into 30 Hz.*

- **NEWCASTLE** This dataset collects sleeping accelerometer data for the task of*Sleep*classification. 28 Participants used the device GENEActiv and were invited to participate in the study at the hospital.

4.3 Dataset Preprocessing

This section focuses on the preprocessing techniques employed on tri-axial accelerometer data gathered from wrist-worn activity trackers, leading to the formation of the datasets mentioned earlier. Despite their inherent differences in properties, these datasets were processed to achieve a uniform format before fine-tuning and evaluation phases.

4.3.1 Resampling

The utilized 17 datasets present a variety of sampling rates in a range of 20 to 256 Hz, as can be seen in Figure 4.2. To standardize our approach we decided to resample the signals linearly. Following the same approach as in Yuan et al. (2023), each dataset has been linearly re-sampled to a resolution of 30 Hz. This decision was guided by the understanding that most human activities have a frequency of less than 10 Hz. Moreover, according to Nyquist's theorem, accurately capturing these frequencies without aliasing requires a sampling rate at least twice the highest frequency present in the signal. Thus, a resampling rate of 30 Hz that exceeds the presumed Nyquist rate of 20 Hz is fitting and essential to prevent loss of useful signal.

4.3.2 Windowing

We segmented the accelerometer signals into windows of equal duration and frequency based on Yuan et al. (2023); Bulling et al. (2014). This approach treats each window as an independent input for HAR models, allowing us to label each window with a specific activity class. In particular, as human activities are continuous, often blur into one another making it difficult to define

exact activity boundaries. In Bulling et al. (2014), they employed a moving window over the time series data to extract segments for subsequent processing. The window size influences the delay in the recognition system and the precision of segmentation. Lastly, following the same approach as Yuan et al. (2023) we used a 10-second window that slides by 5 seconds, recognizing that this approach is neutral and does not depend on the specific type and structure of the underlying time series data.

4.3.3 Relabeling

The main objective of dataset preprocessing is to ensure uniformity across all activity samples from the 17 datasets. However, a notable challenge arises due to variances in labeling even for the same human activity across different datasets. For example, the *Walking* activity is labeled as *Walk, Walk at slow pace, Walking upstairs, Walking fast*, and so on. This divergence not only complicates classification using a unified model but also poses challenges for consistent evaluation. Especially, in the context of cross-domain evaluation tasks, the crucial step of realignment that all original labels are standardized into a single scheme becomes necessary. Therefore, we opted to realign the labels of the remaining 16 datasets to match each label into one of Willetts2018 type labels (Figure 4.1): *Sit-Stand, Sleep, Mixed, Walking, Vehicle, and Bicycling*. The specific steps involved in this process are illustrated in Figure 4.3 and described as follows:

1. ***Is it similar with one of the labels of Willetts2018 type?***
   If a label from the non-Capture24 dataset matches one of the six primary labels, *Sit-Stand, Sleep, Mixed, Walking, Vehicle, and Bicycling*, change the label into the corresponding one. For instance, *Sitting and Standing* is relabeled as *Sit-Stand*.

2. **Is it described in 'annotation-label-dictionary.csv' of Capture24 dataset?**
   When the original label does not align with the Willetts2018 type, we turn to the original annotations of the Capture24 dataset for the reference of further steps. This process is facilitated by a detailed file 'annotation-label-dictionary.csv'[7], which is provided with Capture24 dataset and contains both the original annotations of Capture24 and their corresponding label types from several studies. Let's assume that there is a sample of the label *Eating*. Table 4.6 shows one part of this file used for this example scenario. The activity *Eating* is not one of 6 labels defined in Willetts2018 type. However, we can find annotations in the file that describe the activity *Eating* and the corresponding label type in Willetts2018. Therefore, we refer to this file to identify the most frequently related Willets2018 label of *Eating* and, finally, relabel it into *Sit-Stand* in this case.

3. ***What if neither 1 nor 2 applies to this label?***
   It is possible to happen that the label is not one of the Willetts2018 type but also, there is no description of this label in the annotation file. All activities that could not be relabeled based on the previous steps are assigned a new label: *Unknown*, since the activity has never been seen during the Capture24 data collection.

In Section A.3, an example of the relabeling process and the relabeling results for each HAR dataset are presented.

[7] https://ora.ox.ac.uk/objects/uuid:99d7c092-d865-4a19-b096-cc16440cd001/

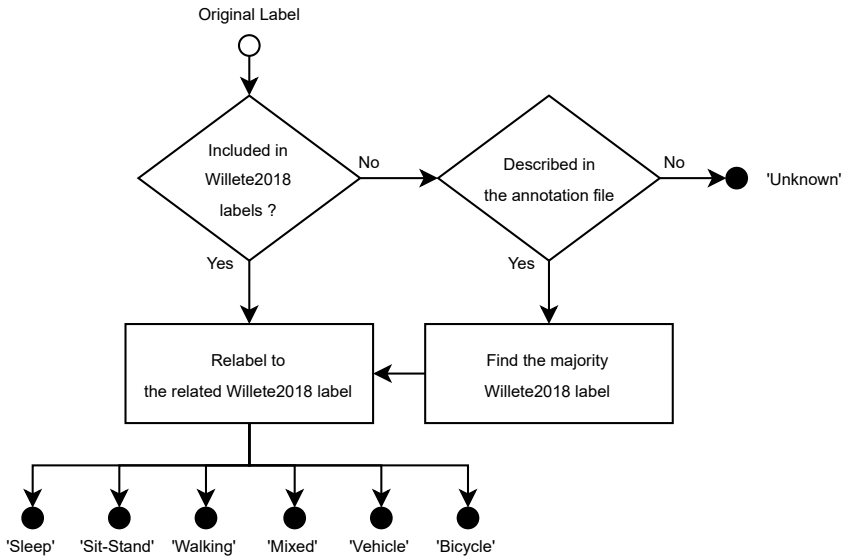


Figure 4.3: FLOWCHART FOR RELABELING PROCESS. *This figure shows the workflow of relabeling during the data preprocessing. The original label from non-Capture24 datasets is changed to either one of the labels in Willetts2018 or 'Unknown'.*

4.3.4 Dataset Splitting

After standardizing the formats of various datasets through preprocessing, we proceed to reorganize and generate three distinct data splits for our experimental purposes. Table 4.7 shows the counts of each sample by data splits.

First of all, the training dataset for the source domain (SD^{train}) is assembled by combining 75% of Capture24 (Id 1) and an equal proportion of negative samples drawn from datasets (Id 2 to Id 7). This set of samples serves as the training data for various models detailed in Chapter 5, excluding the base model and var-C models, which do not require a negative class during training. Secondly, the source domain test dataset (SD^{test}) comprises the remaining 25% of Capture24 and negative samples. Additionally, we introduce unknown samples sourced from datasets (Id 8 and Id 9) to assess the performance of open-set classification. Lastly, the test dataset for the target domain (TD^{test}) is constructed using the remaining 8 datasets (Id 10 to Id 17). As a result, TD^{test} consists of 6 known labels and unknown samples from diverse domain set.

annotation	label:Willetts2018
eating standing indoor/outdoor;MET 2.0	*Mixed*
eating standing alone or with others;MET 2.0	*Sit-Stand*
buying foods or drinks as a takeaway;MET 2.3	*Sit-Stand*

Table 4.6: EXAMPLE OF THE ANNOTATION IN WILLETTS2018 LABEL TYPE. *This table shows the part of the result for the activity 'Eating' in the annotation columns and its corresponding Willetts2018 labels. The Willetts2018 label, 'Sit-Stand', 'Sleep', 'Walking', 'Bicycling', 'Vehicle', and 'Mixed', do not have 'Eating'. However, it can be considered as 'Sit-Stand' since it appears in the annotation and 'Sit-Stand' is the majority label corresponding to Willetts2018.*

Dataset	***Total***	***Known classes (D_{kn})***						***Negative***	***Unknown***
		sl	***ss***	***wk***	***mx***	***vh***	***bc***	***(D_{neg})***	***(D_{unkn})***
SD^{train}	757.1K	271.6K	288.4K	42.4K	80.7K	19.0K	6.6K	48.3K	-
SD^{test}	216.4K	68.4K	65.1K	14.9K	34.9K	15.3K	2.2K	12.1K	3.5K
TD^{test}	395.5K	370.6K	9.6K	10.4K	2.5K	-	0.6K	-	1.7K

Table 4.7: SAMPLE COUNTS FOR EACH CLASS BY DATA SPLITS. *This table shows the number of samples for each class by data splits. The name of the known class is abbreviated: 'sleep' → 'sl', 'sit-stand' → 'ss', 'walking' → 'wk', 'mixed' → 'mx', 'vehicle' → 'vh', 'bicycle' → 'bc'*

Chapter 5

Methodology

In this project, we employs a systematic approach to enhance Human Activity Recognition (HAR) for cross-domain datasets through a series of model variations. In this chapter, we introduce the base model and progressively deliver enhancements in subsequent variants during the downstream task. Also, we present a variety of evaluation metrics that will assist us in evaluating certain strengths and weaknesses of our models.

The first subsection Section 5.1 is about the base model. It utilizes Multi-Task Self-Supervised Learning (MTSSL) methodologies, incorporating a ResNet-type feature extractor. In Section 5.2, the first variant model 'var-C' is established to increase the generality of extracted features to cover the common HAR task dataset for the downstream task. With Section 5.3, we introduce the subsequent variant model 'var-CU'. On top of 'var-C' features, we also trained it to distinguish between known and unknown samples by adding additional samples to the training dataset intentionally. Consecutively, Section 5.4 describe the final variant model 'var-CUA'. It extends the model 'var-CU' capabilities further by incorporating diverse augmentation techniques during fine-tuning. Lastly, 5.5 outlines the array of evaluation metrics we employ, including quantitative measures such as *balanced Accuracy* and *balanced Open-Set Classification Rate*, alongside qualitative assessments like the visualization of feature spaces, providing a comprehensive evaluation of our models' performance across various dimensions.

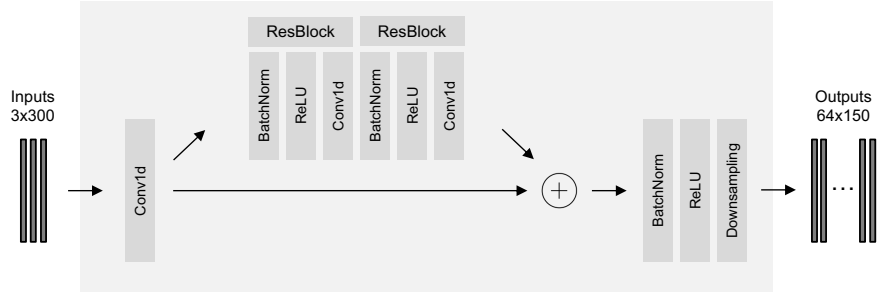
5.1 *base*: Multi-Task Self-Supervised Learning

In this project, we choose the Multi-Task Self-Supervised Learning (MTSSL) method introduced by Saeed et al. (2019) as a foundation model for HAR tasks. The MTSSL model, situated within the broader realm of the HAR field, offers an innovative solution to challenges associated with the requisite for extensive and accurately labeled datasets.

5.1.1 Network architecture

The referenced MTSSL architecture employed a ResNet-v2 (He et al., 2016) as the backbone of the feature extractor. A total of five residual blocks are layered (Table 5.1), and each residual block consists of an additional 1D convolutional layer at the beginning and a downsampling unit at the end (Figure 5.1). All 1D convolutions have the same kernel size 5 with 2 paddings on both sides. In addition, to prevent the downside of model performance caused by the shifting of an input signal, we applied a specialized downsampling technique into the deep network introduced by Zhang (2019). This technique uses a 1D convolution with the blurred box kernel

Figure 5.1: 1ST LAYER OF MTSSL BACKBONE ARCHITECTURE. *This figure shows the first layer of the base MTSSL feature extractor. Several layers of this configuration come together to form an overall feature extractor. The output size of the convolutional layer, the number of residual blocks, and downsampling configurations are parameterized.*



Layer	*Input Size*	*Conv1d Output Size*	*# of Residual Blocks*	*Downsampling*		*Output Size*
				Factor	*Order*	
1	3 x 300	64 x 300	2	2	2	64 x 150
2	64 x 150	128 x 150	2	2	2	128 x 75
3	128 x 75	256 x 75	2	5	1	256 x 15
4	256 x 15	512 x 15	2	5	1	512 x 3
5	512 x 3	1024 x 3	0	3	1	1024 x 1

Table 5.1: PARAMETERS FOR THE FEATURE EXTRACTOR IN THE BASE MTSSL. *This table shows the parameters of each layer in the feature extractor. Input and output size are displayed in the form of 'Channel $\times$ Length'. Following this setting, the 3-axes accelerometer signal '3 $\times$ 300' passes through the feature extractor and is finally transformed into a feature set size of '1024 $\times$ 1'.*

on each channel set by two attributes: *Factor* and *Order*. With these parameters, it creates the blurred box kernel values and appropriate size of padding and stride. Consequently, the length of each channel is exactly reduced into $1/Factor$.

The classification layer that follows the feature extractor is simply constructed with fully connected linear layers. However, it is used in different forms according to the purpose of each process, pre-training and fine-tuning. For pretraining, the last layer of the feature extractor is fully connected with multiple binary output layers. Each fully-connected layer is used to detect a change in a signal affected by a predefined multi-task. On the other hand, the fine-tuning has one type of fully connected layer which has an intermediate layer of size 512. In other words, this fully connected layer for fine-tuning is configured to classify the 6 labels defined in the HAR correctly. The classification layers for each purpose are expressed in the first and second steps of Figure 5.2, respectively.

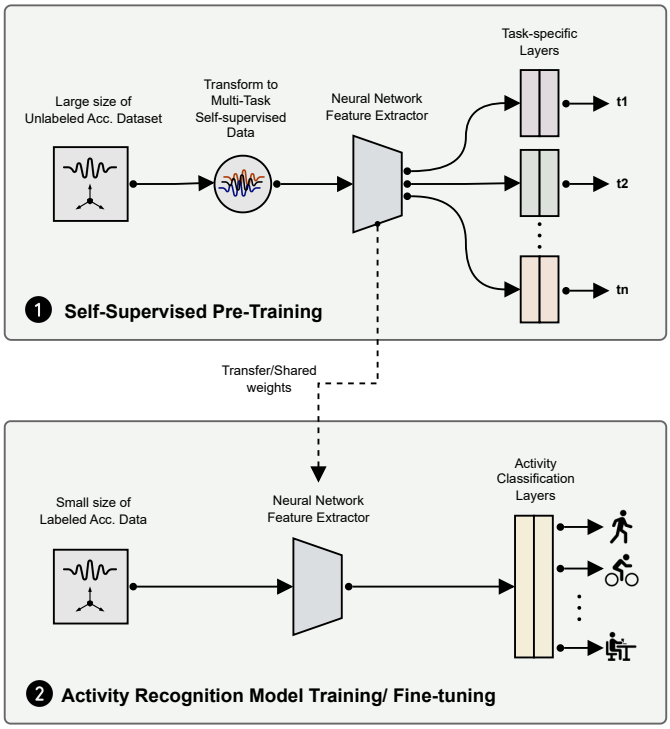


Figure 5.2: PROCESS OF MULTI-TASK SELF-SUPERVISED LEARNING FOR HAR TASK. *This figure shows the basic workflow of Multi-Task Self-Supervised Learning(MTSSL) for HAR task. The methodology involves training a temporal convolutional network to recognize diverse transformations as an initial task (Step 1). The transformations are selected and randomly applied to the original unlabeled samples beforehand. Lastly, the learned weights of the feature extractor are transferred to enhance the performance of the actual activity recognition model in the subsequent step (Step 2).*

5.1.2 Model training

The base model, MTSSL, has two phases for training. First, during the pre-training phase, the network is trained to learn the general features of the HAR signal by classifying self-defined tasks. In Algorithm 1, the large unlabeled accelerometer data D_U is selected and transformed by following the guidelines of pre-defined multi-tasks. For multi-tasks, we define a set of $|T|$ distinct transformations (or tasks) $T = \{J_t(.)\}_{t \in T}$, where $J_t(.)$ is a function that applies a particular signal alteration technique t to the temporal sequence $x \in D_U$ to yield a transformed version of the signal $J_t(x)$. The network $P_\theta(.)$ that has a common feature extractor and individual head for each task, takes an input sequence x and produces $|T|$ logit values for each task. In this network, θ represents the learnable parameters. Eventually, The backbone network layer is trained by updating θ whether each sample is transformed into t or not. To do so, the binary cross-entropy losses from each multi-task t are calculated and the mean loss is used to update network weights θ_P (5.1).

$$\mathcal{L}_{PT}(x, y) = -\frac{1}{|T|} \sum_{t \in T} y_t \, log \left[Sigmoid(P_\theta(x)_t) \right] + (1 - y_t) \, log \left[1 - Sigmoid(P_\theta(x)_t) \right] \tag{5.1}$$

Algorithm 1: Pre-training of Multi-task Self-Supervised Learning

Input: unlabeled sample set D_U, sample channel size C, sample length L, multi-tasks T, numbers of epochs E_P
Output: Self-supervised network P
initialize (X, Y) where $X \in \mathbb{R}^{(\,|D_U|,\,C,\,L\,)}$ and $Y \in \mathbb{R}^{(\,|D_U|,\,|T|)}$;
initialize P with parameters θ_P;
// Labeled data generation for multi-task self-supervision;
for *each instance* $x \in \mathbb{R}^{(\,C,\,L)}$ *in* D_U **do**
  initialize $y = \{y_1, \ldots, y_{|T|}\}$;
  for *each transformation* $t \in T$ **do**
    change (x, y_t) to $(J_t(x), 1)$ or $(x, 0)$
  end
  insert (x, y) to (X, Y);
end
// Pre-training the network for the multi-task self-supervision;
for *each epoch* e_p *from 1 to* E_P **do**
  Randomly sample a mini-batch of m samples from (X, Y);
  Update θ_P by descending along its gradient;
  $\nabla_{\theta_P} \left[\frac{1}{m} \sum_{i=1}^{m} \mathcal{L}_{PT}(x_i, y_i)\right]$
end

Secondly, the trained backbone network layer is transferred and fine-tuned for the specific downstream task. Since the transferred layer has already learned how to extract features from the general signal, the dataset for fine-tuning does not need to be as large as the pertaining dataset. However, it has to be labeled appropriately to serve the specific purpose of downstream tasks. This process is well illustrated in Algorithm 2. The labeled dataset for fine-tuning is represented as D_L, and a set of $|\mathcal{C}|$ activity classes needs to be classified. The network $F_\theta(.)$ has the same feature extractor with $P_\theta(.)$, but it includes only one classification head, which has the output size of activity classes $|\mathcal{C}|$. Before training begins, the feature extractor parameters θ_F are initialized to the corresponding parameters of the pre-training network θ_P. It means the network exploits the knowledge gained from large unlabeled data D_U. In the fine-tuning process, the learnable parameter update is based on the multi-label cross-entry loss with the softmax-activated function (5.2). However, labeled HAR data sets in reality generally contain an imbalance problem of classes. Therefore, we try to solve it by assigning a weight according to the sample class to the loss value. Each class weight is inversely proportional to their frequency in D_L (5.3).

$$\mathcal{L}_{FT}(x, y) = -\log\left[Softmax_c(F_\theta(x))\right], \quad \text{where } y \text{ is } c \in \mathcal{C} \tag{5.2}$$

$$\psi_c = \left[\frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_c(y_i)\right]^{-1}, \quad \text{where } D_L = \{(x_i, y_i)\}_{i=1}^{N} \text{ and } c \in \mathcal{C} \tag{5.3}$$

Algorithm 2: Fine-tuning of Multi-task Self-Supervised Learning

Input: Labeled Finetuning Dataset D_L, numbers of epochs E_F, Pre-trained model parameters θ_P
Output: Activity classification model F with $|\mathcal{C}|$ classes
calculate and initialize the class weight ψ_c;
initialize learnable parameters θ_F;
transfer feature extractor parameters of θ_P to θ_F;
// Finetuning the network for the specific classification task;
for *each epoch* e_f *from 1 to* E_F **do**
  Randomly sample a mini-batch of labeled samples from D_L, $\{(x_1, \ldots, x_m), (y_1, \ldots, y_m)\}$;
  Update θ_F by descending along its gradient;
  $\nabla_{\theta_F} \left[\frac{1}{m} \sum_{i=1}^{m} \psi_{y_i} \left(\mathcal{L}_{FT}(x_i, y_i)\right)\right]$
end

5.2 *var-C*: Adding Classic Feature Extractor

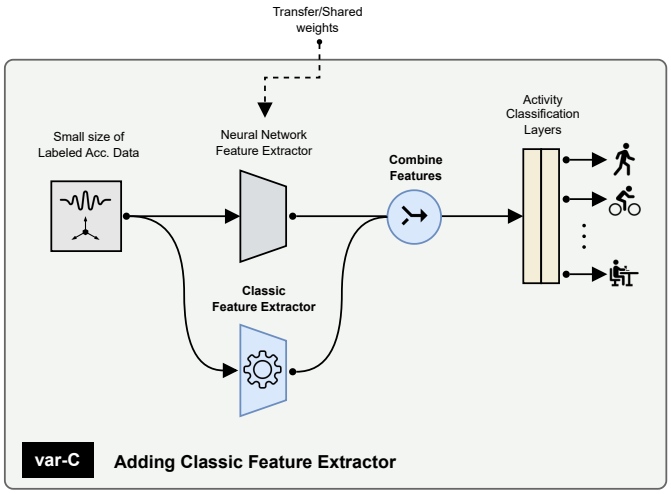
This variant model was designed to enhance the type and number of features extracted from the MTSSL model. We selected various mathematical features traditionally used in HAR (Bao and Intille, 2004; Ravi et al., 2005) and named them the classic features to distinguish them from existing ResNet-based features. As shown in Figure 5.3, these classic features are used with the feature extractor to deliver more intuitive and robust information to the classification layer.

As described in Table 5.2, the classic features are divided into two types depending on what type of data is used: Axis-based(12) and Point-based(10). Axis-based classical features can be obtained as statistical values of each of the three accelerometer axes. Therefore, all statistical items have three values each. Point-based classical features are calculated using 300 Euclidean distance values calculated from (x, y, z) sample points included in a window. In addition to basic statistical values and frequency analysis values, this type of classic feature also includes specific measurements for the accelerometer-based activity signal, *Euclidean Norm Minus One (ENMO)* and *Mean Amplitude Deviation (MAD)*. ENMO is used for adjusting for a gravity effect on the signal via subtracting a fixed offset of one gravitational unit (Van Hees et al., 2013). Moreover, MAD has the benefit of separating sedentary and pace-specific ambulatory activities from each other (Aittasalo et al., 2015).

Hence, a total of 22 classic features are appended to the existing 1024 feature output and all 1046 characteristics are delivered to the fully connected layer for classification. However, in this process, the range of classical features shows a significant difference from the original ResNet-based features. This range difference of feature values causes problems that slow the convergence speed of the model or lower its performance. The main factor behind this is that the model's weight-updating process can be highly biased to large feature nodes. To prevent this problem, we additionally performed normalization for each feature in the same batch.

The change from base to var-C can be explained through the newly defined downstream network $F_\theta(.)$. It is originally defined as $\mathcal{C}_\theta\left(\mathcal{F}_\theta\left(.\right)\right)$, which represent $\mathcal{F}_\theta\left(.\right)$ as the feature extractor and $\mathcal{C}_\theta\left(.\right)$ as the classification layer. For this variant model, we used an additional feature extractor $\mathcal{F}_{classic}\left(.\right)$ to consider classical features, but also the technique of batch normalization $BatchNorm$ to re-scale all features in $\left(\mathcal{F}_\theta\left(x\right), \mathcal{F}_{classic}\left(x\right)\right)$ in the range of 0 and 1 by each. Eventually, $\mathcal{C}_\theta\left(.\right)$ of the model var-C takes features from both $\mathcal{F}_\theta\left(.\right)$ and $\mathcal{F}_{classic}\left(.\right)$ with the same scale (5.4).

Figure 5.3: PROCESS OF VAR-C MODEL FINE-TUNING. *It shows the activity recognition model architecture of var-C. Compared to the base model, it has Classic feature extractor additionally. Finally, 1024 features from the neural network and 22 classical features are appended together and used to predict the activity through the classification layer.*



$$F_\theta(x) = \mathcal{C}_\theta\left(BatchNorm\left(\mathcal{F}_\theta\left(x\right), \mathcal{F}_{classic}\left(x\right)\right)\right) \tag{5.4}$$

$$\text{where } \mathcal{F}_\theta\left(.\right) \in \mathbb{R}^{1024} \text{ and } \mathcal{F}_{classic}\left(.\right) \in \mathbb{R}^{22} \tag{5.5}$$

5.3 *var-CU*: Learning with Unknown Samples

In real-world HAR tasks, it is impossible to establish a dataset that can cover all human activity classes. This restriction is problematic when responding to unknown activity samples (x, y), which has y as $c \notin \mathcal{C}$ where $\mathcal{C}$ is the set of the known activity class. Therefore, if unknown samples can be removed through a certainty-based threshold in the prediction process, it is expected to prevent the case that the model unconditionally predicts all samples as one of the known classes. To implement this concept, the training data and loss function were changed while maintaining the model structure used in the model var-C. As a result, it can predict not only the activity but also the accurate certainty level of the result. With the predicted information, unknown samples are separated from the predicted value set through post-processing such as the softmax threshold. The concept of the model var-CU is described in Figure 5.4.

Loss Function

There are two main ways to effectively remove unknown samples from cross-domain environments during prediction. The first is eliminating samples that are not known classes using thresholding, and the second is adding dummy classes to the existing known class dataset so that the features of the unknown sample are also learned during the training process. The approach of Dhamija et al. (2018) properly uses these two methods together. So, we decided to

Data Type	*Feature Item*		*# of Outputs*
Array of each axis in a 10-sec window	Mean		3
	Standarad Deviation		3
	Peak-to-Peak		3
	Correlation Coefficient		3
Euclidean norm of each sample points (x, y, z) in a 10-sec window	Mean		1
	Standard Deviation		1
	Peak-to-Peak		1
	Kurtosis		1
	Skewness		1
	ENMO		1
	MAD		1
	Freq. Analysis	Spectral Entropy	1
		Peaks	2

Table 5.2: 22 CLASSICAL FEATURES. *This table shows 22 feature items implemented in the classical feature extractor. A total of 22 features are extracted from a 10-second window. Each item can be divided into Axis-based and Euclidean Norm-based according to the type of data used to obtain features.*

apply it to this project. The paper introduced a simple but effective new form of loss function: *Entropic open-set and Objectosphere loss*, that will help model learning features of unknown samples and thresholding after the prediction.

- **Entropic open-set loss**

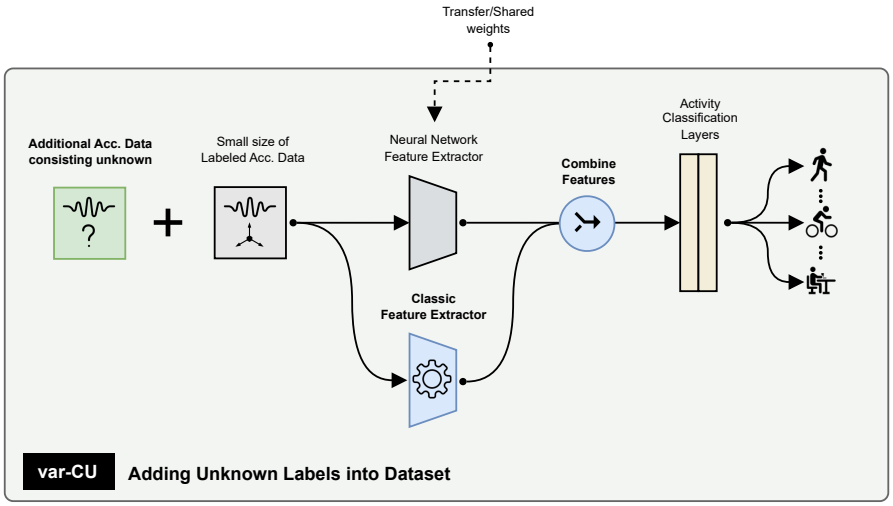
  This loss function maintains the categorical cross-entropy loss upon softmax-activated logits to known labels but, in the case of an unknown label, it uses the maximum entropy distribution of uniform softmax scores over the known classes. It is described in (5.6) with following the definition of $F_\theta(x)$ in (5.4). As a result, this loss function affects network layers to consider not only the high probability for the correct prediction of known samples but also having maximal entropy to the resulting probability distribution against unknown samples.

$$\mathcal{L}_{ent}(x, y) = \begin{cases} -\log\left[Softmax_c(F_\theta(x))\right] & \text{where } y \text{ is } c \in \mathcal{C} \\ -\frac{1}{|\mathcal{C}|}\sum_{c' \in \mathcal{C}} \log\left[Softmax_{c'}(F_\theta(x))\right] & \text{where } y \text{ is } c \notin \mathcal{C} \end{cases} \tag{5.6}$$

- **Objectosphere Loss**

  The term *Objectosphere* has a meaning of the boundary with a magnitude ξ in the deep feature space. Following the equation below, this part of the loss function trains the network to put known samples out of this boundary whilst pushing unknown samples inside of it. $\varphi(x) \in \mathbb{R}^N$ for each sample x represents the feature vector with size N and $\|\varphi(x)\|^2$ is the magnitude of it. Consequently, by summing Entropic open-set with Objectosphere loss (5.7), known samples have large feature magnitude with low entropy, and unknown samples have small feature magnitude with high entropy. In other words, it helps to easily

Figure 5.4: PROCESS OF VAR-CU MODEL FINE-TUNING. *It shows the activity recognition model architecture of var-CU. Compared to the var-C model, it uses negative samples together during training to get an appropriate threshold filtering out unknown samples. The added blocks are colored as Green in the diagram.*



distinguish between known and unknown samples by the softmax threshold after the inference. Finally, this loss function is used for updating the fine-tuning network parameter θ_F (5.8).

$$\mathcal{L}_{obj}(x,y) = \mathcal{L}_{ent}(x,y) + \lambda \begin{cases} \max(\xi - \|\varphi(x)\|, 0)^2 & \text{where } y \text{ is } c \in \mathcal{C} \\ \|\varphi(x)\|^2 & \text{where } y \text{ is } c \notin \mathcal{C} \end{cases} \tag{5.7}$$

$$\mathcal{L}_{FT}(x,y) = \mathcal{L}_{obj}(x,y) \tag{5.8}$$

Note that this loss function has λ and ξ that require an optimization process. However, the optimization process requires a lot of computing power and is outside the scope of this project. Therefore, we simply set λ as 1 and ξ as 1.38, which is the 90th percentile $\|\varphi(x)\|^2$ value when testing with $x \in SD_{kn}^{test}$ on the *var-C* model with .

As we refer to SD^{train} in Table 4.7, the small size of fine-tuning data is prone to inherit imbalances on the data between known classes, but also total known samples and unknown samples. Especially, the size inequality of known and unknown samples can act as a factor that hinders the influence of the Entropic open-set and Objectosphere loss function on the model. Hence, while maintaining the weighting ratio between the existing known classes (5.3), we also tried to consider the weighting between the known and the unknown sample (5.9). For the convenience of the symbols to be used in the following equation, the labels in the training dataset are depicted as y_i with i is $1...N$, and the set of unknown activity labels is defined as $c' \notin \mathcal{C}$.

$$\psi_c = \begin{cases} \left[\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_c(y_i)\right]^{-1} \left[1 - \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{c'}(y_i)\right]^{-1} & \text{where } c \in \mathcal{C} \\ \left[\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{c'}(y_i)\right]^{-2} & \text{where } c \notin \mathcal{C} \end{cases} \tag{5.9}$$

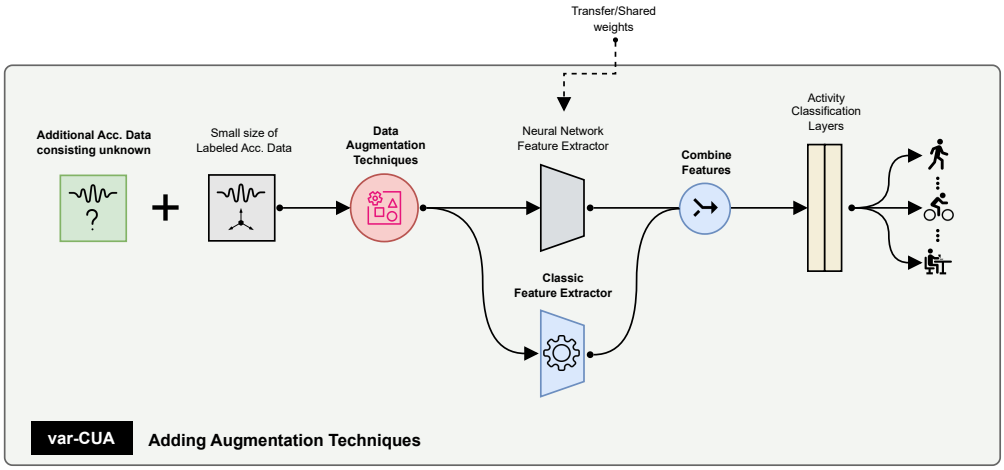


Figure 5.5: PROCESS OF VAR-CUA MODEL FINE-TUNING. *It shows the activity recognition model architecture of var-CUA. Compared to the var-CU model, it has Data Augmentation Process during the training dataset.*

5.4 *var-CUA*: Training Data Augmentation

This variant model augments the training dataset in several forms to keep the model performance constant even on various domain datasets. The importance of data augmentation techniques for HAR tasks has also been researched in Um et al. (2017); Xu et al. (2023). They investigated various data augmentation techniques for Parkinson's Disease Monitoring tasks. As a result, appropriate augmentation improves the classification performance by around 12% from the original model which has no augmentation processes.

Augmentation Techniques

In this project, we choose 3 different data augmentation techniques that are commonly used in the wearable device dataset: *Switching Axes*, *Rotating Axes*, and *Amplitude Scaling*. Figure 5.6 shows 10-second window samples augmented to a different form. These augmentation techniques are used depending on the type of variant model, which can be confirmed through the alias augmentation techniques added at the end of the model name. For example, in the case of a model using the switching axes technique, it is indicated as var-CUA-s, and when two or more techniques are used together, it is indicated as var-CUA-xxx.

- **Switching Axes (s)**

  The axis orientation of accelerometer sensors can be varied by the type of wearable device since there is no universal standard for sensor axis settings across all wearable devices. Therefore, we apply the switching axes for one of the augmentation techniques in this experiment. As a result, the original axes order x, y, and z are switched randomly into one of 6 different combinations of 3-axes orders. All possible sets of switching schemes are chosen with a uniform distribution.

- **Rotating Axes (r)**

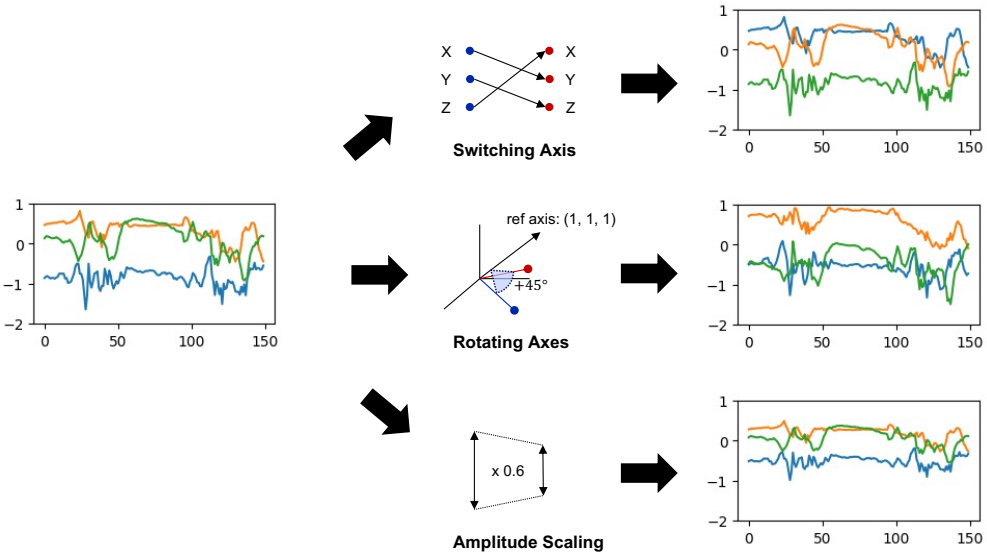


Figure 5.6: EXAMPLE OF 3 DIFFERENT TYPES OF THE AUGMENTATION TECHNIQUE. *This figure shows an example of the result of data augmentation. After applying 3 different types of data augmentation, the original signal input (left) is transformed into one of the signals (right). It is also possible that multiple augmentations are applied on the sample at the same time. (Switching Axes: (x,y,z)→(y,z,x), Rotating Axes: 45° with a reference axis (1,1,1), Amplitude Scaling: 0.6)*

The augmentation of rotating axes is beneficial to overcome the challenges coming from different participants. Since the sensor placement is different between participants or its position on the human body, the difference in the angle of the sensor's default axes is a common occurrence in real-world scenarios. Hence, by rotating all axes to some degree together, the training sample can be augmented to handle various scenarios from unseen participants and wearing locations. To rotate axes, the model multiplies the rotation matrix which includes the information of randomly selected angle($-\pi \sim \pi$) and the reference axis for rotating. In other words, each sample point (x, y, z) in a 10-sec window is rotated by a randomly selected angle around the reference axis.

- **Amplitude Scaling (a)**

  This approach involves changing the magnitude of the data in a window by applying a random scalar. Depending on participants and wearable device settings, the actual amplitude of the signal may differ in a certain range. To tackle this signal invariance, scaling with various factors is a commonly used technique in time-series sensors. However, if the scaling factor is too large, it may also damage the label information. Therefore, we limited the range of the scalar factor from 0.6 to 1.4 linearly.

As a result, a total of seven different var-CUA models are created: var-CUA-s, var-CUA-r, var-CUA-a, var-CUA-sr, var-CUA-sa, var-CUA-ra, and var-CUA-sra. During fine-tuning each model, all training samples go through an augmentation process. Therefore, the overall process of var-CUA training can be expressed as shown in Figure 5.5.

5.5 Evaluation Metrices

Following model training, we conduct a thorough evaluation of our various models for the different domains. To achieve this, a series of qualitative and quantitative evaluation matrices are used in this project.

5.5.1 Balanced Accuracy

Identifying the ratio of the number of correct predictions to the total test of samples can be achieved by calculating the overall accuracy of a model. However, in cases where datasets have an imbalanced number of each class, the performance of the frequently-appeared labels will be dominant in the overall performance. To tackle that, a balanced Accuracy is needed, by employing, for example, a macro-averaging method (5.10). For convenience, the Softmax function is symbolized as $S(x)$. This method calculates the accuracy separately for each known class $c \in \mathcal{C}$. Hence, it involves only the known classes of each dataset, denoted as $\mathcal{D}_{c\in\mathcal{C}}$ and averages the result across all classes. This technique guarantees that each class, irrespective of its size in the dataset, equally contributes to the overall accuracy metric.

$$balanced\ Accuracy = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{|\{x \mid x \in \mathcal{D}_{kn,c} \wedge \arg\max \mathrm{S(x)} = \mathrm{c}\}|}{|\mathcal{D}_{kn,c}|} \tag{5.10}$$

5.5.2 Balanced Open-Set Classification Rate

For model creation cases that involve unknown samples like var-CU and var-CUA, it is important to assess a model's ability to distinguish correctly between known samples as well as satisfy a specific false positive rate for the unknown samples. For that scope, Dhamija et al. (2018) proposed an Open-set Classification Rate. This is an evaluation metric that conceptually parallels the Receiver Operating Characteristic (ROC) curve by illustrating a classification model's performance across various classification thresholds. Moreover, it diverges by being specifically designed for open-set classification tasks, which entail handling datasets that include unknown samples. Its calculation process commences with the computation of the softmax function $S(x)$ for each test sample. Then, depending on the sample whether it is the set of known samples $\mathcal{D}_{\mathcal{C}}$ or unknown samples $\mathcal{D}_{\mathcal{U}}$, both the Correct Classification Rate (CCR) and the False Positive Rate (FPR) are computed by adjusting the softmax score threshold θ, as delineated in (5.11).

In case of class imbalances in the used datasets, a macro-averaging method should be employed for the calculation of CCR. In other words, for each class $c \in \mathcal{C}$, the CCR is calculated individually at the specified threshold, and then these rates are averaged across all the classes in $\mathcal{C}$. Macro-averaging ensures that each class, regardless of its size or frequency within a dataset, contributes equally to the overall CCR metric. This approach is particularly crucial for datasets, where class imbalances could skew the evaluation if not accounted for.

The main goal of this evaluation metric is to identify the appropriate threshold for each model and the best-performed model configuration in the point of *balanced CCR* when FPR equals 0.1. This refined methodology allows us to comparatively evaluate models in a manner that is both equitable and representative of their ability to classify across a diverse set of classes. In the curve depicting *balanced OSCR*, a higher threshold corresponds to the left side, while a lower threshold corresponds to the right. Ideally, a robust and accurate classifier achieves a high *balanced CCR* at a certain FPR. Also, it is worth mentioning that the value of *balanced CCR*

when FPR equals 1 represents *balanced Accuracy* used in the closed-set evaluation since $S(.) > \theta$ becomes always true.

$$\begin{aligned} FPR(\theta) &= \frac{|\{x \mid x \in \mathcal{D}_{\mathcal{U}} \wedge \max S(x) \geq \theta\}|}{|\mathcal{D}_{\mathcal{U}}|} \\ balanced\ CCR(\theta) &= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{|\{x \mid x \in \mathcal{D}_c \wedge \arg\max S(x) = c \wedge S_c(x) > \theta\}|}{|\mathcal{D}_c|} \end{aligned} \tag{5.11}$$

5.5.3 Semantic Analysis

In addition to assessing a model's capability to accurately identify activities, exploring how test samples are distributed within a deep feature space is pivotal. This examination can reveal potential semantic similarities among relabeled categories. This semantical analysis can be utilized by visualizing each class sample on the feature space. By analyzing its clustering, the model becomes more explainable.

However, the visualization of the high-dimensional features, 1024 or 1046 in our case, is impractical due to the dimensional limitations perceivable by the human eye. To navigate this complexity, we incorporate the cutting-edge dimensionality reduction technique Uniform Manifold Approximation and Projection (UMAP), as proposed by McInnes et al. (2020). UMAP aims to project the data into a lower-dimensional space that closely mirrors the original high-dimensional space's fuzzy topological structure. This method is particularly chosen for its ability to handle non-linearities, akin to t-SNE, while also offering superior runtime performance.

Note that the primary focus of UMAP is on the relative distances between samples since this method is based on calculating the distance between samples. In other words, UMAP tries to make a group of samples which has smaller distance as dense as possible, but far apart from other groups. Hence, the distribution and scale of the original feature values may not be preserved after the reduction.

Chapter 6

Experiments and Results

In this chapter, we focus on the experimental setup and evaluating our advanced Human Activity Recognition (HAR) models. Section 6.1 presents the experimental setup, detailing the fine-tuning processes for three model variants, var-C, var-CU, and var-CUA, and the choices of evaluation metrics used to assess their predictions across closed-set and open-set scenarios for both SD^{test} and TD^{test}. Then, Section 6.2 presents the results of these evaluations, across the different variants and scenarios.

6.1 Experimental Setup

In this section, we describe the pre-trained model that served as a foundation for our experiments, along with details of the fine-tuning process and the dataset utilized therein. Subsequently, we introduce various scenarios in which we will assess the performance of the variant models, along with the specific metrics we plan to employ.

6.1.1 Pre-Training and Fine-Tuning

Pre-training and Fine-tuning follow the processes of Algorithm 1 and Algorithm 2, respectively. However, depending on each model variant, some of the existing network components, such as Loss Function $\mathcal{L}(.)$ and the Downstream Network model $F(.)$, are replaced and the configuration of the dataset SD^{train} is changed.

First of all, the pre-training model, we use the pre-trained public HAR model because the size of the unlabeled dataset is hard to access publicly and the process of pre-training needs high computing power. However, the shared model pre-trained by (Yuan et al., 2023), utilizes the UK-Biobank dataset containing the data from more than 100,000 participants wearing the device in seven days. Also, it follows well-defined self-supervision tasks: *Arrow of Time*, *Permutation*, and *Time warping*. It is described shortly in Table 6.1

After then, perform HAR tasks for label types in Figure 4.1, we further fine-tuned the pre-trained model on the small-size labeled dataset, SD^{train} in Table 4.7. Since the base and var-C model only consider closed-set classification, 708.8K samples are used without 48.3K negative samples. On the other hand, other models var-CU and var-CUA, used all samples 757.1K in SD^{train} including negative samples. The entire network is optimized using Adam (Kingma and Ba, 2017) with a learning rate of 1e-3 and a batch size of 1000.

Self-Supervision Task	*Description*
Arrow of Time	Reverses the signal by flipping it along the time axis. It is the same with playing the signal backward.
Permutation	Divides the signal into time-series segments and rearranges them in a random order. It introduces invariant permutations on the samples.
Time warping	Alters the duration of arbitrary segments of the signal, effectively introducing random variations in speed by slowing down and speeding up.

Table 6.1: SELF-SUPERVISED TASKS USED FOR THE PRE-TRAINING MODEL. *This table lists the item of self-supervision tasks which are used for the pre-trained model.*

6.1.2 Evaluation Scenarios

After fine-tuning the pre-trained model for each purpose, all variant models will undergo a rigorous evaluation using datasets classified by SD^{test} and TD^{test}, as detailed in Table 4.7. Initially, we utilize SD^{test} to establish a baseline for performance. Our analysis then extends to include assessing model performance on TD^{test}. To ensure an effective assessment, our models are evaluated from various perspectives: closed-set, open-set, and semantic analysis.

First, we assess our models in the context of a closed-set scenario, as we aim to measure their ability to identify known activities correctly. Considering the significant label imbalances in SD^{test} and TD^{test}, we will use *balanced Accuracy* to achieve a fair evaluation as presented in 5.5.1. Since this evaluation scenario considers how many known samples are correctly classified, we exclude all unknown samples from test datasets in this evaluation.

Furthermore, as our models also incorporate unknown samples, it is crucial to evaluate them in open-set classification scenarios. Here, given the label imbalances in SD^{test} and TD^{test}, we employ *balanced OSCR*, as presented in 5.5.2 for a comprehensive assessment. The algorithm was applied to the following processes. First, a softmax value is derived through logit values of each sample. Next, balanced CCR and FPR are calculated using the obtained softmax values as parameters. Finally, after plotting each pair of balanced CCR and FPR, the value of balanced CCR and threshold when FPR is 0.1 are checked through interpolation.

Finally, we aim to expand our analysis to the qualitative aspects, which are critical for capturing potential complex relationships within the data. This approach helps us identify semantic overlaps and intricate patterns in our models. To achieve this, we explore the feature values of test samples from each model by visualizing them on the feature space. However, our original feature space is too large for practical visualization. Therefore, as mentioned in 5.5.3, we adopt the UMAP dimensionality reduction technique. In the initial stage of semantic analysis, we carefully resample 1000 samples by each class from SD^{test} and TD^{test}, respectively. It ensures an even distribution of samples across labels to mitigate potential visualization errors caused by data imbalance. Lastly, we perform standardization on the feature vectors to fit into the consistent scaling for the UMAP algorithm. In conclusion, the dimension of feature vectors for each sample is reduced from 1024 to 2 for the *base* model, and 1046 to 2 for other variant models.

6.2 Results

In this section, we present the results of our experiments, analyzing the evaluation metrics presented in Section 6.1. The findings are based on each model variation applied to SD^{test} and TD^{test}. We are focusing on understanding the effects of different classical features, unknown samples, and augmentation techniques. The results are substantiated through a combination of statistical data, confusion matrices, and graph representations.

6.2.1 Balanced Accuracy

For closed-set scenario experiments, we calculate the balanced Accuracy for each model variation involving only known samples. As outlined in Table 6.2, we observe that balanced accuracy yields similar patterns in the results for both SD^{test} and TD^{test}. Notably, the var-CUA variant demonstrates the highest accuracy in both domains, with the combinations of augmentation techniques consistently leading to improved outcomes. In particular, the accuracy in SD^{test} increased by approximately 0.01 (2%) from the base model, while in TD^{test} increased even more by approximately 0.06 (29%). This emphasizes the significant impact of data augmentation techniques on enhancing the correct classification of known labels, especially in the case of TD^{test}.

In the SD^{test}, the combination of switching and rotating axes (var-CUA-sr) emerges as the most effective variation, achieving an overall balanced Accuracy of 0.6772. This was closely followed by the variation that combines rotating axes and amplitude scaling (var-CUA-ra), and then by the model that uses only rotating axes (var-CUA-r). However, these two model variations have little difference from the best-performed one with a marginal difference of only 0.001 to 0.002. Similarly, in TD^{test}, the combination of rotating axes and amplitude scaling (var-CUA-ra) exhibited the best performance, followed closely by the combination of switching axes and amplitude scaling (var-CUA-sa), and integration of all three techniques (var-CUA-sra). This trend reaffirms that combinations of augmentation techniques generally surpass both the other models that have no augmentation, and the var-CUA model with a single augmentation.

Further examination of the classification outcomes for each category was conducted through an in-depth analysis using the confusion matrix, as depicted in Figure 6.1. Each cell in the matrix means the number of samples corresponding to the pair of the truth and predicted label. In addition, normalization was applied for each true value to minimize the visual effect of the label imbalance. The analysis highlights that the SD^{test} yields a high rate of true positives for all labels in every model. Nonetheless, the labels *Mixed* and *Walking* present noticeable challenges in terms of classification, with a notable number of samples being misclassified by each other. However, by improving the model from base to var-CUA, this problem becomes resolved gradually. It can be shown that the instances of *Walking* samples erroneously classified as *Mixed* have reduced, while the correct classification of *Walking* samples has increased.

Intriguingly, when comparing findings from the SD^{test}, the confusion matrix for the TD^{test} indicates a substantial misclassification across almost all labels. This is particularly evident in the case of *Sleep* samples, which are frequently misclassified as *Sit-Stand*. Also, in the case of *Bicycling* samples, they are frequently misclassified as *Mixed* or *Vehicle*. However, in the case of var-C, there is a significant improvement in their correct classification and a decrease in the ratio of misclassification from each other. Moreover, it is worth mentioning that compared to the base model, all variant models have led to improvement in the correct classification of *Walking* samples like the similar trend in SD^{test}. Last but not least, despite the absence of *Vehicle* samples in TD^{test}, each model consistently misclassified *Sit-Stand* and *Bicycling* samples as *Vehicle*.

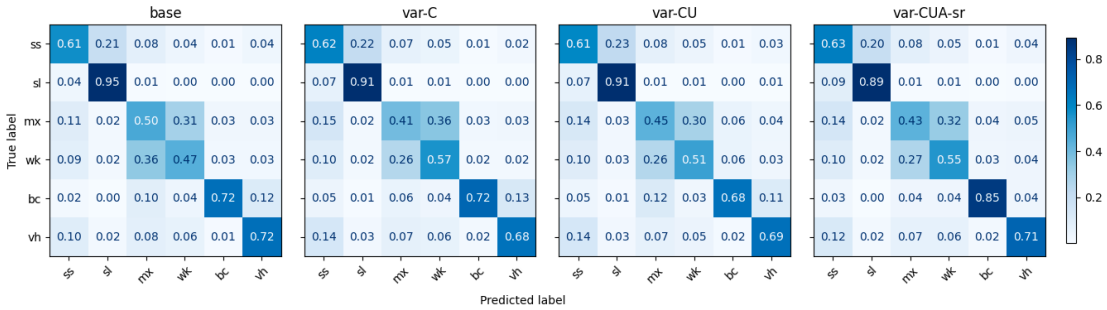
Table 6.2: BALANCED ACCURACY RESULTS FOR DIFFERENT MODELS. *This table shows the result of the balanced Accuracy of each model by test dataset SD^{test} and TD^{test}. Table 6.2a shows the overview of differences between the base and its variant. Note that the value of var-CUA for SD^{test} and TD^{test} is the same as the highest value in Table 6.2b for each. The highest accuracy for each case is highlighted in bold. And, the second and third are underlined.*

(a) for the base model and its variants

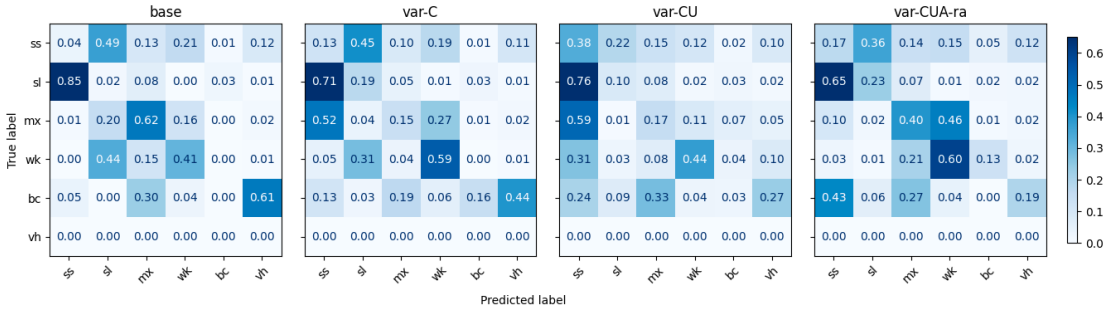
	base	***var-C***	***var-CU***	***var-CUA***
SD^{test}	<u>0.6638</u>	<u>0.6506</u>	0.6393	**0.6772**
TD^{test}	0.2174	<u>0.2448</u>	<u>0.2268</u>	**0.2805**

(b) all var-CUA cases

	var-CUA						
	s	***r***	***a***	***sr***	***sa***	***ra***	***sra***
SD^{test}	0.6710	<u>0.6754</u>	0.6383	**0.6772**	0.6621	<u>0.6765</u>	0.6723
TD^{test}	0.2188	0.2074	0.2588	0.2241	<u>0.2635</u>	**0.2805**	<u>0.2630</u>



(a) Confusion matrix of SD^{test}



(b) Confusion matrix of TD^{test}

Figure 6.1: CONFUSION MATRIX FOR DIFFERENT MODELS. *This figure shows the ratio of prediction labels by each true label. It means that the more dark-colored labels for each true label, the more likely they are predicted. Figure 6.1a and Figure 6.1b are derived from SD^{test} and TD^{test}, respectively. For the model var-CUA, 'sr' and 'ra', which showed the highest values in Table 6.2b, were chosen and arranged. The remaining var-CUA models are in Figure A.2. The name of the known class is abbreviated: 'sit-stand'→'ss', 'sleep'→'sl', 'mixed'→'mx', 'walking'→'wk', 'bicycle'→'bc', 'vehicle'→'vh'*

6.2.2 Balanced OSCR

In the case of open-set scenario experiments, we calculated the balaced OSCR for each model variation incorporating unknown samples. The findings presented in Table 6.3a indicate that selecting the appropriate augmentation technique enables the var-CUA model to attain a higher CCR at an FPR of 0.1 for both domains, reaching values of 0.5777 for the SD^{test} and 0.0843 for the TD^{test}. Comparing the two models, var-CU and var-CUA, that specifically take Open-Set Classification into account, the effect of augmentation techniques is remarkable. The addition of augmentation shows improvements of 13% and 54% for SD^{test} and TD^{test}, respectively. Specifically, in the Table 6.3b, we can observe the details of var-CUA performance. For the SD^{test}, the one that exhibits high performance is the var-CUA-s model followed by var-CUA-ra and var-CUA sr. On the other hand, for the TD^{test}, the one that exhibits high performance is the var-CUA-a model followed by var-CUA-sra and var-CUA sa. Those findings indicate that for both domains the single augmentation shows better performance than multi-augmentation.

Furthermore, the balanced OSCR curve in Figure 6.3 shows other insights into the balanced CCR of different models for each dataset. In particular, for the SD^{test} as depicted in Figure 6.3a, var-CUA is plotted based on the range between the augmentation techniques that presented minimum and maximum balanced CCR at each FPR, in this case, the var-CUA-a and var-CUA-s respectively. We can observe that all cases of var-CUA always performed better than var-CU and it is worth mentioning that the performance of var-CUA-s and var-CUA-a intersect with each other when FPR is 0.05. These two findings are represented in detail through the subplots *A* and *B* on the right side of the figure. Similarly, for the TD^{test} as depicted in Figure 6.3b var-CUA was presented in a range between the var-CUA-r and var-CUA-a. These are the augmentation techniques that presented a minimum and maximum balanced CCR corresponding to each FPR in TD^{test}. Moreover, it is noticeable that var-CUA-r always shows a higher *balanced CCR* compared to other models. However, other augmentation models show lower performance than even var-CU when FPR is 0.1. In particular, var-CUA-s, r, and sr show this trend continually even when FPR increases. These two findings are also represented in detail through the subplots *A* and *B* on the right side of the figure.

The OSCR curve shows the pair of values in FPR and CCR according to changes in Softmax. Therefore, it is possible to analyze the phenomenon of the OSCR curve from another perspective through the histogram of the known sample and the unknown sample by the value of Softmax having the range from 0 to 1. For this purpose, Figure 6.2 illustrates the histogram of known and unknown samples by the softmax score across different domains and model variations. The more the distribution between the unknown sample and the known sample is distinguished, the higher the chance of having a highly balanced CCR at a certain FPR. First of all, for the SD^{test} in Figure 6.2a, var-C presents a positive change in comparison to the base model since the overlaps between the histograms for known and unknown samples are reduced. On the other hand, as we see in TD^{test} (Figure 6.2b), the overlaps of known and unknown distributions tend to increase from the base to the var-C model. However, from the var-C to the var-CU and further to var-CUA, both domains not only present a noticeable decrease in the softmax scores of unknown samples but also overlap between the area of known and unknown. As introduced in 5.3, this seems to be the expected effect of using the unknown sample in the training dataset and applying the objectosphere loss function. As a result, this change in the fine-tuning process can be interpreted as playing an important role in improving the open-set classification performance for both cases, SD^{test} and TD^{test}.

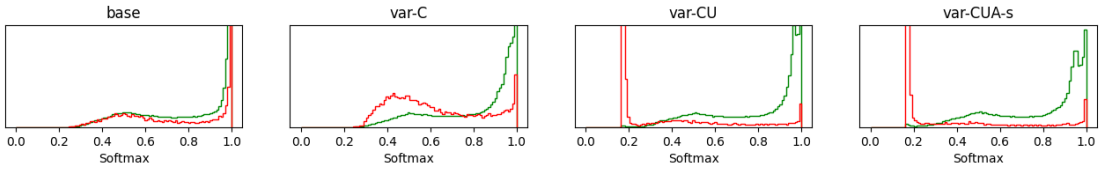
Table 6.3: BALANCED CCR RESULTS AT FPR OF 0.1 FOR DIFFERENT MODELS. *This table shows the result of the balanced CCR at FPR of 0.1. Table 6.3a shows the overview of differences between the base and its variant models. And, Table 6.3b shows the detailed result of each augmentation technique in the var-CUA model. The highest accuracy for each case is highlighted in bold. And, the second and third are underlined. Note that the value of var-CUA for SD^{test} and TD^{test} in Table 6.3a is the same as the highest value in Table 6.3b for each.*

(a) the base model and variants

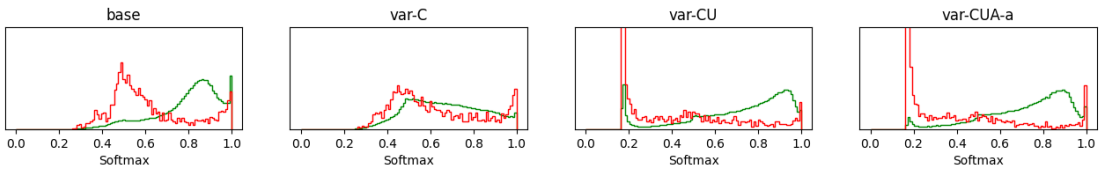
	base	***var-C***	***var-CU***	***var-CUA***
SD^{test}	0.0003	<u>0.2971</u>	<u>0.5106</u>	**0.5777**
TD^{test}	<u>0.0197</u>	0.0160	<u>0.0546</u>	**0.0843**

(b) all var-CUA cases

	var-CUA						
	s	***r***	***a***	***sr***	***sa***	***ra***	***sra***
SD^{test}	**0.5777**	0.5439	0.5124	<u>0.5446</u>	0.5427	<u>0.5482</u>	0.5444
TD^{test}	0.0379	0.0322	**0.0843**	0.0475	<u>0.0479</u>	0.0448	<u>0.0516</u>

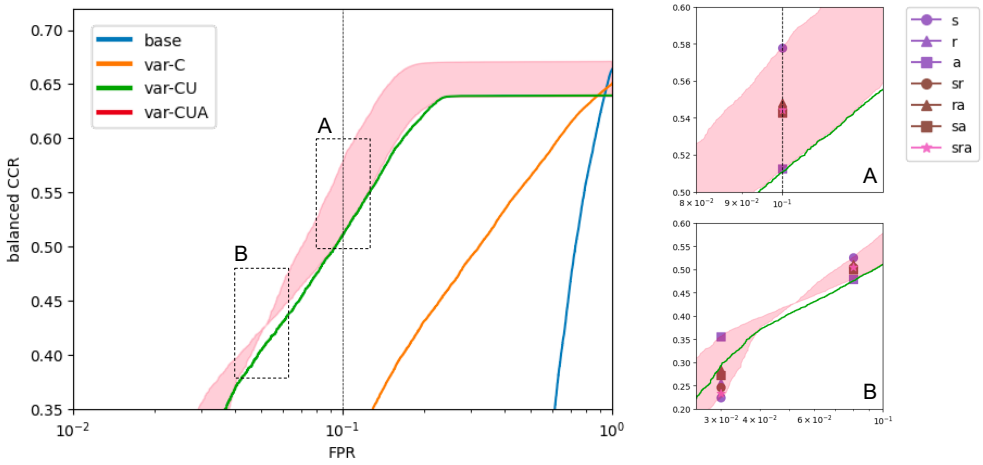


(a) histogram of SD^{test} softmax score

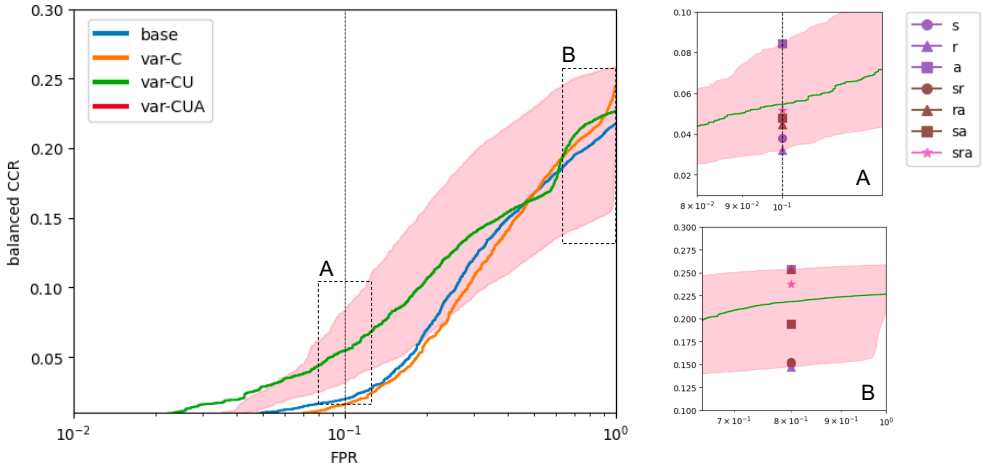


(b) histogram of TD^{test} softmax score

Figure 6.2: SOFTMAX SCORE HISTOGRAM BY KNOWN AND UNKNOWN SAMPLES. *This graph shows the distribution of the softmax score by known and unknown samples for the base and its variations. In each softmax, the density of the known sample and the unknown sample is expressed in green and red, separately. Figure 6.2a and Figure 6.2b represent the result from SD^{test} and TD^{test}, respectively. For the model var-CUA here, 's' is selected for SD^{test} and 'a' is selected for TD^{test} because these two augmentations are the highest balanced CCR in Table 6.3b for each dataset. The remaining var-CUA models are in Figure A.3 and the comparison between each known label vs. unknown is depicted in Figure A.4. Note that the lower area overlapped between the green and red, the easier distinction between known and unknown samples through softmax thresholding.*



(a) Source domain test dataset SD^{test}



(b) Target domain test dataset TD^{test}

Figure 6.3: BALANCED OSCR CURVE FOR DIFFERENT MODELS BY SOURCE AND TARGET. *This figure shows the values of the pairs of balanced CCR and FPR that vary with the softmax threshold. Figure 6.3a and Figure 6.3b represent the result from SD^{test} and TD^{test}, respectively. Each model is depicted with different colors. In particular, the model var-CUA is expressed in the form of a range with maximum and minimum values because various results can be obtained depending on which augmentation was used. In addition, specific ranges such as A and B are displayed as detailed plots on the right side of each main plot. Accordingly, it is possible to compare the balanced CCR value between various augmentation types belonging to the var-CUA and the surrounding values.*

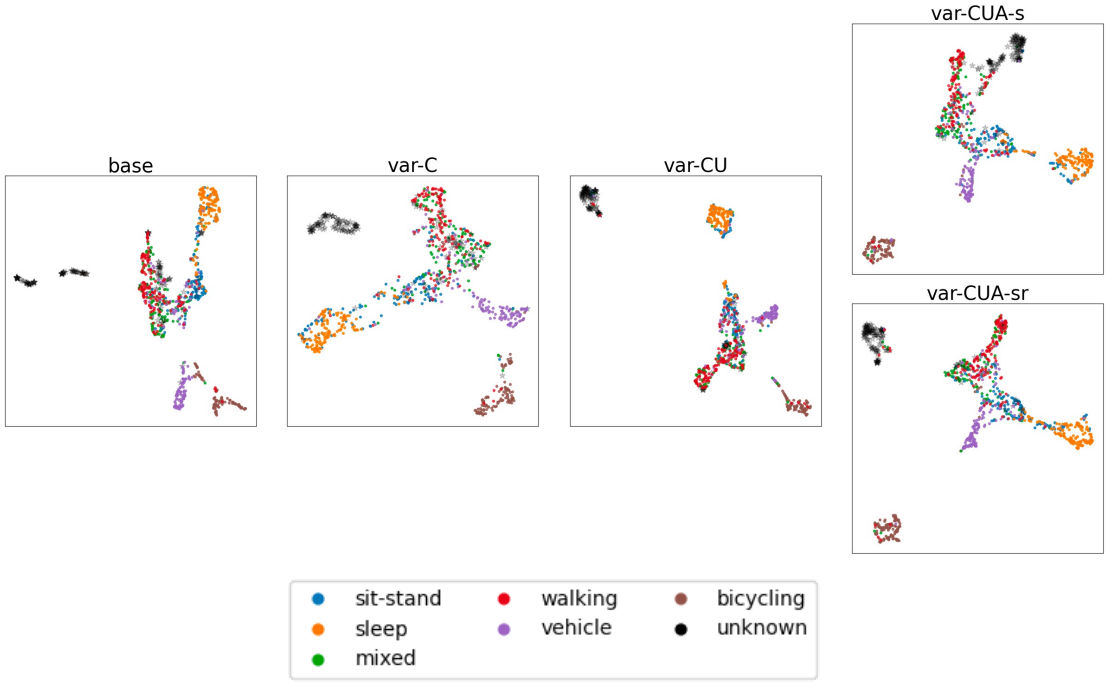
6.2.3 Semantic Analysis in Deep Feature Space

Aside from quantitative analysis, we have chosen to derive also qualitative insights from the feature space visualization, particularly on semantic overlaps in the feature space.

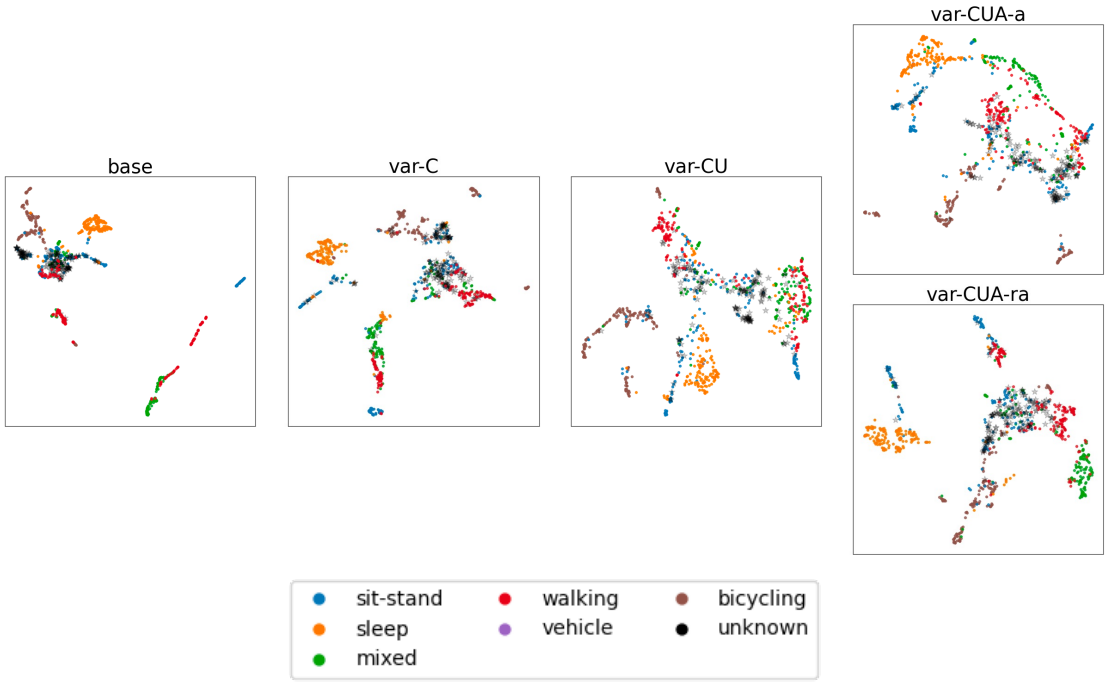
Starting with analyzing SD^{test}, Figure 6.4a highlights a clear separation of *Unknown* features from other classes in all models. Furthermore, as models progress from base to var-CUA-sr, the distance between the cluster of *Unknown* cases and those of known samples increases. In other words, as the finetuning process is newly defined, this phenomenon supports that the values of features extracted from the *Unknown* samples have a different distribution from the feature values of other known samples.

Let's take a closer look at the distribution of known samples in Figure 6.4a. It is worth mentioning that *Bicycling* stands out as distinctly clustered apart from others, followed by *Vehicle* and *Sleep* showing similar phenomena. In particular, the distinction between *Bicycling* and *Vehicle* grows as the model's enhancements advance from the base to the var-CUA model. This can also be seen as a situation in which the *Vehicle* cluster absorbs into the larger central cluster, which includes *Sit-Stand* and *Mixed*. However, unlike *Vehicle*, *Walking* shows a phenomenon of increasing separation from the central cluster. As a result, as the model evolves, the *Unkown, Bicycle, Sleep, Walking* classes have more and more clear features, while the features of *Sit-Stand, Vehicle, Mixed* seem to become more ambiguous.

Interestingly, in the feature space visualization using TD^{test}, it can be seen that the overlap between classes is more prominent. As illustrated in Figure 6.4b, the deep feature space for TD^{test} shows that the *Unknown* samples are centrally positioned among other clusters, causing overlaps with various samples. Notably, in the TD^{test} domain, there's a marked overlap between features attributed to the *Unknown* and *Sit-stand* categories. Furthermore, the *Walking* category exhibits a dispersed clustering pattern, appearing adjacent to multiple other clusters rather than forming a distinct group. This dispersion is consistently observed across all models, with the *Walking* cluster notably aligning closely with the *Mixed* cluster. Looking at the change according to the model, it can be seen that the boundaries between classes become ambiguous when changing from var-C to var-CU. However, it can be seen that the boundaries become clear again when changing from var-CU to var-CUA.



(a) The distribution of 1000 source test samples in the feature space of each model



(b) The distribution of 1000 target test samples in the feature space of each model

Figure 6.4: VISUALIZATION OF THE DISTRIBUTION OF SAMPLES IN THE DEEP FEATURE SPACE. *This figure visualizes the distribution of test samples from SD^{test} and TD^{test}. To prevent visual bias caused by data imbalance, 1000 samples were randomly selected and plotted for each label. For the var-CUA models, the models that have the highest performance in closed-set and open-set are chosen by each dataset. The remaining var-CUA models are in Figure A.1. Due to the nature of UMAP used for dimensional reduction, the axis scale and axis values are not visualized because they have no specific meaning. Only relative distances between samples or clusters in the same plot matter.*

Chapter 7

Discussion

This project begins with a fundamental question: *'How to tackle the model generalization issue of the HAR task by enhancing the fine-tuning process while preserving the pre-trained model?'*. This inquiry is pivotal as it aims to ensure that we can build models for Human Activity Recognition (HAR) that accurately classify activities regardless of the device used, its body location, or the surrounding environment. The significance of preserving the pre-trained model lies in the fact that these models are built upon vast amounts of data and computational resources, which might not be readily available for all projects. By focusing on fine-tuning these models with smaller, domain-specific datasets, we aim to leverage the extensive learning and insights they contain. This approach underscores the efficiency and necessity of exploring advanced fine-tuning techniques. It represents a strategic method to enhance model generalization and performance across diverse conditions, making it an essential step towards harnessing the full potential of pre-trained models for HAR tasks.

Our initial analysis, using diverse evaluation metrics, uncovered a notable performance drop in the TD^{test} compared to the SD^{test} with the base model. This decline emphasizes the significant impact of domain-specific factors, such as device type and environmental settings, on sensor signal characteristics. Despite leveraging a large dataset of 700,000 activity samples, for the creation of the pretraining model, these domain differences posed challenges to the MTSSL model's conventional fine-tuning process. In addressing these issues, we systematically improved the model from the base model to the var-CUA model, leading to significant enhancements in cross-domain performance. This demonstrates how methodical refinements to the fine-tuning approach can be applied with successful outcomes while preserving the same pre-trained model.

Impact of Enhanced MTSSL Fine-Tuning Process

Initiating our enhancement with var-C, we integrated 22 classical features to broaden domain coverage. This inclusion, as detailed in Table 6.2, yielded a performance uplift in TD^{test}, enhancing the baseline by roughly 0.03 (12%). This was evident in Figure 6.1b, where activities such as *Walking* saw improved classification accuracy in both SD^{test} and TD^{test}. This success, however, did not uniformly extend to other activities in TD^{test} that presented an increased rate of misclassification as *Sit-Stand*, which shows high overlaps with other activities in the semantic analysis. This result is potentially derived from the relabeling process of original datasets involved in TD^{test}. For instance, in the dataset ADL which is a part of TD^{test}, the relabeling process showcases the phenomenon of semantic overlap. Following the Table A.7a, for example, activities like *Sit Down Chair* and *Stand Up Chair*, which have a transitional nature, were both labeled as *Sit-Stand*. However, this caused some confusion, especially with activities like

Getup Bed, which is also a transitional nature. As it was not stated clearly sit or stand it was labeled as *Unknown*. This approach, while intended to make things clearer, actually made it harder to differentiate between some activities, illustrating the challenge of labeling transitional movements accurately.

Following this, with var-CU, we assessed the model's capacity to distinguish false positives through the inclusion of unknown samples. As depicted in Figure 6.2, this addition led to a reduced overlap of histogram regions representing known and unknown samples. That is also evident in Figure 6.4, which demonstrates a more precise feature space clustering for the *Unknown* category in SD^{test}, suggesting improved identification of *Unknown* cases. Despite this advancement, balanced Accuracy metrics indicated a reduction compared to var-C, pointing to a trade-off in classifying known cases. This likely results from our customized weighted loss function's focus on minimizing false positives by correctly identifying unknown samples, a strategy that necessitates a more cautious approach in classifying known classes, thereby slightly reducing precision. Yet, a detailed examination of the balanced OSCR in (Table 6.3) showed var-CU significantly improved balanced CCR across both domains at a fixed FPR of 0.1, outperforming both base and var-C models. Notably, in TD^{test}, balanced CCR with var-CU was more than threefold higher than with var-C, highlighting var-CU's effectiveness in open-set scenarios by better filtering out unknown samples, thereby enhancing generalization. This distinction in performance indicates that although var-CU experiences a decrease in balanced Accuracy, it significantly excels in open-set classification. This is particularly true for TD^{test}, where var-CU's ability to effectively distinguish unknown samples marks a notable advancement, underscoring its benefits in scenarios where identifying unknown samples is critical. Thus, var-CU's advantage is most pronounced in situations that demand accurate identification of unknowns, making it particularly beneficial for TD^{test} compared to TD^{test}.

Lastly, we explored the impact of multiple augmentation techniques with var-CUA, which outperformed previous models in balanced Accuracy across both domains, as detailed in Table 6.2. In particular, var-CUA for the majority of the augmentation technique combinations showed better closed-set classification results compared to other variant models for both SD^{test} and TD^{test}. Notably, for the appropriate augmentation technique var-CUA achieved an increase in balanced Accuracy by approximately 0.06 (29%) in TD^{test} and 0.01 (2%) in SD^{test} when compared to their respective base models. This suggests that augmentation techniques not only improve the correct classification of known samples but also enhance the model's generalizability. Importantly, these augmentation techniques effectively mitigate performance drops that came from the model var-CU. Furthermore, balanced OSCR measurements (Table 6.3) indicate that var-CUA models exhibit a substantial increase in balanced CCR at a fixed FPR of 0.1. In particular, the var-CUA model with the appropriate augmentation techniques presents an overall increase of 13% in SD^{test} and an overall increase of 54% in TD^{test} from their respective var-CU model. Those observations highlight the model's enhanced capability to accurately identify both known and unknown activities, essential for maintaining efficacy in real-world applications.

Diving deeper into our analysis, we found that different augmentation techniques in the var-CUA model affected performance in complex ways. While most techniques improved results, a few made performance slightly worse. This shows how tricky it can be to pick the right augmentation strategy: some work better for certain types of tests, like open-set or closed-set classifications. For example, the augmentation with switching axes consistently improved results in SD^{test} for both closed- and open-set scenarios. On the other hand, the augmentation with amplitude scaling was particularly helpful in TD^{test}, indicating that it is better suited for more complicated contexts. This highlights the importance of choosing the right augmentations to best suit the specific needs of different tests, ensuring our model performs well in a variety

of situations.

Limitations

Despite the progress made with the var-C, var-CU, and var-CUA models in enhancing model performance and generalizability, there are limitations to our approach. Following we outline these limitations and suggest directions for future research:

- Since the domain had to be defined between the available datasets, this project proceeded with a cross-domain evaluation under a limited domain definition of a combination of device type and data collection environment setting. While the experiment within this predefined domain has provided valuable perspectives on the challenges and potential solutions associated with cross-domain evaluations, it is expected that a deeper consideration of the domain definition and specifically customized dataset for Cross-Domain Evaluation will help us understand this problem more accurately.

- An additional limitation of this project stems from the presence of imbalanced labels within the dataset. While we implemented a weighted loss function and correspondingly adjusted evaluation metrics to mitigate the impact of this imbalance, it is important to acknowledge that no approach can provide foolproof prevention of biased model training or ensure optimal performance. A dataset with a good balance between labels of each domain's dataset and between known and unknown samples is required. In cases where obtaining such datasets proves challenging in practice, the necessity for more sophisticated weighting strategies in both loss functions and evaluation metrics becomes apparent.

- The datasets utilized in this study varied in the level of detail provided in their label descriptions. Consequently, our relabeling process relied solely on the original names of each label. This approach, while straightforward, inadvertently led to frequent overlaps in the feature space, as activities with different practical implications were often relabeled into the same class. This issue, partly stemming from label ambiguity as highlighted in Willets2018, posed significant challenges in training our classifier model. Such limitations became apparent in the confusion matrix and in the overlaps observed in the feature space representations. These findings underscore the need for revisiting the criteria for defining known labels and suggest the potential benefits of a more refined relabeling process that more accurately accounts for semantic differences between activities.

- In this project, we employed the Multi-Task Self-Supervised learning model as the foundational framework, enhancing the fine-tuning process based on this model. However, this particular setup can not represent the overall cross-domain evaluation of Self-Supervised learning in HAR models. Notably, Haresamudram et al. (2022) shows various types of Self-Supervised HAR models. Exploring cross-domain evaluation with different types of Self-Supervised models presents an intriguing avenue for future investigation. Additionally, there is potential to identify common strategies that enhance the generalization performance of Self-Supervised learning models across diverse domains.

- The objective of this project was to explore diverse avenues to enhance cross-domain performance, prioritizing comprehensive strategies within each method rather than delving into the intricacies of a single method. Consequently, when incorporating the objectsphere loss into the var-CU model, the detailed exploration of various hyperparameters was intentionally omitted. The possibility remains that the identification of an optional hyperparameter could yield performance enhancements.

- For the variant model *var-CUA*, we focused on implementing 3 different augmentation techniques in this project. However, these data augmentation techniques for HAR tasks can be varied. By comparing the effect of each method presented in Um et al. (2017), it is expected that the effect of data augmentation on cross-domain performance can be analyzed from more diverse perspectives.

Chapter 8

Conclusion

Our investigation into enhancing the generalization capabilities of self-supervised Human Activity Recognition (HAR) models across diverse datasets has led to several discoveries. By incorporating classic human activity features, integrating unknown samples, and employing a strategic mix of data augmentation techniques, we've addressed the inherent complexity and variability of human activities across different domains. This multi-faceted approach not only improved the base model, Multi-Task Self-Supervised Learning (MTSSL)'s performance, but also highlighted the importance of a nuanced fine-tuning process for effective cross-domain application.

A notable outcome of this research is the clear demonstration that no single enhancement method suffices on its own. Instead, the synergy among the introduced methodologies—classic feature integration, unknown sample inclusion, and data augmentation—collectively contributed to a marked improvement in model performance. These methods, particularly when combined in the var-CUA model, significantly elevated both the balanced Accuracy and the model's robustness across diverse datasets. This underscores the pivotal role of tailored fine-tuning strategies in overcoming the challenges posed by the variability of activities and environmental conditions in HAR tasks.

Despite these advancements, our exploration acknowledges the persisting challenges of dataset variability and environmental diversity as substantial hurdles in HAR model generalization. This research journey has systematically assessed the impact of each fine-tuning method, laying a solid groundwork for future exploration. As we move forward, it's clear that continuous refinement and exploration of fine-tuning techniques are essential in enhancing the adaptability and accuracy of HAR models, paving the way for their application in a wider array of real-world scenarios.

Appendix A

Attachments

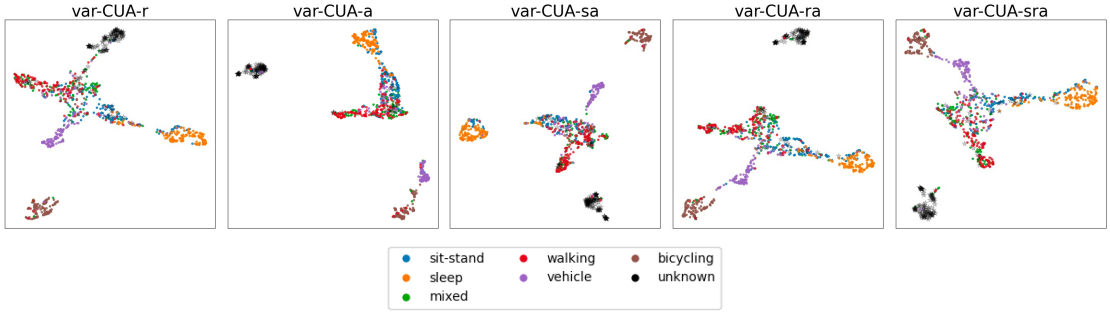
A.1 Dataset Details

Dataset	Total	Willetts2018						unknown
		Sit-Stand	*Sleep*	*Mixed*	walking	vehicle	bicycling	
CAPTURE24	909,535	353,443 (39%)	340,076 (37%)	115,586 (13%)	57,239 (6%)	34,380 (4%)	8,811 (1%)	-
PAMAP2	5,250	1,366 (26%)	385 (7%)	870 (17%)	1,282 (24%)	109 (2%)	329 (6%)	909 (17%)
GOTOV	18,873	4,922 (26%)	2,708 (14%)	1,370 (7%)	4,451 (24%)	-	2,736 (14%)	2,686 (14%)
REALWORLD	13,109	3,764 (29%)	1,896 (14%)	2,066 (16%)	5,114 (39%)	-	-	269 (2%)
SELFBACK	9,642	2,359 (24%)	1,182 (12%)	1,569 (16%)	4,532 (47%)	-	-	-
ADL	1,594	572 (36%)	-	167 (10%)	558 (35%)	-	-	297 (19%)
WISDM	29,136	13,697 (47%)	-	6,880 (24%)	3,420 (12%)	-	-	5,139 (18%)
HARVARDLEO	3,089	800 (26%)	-	1,200 (39%)	692 (22%)	-	-	397 (13%)
MENDELEYDAILY	3,644	1,064 (29%)	-	521 (14%)	824 (23%)	-	-	1,235 (34%)
PAAL	8,871	3,837 (43%)	-	3,004 (34%)	-	-	-	2,030 (23%)
COMMUTING	5,545	1,712 (31%)	-	2,319 (42%)	-	-	-	1,514 (27%)
OPPO	3,882	2,931 (76%)	165 (4%)	-	786 (20%)	-	-	-
FORTH-TRACE	6,139	2,723 (44%)	-	-	3,285 (54%)	-	-	131 (2%)
HOUSEHOLDHU	93,065	21,224 (23%)	-	20,838 (22%)	-	-	-	51,003 (55%)
WRISTPPG	1,327	-	-	292 (22%)	437 (33%)	-	598 (45%)	-
NEWCASTLE	365,776	-	365,776 (100%)	-	-	-	-	-
ICHI14	3,488	-	3,488 (100%)	-	-	-	-	-

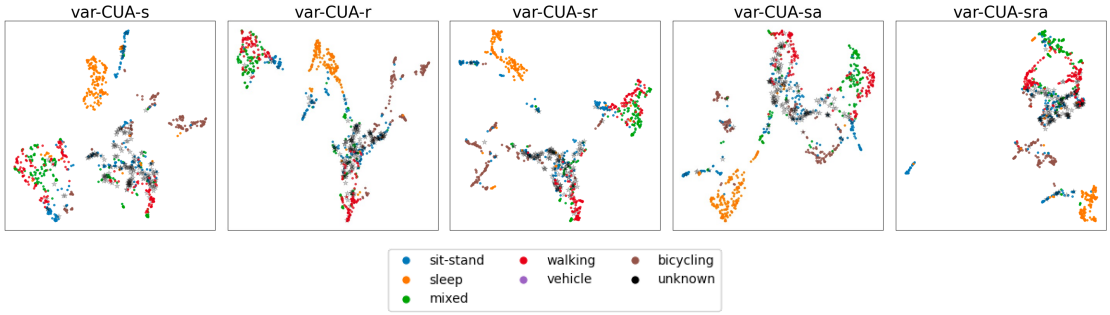
Table A.1: THE COUNT OF LABELS FOR EACH DATASET. *This table describes the counts and percentile of each label after overall dataset preprocessing.*

A.2 Additional results of experiment

In this section, we attached additional results of our experiment. This includes detailed evaluation results skipped from the main contents in Section 6.2.

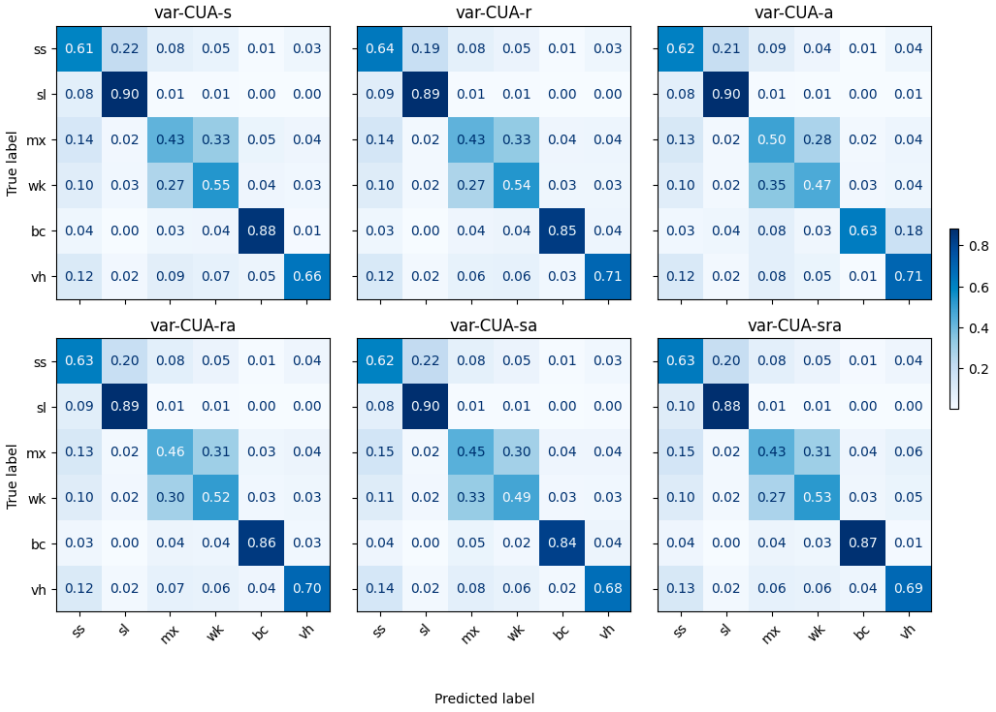


(a) The distribution of 1000 source test samples in the feature space of each model

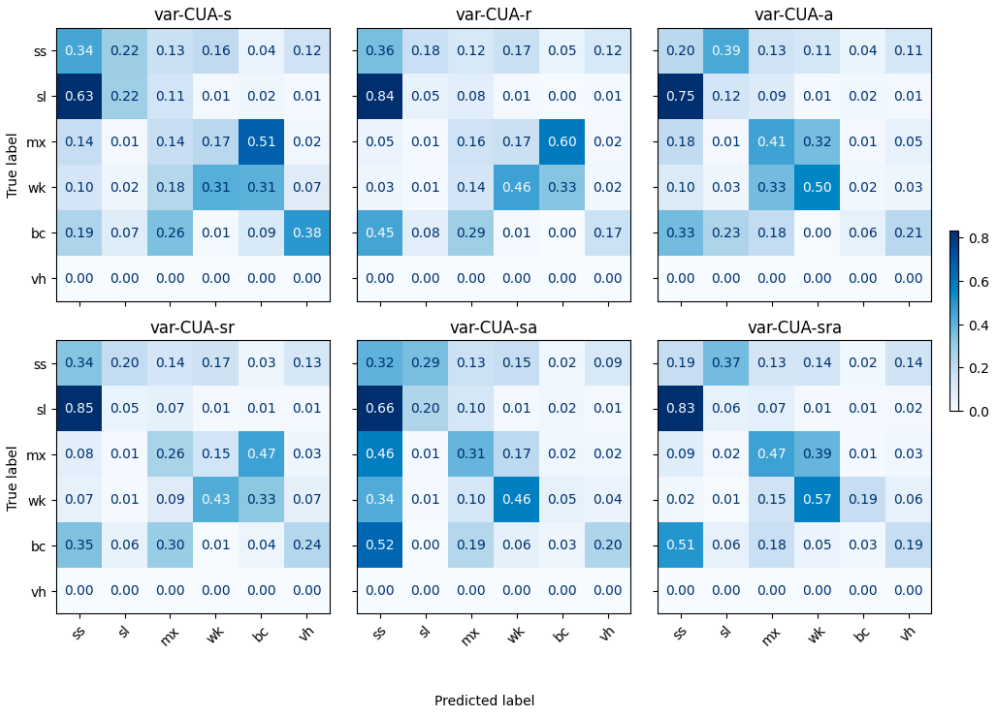


(b) The distribution of 1000 target test samples in the feature space of each model

Figure A.1: VISUALIZATION OF THE DISTRIBUTION OF SAMPLES IN THE DEEP FEATURE SPACE FOR REMAINING VAR-CUA MODELS. *To prevent visual bias caused by data imbalance, 1000 samples were randomly selected and plotted for each label. Due to the nature of UMAP used for dimensional reduction, the axis scale and axis values are not visualized because they have no specific meaning. Only relative distances between samples or clusters in the same plot matter.*

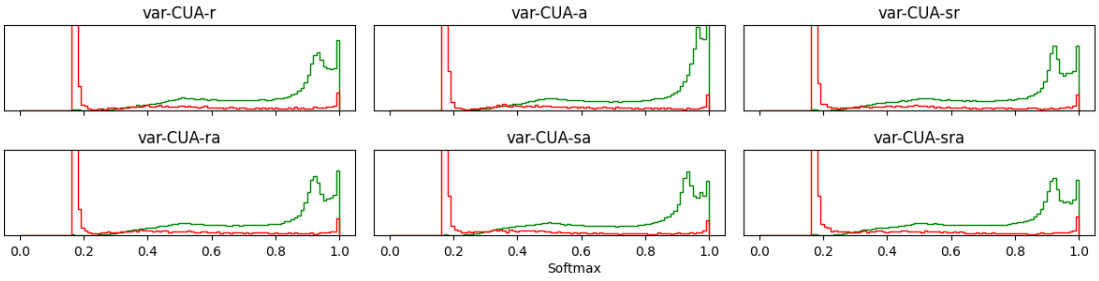


(a) Confusion matrix of SD^{test}

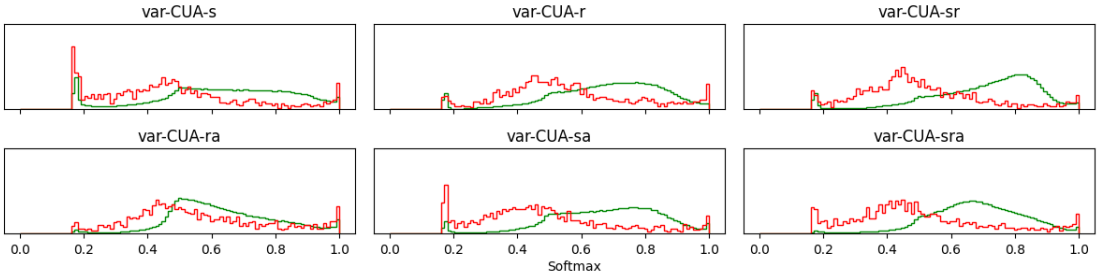


(b) Confusion matrix of TD^{test}

Figure A.2: CONFUSION MATRIX FOR REMAINING VAR-CUA MODELS. *The name of the known class is abbreviated: 'sit-stand'→'ss', 'sleep'→'sl', 'mixed'→'mx', 'walking'→'wk', 'bicycle'→'bc', 'vehicle'→'vh'*

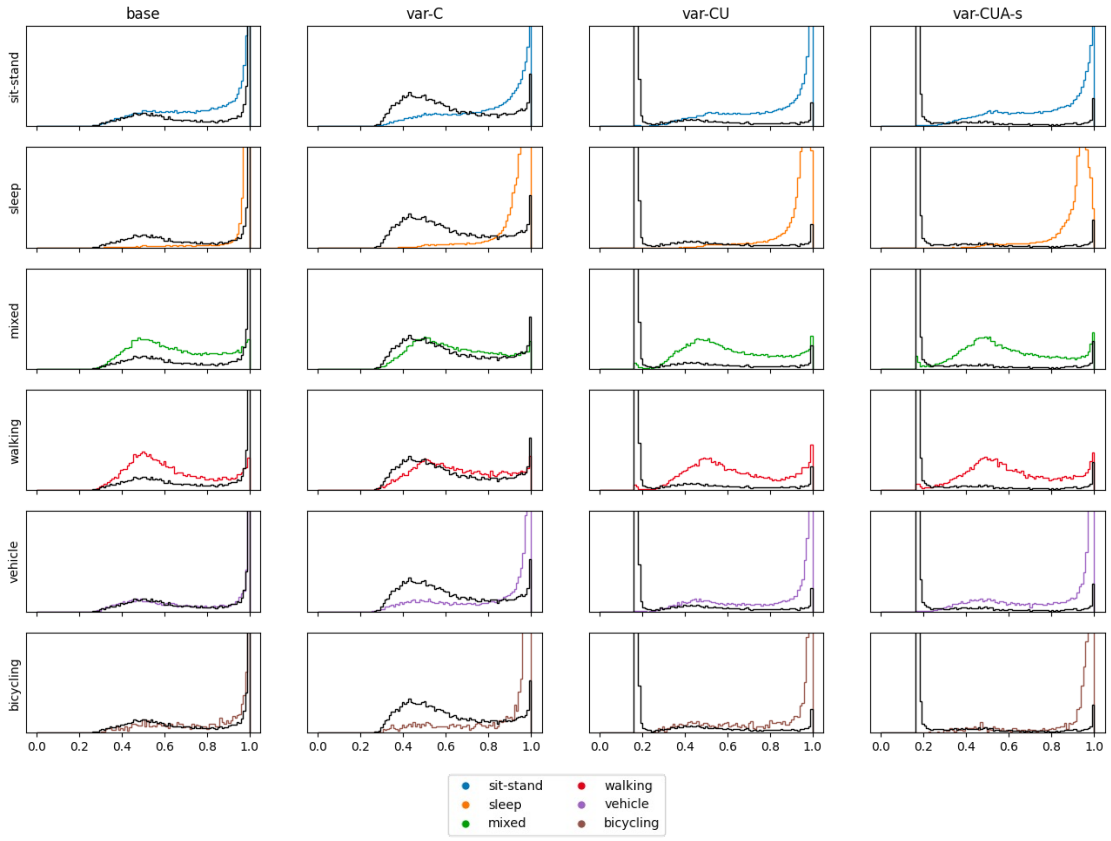


(a) histogram of SD^{test} softmax score

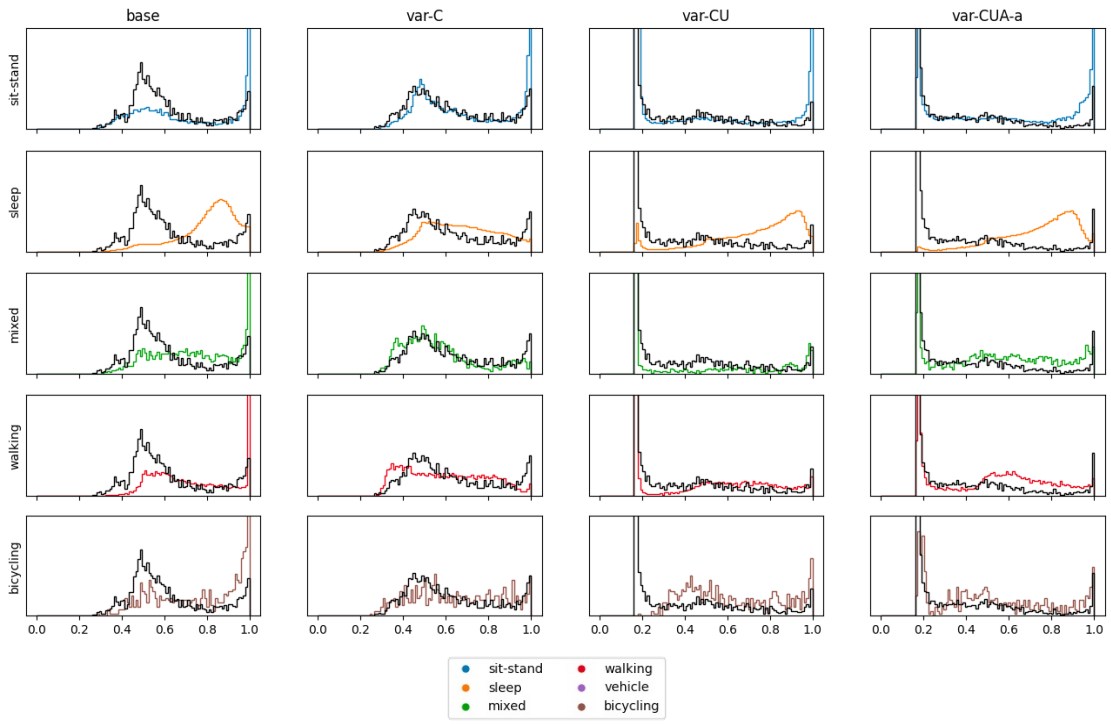


(b) histogram of TD^{test} softmax score

Figure A.3: SOFTMAX SCORE HISTOGRAM BY KNOWN AND UNKNOWN SAMPLES FOR REMAINING VAR-CUA MODELS. *In each softmax, the density of the known sample and the unknown sample is expressed in green and red, separately.*



(a) histogram of SD^{test} softmax score



(b) histogram of TD^{test} softmax score

Figure A.4: SOFTMAX SCORE HISTOGRAM BY EACH KNOWN LABEL AND UNKNOWN SAMPLES. *Unknown samples are expressed in black color.*

A.3 Example of the relabeling process

In this section, we illustrate the relabeling process applied to the SELFBACK (Table A.6a) and PAAL (Table A.10) datasets, showcasing how Willetts2018 label type has been consistently implemented. For example, activities such as *Lying*, *Sitting*, and *Standing* are uniformly relabeled across both datasets to *Sleep* and *Sit-Stand*. This uniformity is in line with Willetts' categorization within the Capture24 dataset, ensuring coherence in labeling across different datasets.

In relabeling the PAAL dataset, we differentiate notably between *Mixed* and the newly added *Unknown* categories. *Mixed* includes activities like *Washing hands* from PAAL, recognized in Capture24 but not aligning with the other 5 specific labels (e.g. *Sleep*, *Walking*, etc.). These are activities acknowledged within the dataset but lacking a precise classification.

Conversely, *Unknown* is introduced for activities beyond the Willetts framework and Capture24's scope, such as *Salute*, *Sneeze cough*, and *Blow nose* from PAAL. These do not fit any existing categories, marking them as *Unknown* and expanding our dataset's range of activities. This distinction enriches our understanding of human behaviors, addressing previously unrecognized activities in the research.

After outlining the relabeling process in our datasets, it is essential to emphasize the diverse range of activities in both the source and target domain datasets, specifically those categorized as *Unknown*. Table 4.4 illustrates these unique activities, highlighting the non-overlapping nature crucial for effective cross-domain evaluation. This diversity ensures that our fine-tuning phase encompasses a broad behavioral spectrum, distinct from the patterns tested in cross-domain evaluations.

Table A.2: RELABELING RESULTS FOR THE DATASET 'PAMAP2' AND 'WISDM'.

(a) PAMAP2 Dataset

PAMAP2	
Original Label	Capture24 Label
lying	*Sleep*
sitting	sit-stand
standing	
walking	walking
cycling	bicycling
car driving	vehicle
running	*Mixed*
ironing	unknown
folding laundry	
vacuum cleaning	
house cleaning	
playing soccer	
rope jumping	

(b) WISDM Dataset

WISDM	
Original Label	Capture24 Label
sitting	sit-stand
standing	
typing	
soup	
chips	
pasta	
sandwich	
writing	
walking	walking
stairs	
jogging	mixed
teeth	
drinking	
folding	
kicking	unknown
catch	
dribbling	
clapping	

Table A.3: RELABELING RESULTS FOR THE DATASET 'HARVARDLEO' AND 'HOUSEHOLDHU'.

(a) HARVARDLEO Dataset

HARVARDLEO	
Original Label	Capture24 Label
Keyboard_Writing	sit-stand
Laptop	
Handwriting	
Eating	
Downstairs	walking
Walking	
Walking_Fast	
Upstairs_Fast	
Upstairs	
Handwashing	mixed
Facewashing	
Teethbrush	
Sweeping	
Dusting	
Rubbin	
Relax	unknown
Vacuuming	

(b) HOUSEHOLDHU Dataset

HOUSEHOLDHU	
Original Label	Capture24 Label
Keyboard typing	sit-stand
Handwriting	
Wiping the table	mixed
Sweeping floor	
Using mouse	unknown
Cutting vegetables	
Stir-frying vegetables	
Using vacuum to vacuum	
Open and close drawer	

Table A.4: RELABELING RESULTS FOR THE DATASET 'OPPORTUNITY' AND 'WRISTPPG'.

(a) OPPORTUNITY Dataset

OPPORTUNITY	
Original Label	Capture24 Label
lie	*Sleep*
sit	sit-stand
stand	
walk	walking

(b) WRISTPPG Dataset

WRISTPPG	
Original Label	Capture24 Label
walk	walking
high_resistance_bike	bicycling
low_resistance_bike	
run	*Mixed*

Table A.5: RELABELING RESULTS FOR THE DATASET 'COMMUTING' AND 'MENDELEY-DAILY'.

(a) COMMUTING Dataset

COMMUTING	
Original Label	Capture24 Label
computer	sit-stand
dinner	
lunch	
work	
shopping	walking
brush_teeth	mixed
shower	
exercise	
commuting	unknown

(b) MENDELEYDAILY Dataset

MENDELEYDAILY	
Original Label	Capture24 Label
BrushTeeth	sit-stand
DrinkGlass	
StandUp	
SitDown	
Walk	walking
CleanTable	*Mixed*
PourWater	unknown
CloseDoor	
OpenDoor	

Table A.6: RELABELING RESULTS FOR THE DATASET 'SELFBACK' AND 'REALWORLD'.

(a) SELFBACK Dataset

SELFBACK	
Original Label	Capture24 Label
Lying	*Sleep*
Standing	sit-stand
Sitting	
Walking Upstairs	walking
Walking Downstairs	
Walking in slow pace	
Walking in medium pace	
Walking in fast pace	
Jogging	*Mixed*

(b) REALWORLD Dataset

REALWORLD	
Original Label	Capture24 Label
lying	*Sleep*
standing	sit-stand
sitting	
climbingdown	walking
climbingup	
walking	
running	*Mixed*
jumping	unknown

Table A.7: RELABELING RESULTS FOR THE DATASET 'ADL' AND 'FORTH-TRACE'.

(a) ADL Dataset

ADL	
Original Label	Capture24 Label
comb_hair	sit-stand
drink_glass	
standup_chair	
sitdown_chair	
eat_meat	
eat_soup	
use_telephone	
climb_stairs	walking
descend_stairs	
walk	
brush_teeth	*Mixed*
getup_bed	unknown
liedown_bed	
pour_water	

(b) FORTH-TRACE Dataset

FORTH-TRACE	
Original Label	Capture24 Label
stand	sit-stand
sit	
sit and talk	
stand ->sit	
sit ->stand	
stand ->sit and talk	
sit and talk ->stand	
walk	walking
walk and talk	
climb stairs	
climb stairs and talk	
climb stairs ->walk	
climb stairs and talk ->walk and talk	
stand ->walk	unknown
walk ->stand	
stand ->climb stairs	
stand ->climb stairs and talk	

Table A.8: RELABELING RESULTS FOR THE DATASET 'ICHI14' AND 'NEWCASTLE'.

(a) ICHI14 Dataset

ICHI14	
Original Label	Capture24 Label
Sleep	*Sleep*

(b) NEWCASTLE Dataset

NEWCASTLE	
Original Label	Capture24 Label
Sleep	*Sleep*

Table A.9: RELABELING RESULTS FOR THE DATASET 'GOTOV'.

GOTOV	
Original Label	Capture24 Label
lyingDownLeft lyingDownRight	sleep
standing sittingSofa sittingCouch sittingChair	sit-stand
step walkingStairsUp walkingSlow walkingNormal walkingFast	walking
cycling	bicycling
dishwashing	*Mixed*
syncJumping stakingShelves vacuumCleaning	unknown

Table A.10: RELABELING RESULTS FOR 'PAAL'.

PAAL	
Original Label	Capture24 Label
writing	sit-stand
type_on_a_keyboard	
brush_teeth	
brush_hair	
drink_water	
phone_call	
eat_meal	
sit_down	
stand_up	
washing_dishes	mixed
ironing	
washing_hands	
dusting	
open_a_box	
put_on_a_shoe	unknown
put_on_a_jacket	
take_off_a_jacket	
take_off_a_shoe	
blow_nose	
put_on_glasses	
open_a_bottle	
salute	
sneeze_cough	
take_off_glasses	

List of Figures

List of Tables

Bibliography

Aittasalo, M., Vähä-Ypyä, H., Vasankari, T., Husu, P., Jussila, A.-M., and Sievänen, H. (2015). Mean amplitude deviation calculated from raw acceleration data: a novel method for classifying the intensity of adolescents' physical activity irrespective of accelerometer brand. *BMC sports science, medicine and rehabilitation*, 7:1–7.

Banos, O., Garcia, R., Holgado-Terriza, J. A., Damas, M., Pomares, H., Rojas, I., Saez, A., and Villalonga, C. (2014). mhealthdroid: a novel framework for agile development of mobile health applications. In *Ambient Assisted Living and Daily Activities: 6th International Work-Conference, IWAAL 2014, Belfast, UK, December 2-5, 2014. Proceedings 6*, pages 91–98. Springer.

Bao, L. and Intille, S. S. (2004). Activity recognition from user-annotated acceleration data. In *International conference on pervasive computing*, pages 1–17. Springer.

Bin Morshed, M., Saha, K., De Choudhury, M., Abowd, G. D., and Plötz, T. (2020). Measuring self-esteem with passive sensing. In *Proceedings of the 14th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pages 363–366.

Borazio, M., Berlin, E., Kucukyildiz, N., Scholl, P., and Van Laerhoven, K. (2014). Towards benchmarked sleep detection with wrist-worn sensing units. In *2014 IEEE International Conference on Healthcare Informatics*, pages 125–134. IEEE.

Bruno, B., Mastrogiovanni, F., and Sgorbissa, A. (2014). Dataset for ADL Recognition with Wrist-worn Accelerometer. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PC99.

Bulling, A., Blanke, U., and Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, 46(3).

Chan Chang, S., Walmsley, R., Gershuny, J., Harms, T., Thomas, E., Milton, K., Kelly, P., Foster, C., Wong, A., Gray, N., Haque, S., Hollowell, S., and Doherty, A. (2021). Capture-24: Activity tracker dataset for human activity recognition.

Climent i Pérez, P., Muñoz-Antón, , Poli, A., Spinsante, S., and Flórez-Revuelta, F. (2022). Dataset of acceleration signals recorded while performing activities of daily living. *Data in Brief*, 41:107896.

Dhamija, A. R., Günther, M., and Boult, T. E. (2018). Reducing network agnostophobia.

Doherty, A., Smith-Byrne, K., Ferreira, T., Holmes, M. V., Holmes, C., Pulit, S. L., and Lindgren, C. M. (2018). Gwas identifies 14 loci for device-measured physical activity and sleep duration. *Nature communications*, 9(1):5257.

Garcia, E. (2014). Dataset long-term activities. *Data*.

Gershuny, J., Harms, T., Doherty, A., Thomas, E., Milton, K., Kelly, P., and Foster, C. (2020). Testing self-report time-use diaries against objective instruments in real time. *Sociological Methodology*, 50(1):318–349.

Gjoreski, H., Bizjak, J., Gjoreski, M., and Gams, M. (2016). Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer. In *Proceedings of the IJCAI 2016 Workshop on Deep Learning for Artificial Intelligence, New York, NY, USA*, volume 10, page 970.

Haresamudram, H., Essa, I., and Plötz, T. (2022). Assessing the state of self-supervised human activity recognition using wearables.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer.

Hu, Z., Zhang, Y., and Pan, S. (2022). Multimodal fine-grained human activity dataset.

Jarchi, C. (2017). Description of a database containing wrist ppg signals recorded during physical exercise with both accelerometer and gyroscope measures of motion. *Data*, 2(1).

Karagiannaki, K., Panousopoulouand, A., and Tsakalides, P. (2016). The FORTH-TRACE dataset for human activity recognition of simple activities and postural transitions using a Body Area Network. DOI:10.5281/zenodo.841301.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Leotta, M., Fasciglione, A., and Verri, A. (2021). Daily Living Activity Recognition Using Wearable Devices: A Features-rich Dataset and a Novel Approach.

Lu, W., Chen, Y., Wang, J., and Qin, X. (2021). Cross-domain activity recognition via substructural optimal transport. *Neurocomputing*, 454:65–75.

McInnes, L., Healy, J., and Melville, J. (2020). Umap: Uniform manifold approximation and projection for dimension reduction.

Morshed, M. B., Saha, K., Li, R., D'Mello, S. K., De Choudhury, M., Abowd, G. D., and Plötz, T. (2019). Prediction of mood instability with passive sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–21.

Paraschiakos, S., , B. M. M., , K. A. A., , C. R. R., and , S. P. E. (2021). Gotov human physical activity and energy expenditure dataset on older individuals.

Qin, X., Wang, J., Chen, Y., Lu, W., and Jiang, X. (2022). Domain generalization for activity recognition via adaptive feature fusion.

Rani, V., Nabi, S. T., Kumar, M., Mittal, A., and Kumar, K. (2023). Self-supervised learning: A succinct review. *Archives of Computational Methods in Engineering*, 30(4):2761–2775.

Ravi, N., Dandekar, N., Mysore, P., and Littman, M. L. (2005). Activity recognition from accelerometer data. In *Aaai*, pages 1541–1546. Pittsburgh, PA.

Reiss, A. (2012). PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NW2H.

Roggen, D., Calatroni, A., Nguyen-Dinh, Long-Vanand Chavarriaga, R., and Sagha, H. (2012). OPPORTUNITY Activity Recognition. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5M027.

Ruzzon, M., Carfì, A., Ishikawa, T., Mastrogiovanni, F., and Murakami, T. (2020). A multi-sensory dataset for the activities of daily living.

Saeed, A., Ozcelebi, T., and Lukkien, J. (2019). Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2):1–30.

Sani, S., Wiratunga, N., Massie, S., and Cooper, K. (2016). Selfback - activity recognition for self-management of low back pain. In *SGAI Conferences*.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. (2013). Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772.

Straczkiewicz, M., James, P., and Onnela, J.-P. (2021). A systematic review of smartphone-based human activity recognition methods for health research. *NPJ Digital Medicine*, 4(1):148.

Sztyler, S. (2016). On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9.

Tang, Y., Zhang, L., Teng, Q., Min, F., and Song, A. (2022). Triple cross-domain attention on human activity recognition using wearable sensors. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5):1167–1176.

Thukral, M., Haresamudram, H., and Ploetz, T. (2023). Cross-domain har: Few shot transfer learning for human activity recognition.

Um, T. T., Pfister, F. M., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., and Kulić, D. (2017). 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.

van Hees, V., Charman, S., and Anderson, K. (2018). Newcastle polysomnography and accelerometer data.

Van Hees, V. T., Gorzelniak, L., Dean León, E. C., Eder, M., Pias, M., Taherian, S., Ekelund, U., Renström, F., Franks, P. W., Horsch, A., et al. (2013). Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity. *PloS one*, 8(4):e61691.

Walmsley, R., Chan, S., Smith-Byrne, K., Ramakrishnan, R., Woodward, M., Rahimi, K., Dwyer, T., Bennett, D., and Doherty, A. (2022). Reallocation of time between device-measured movement behaviours and risk of incident cardiovascular disease. *British journal of sports medicine*, 56(18):1008–1017.

Weiss, G. (2019). WISDM Smartphone and Smartwatch Activity and Biometrics Dataset . UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5HK59.

Willetts, M., Hollowell, S., Aslett, L., Holmes, C., and Doherty, A. (2018). Statistical machine learning of sleep and physical activity phenotypes from sensor data in 96,220 uk biobank participants. *Scientific reports*, 8(1):7961.

Xu, H., Zhou, P., Tan, R., and Li, M. (2023). Practically adopting human activity recognition. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '23, New York, NY, USA. Association for Computing Machinery.

Yang, J., Nguyen, M. N., San, P. P., Li, X., and Krishnaswamy, S. (2015). Deep convolutional neural networks on multichannel time series for human activity recognition. In *Ijcai*, volume 15, pages 3995–4001. Buenos Aires, Argentina.

Yang, Y., Hou, C., Lang, Y., Guan, D., Huang, D., and Xu, J. (2019). Open-set human activity recognition based on micro-doppler signatures. *Pattern Recognition*, 85:60–69.

Yuan, H., Chan, S., Creagh, A. P., Tong, C., Clifton, D. A., and Doherty, A. (2023). Self-supervised learning for human activity recognition using 700,000 person-days of wearable data.

Zhang, R. (2019). Making convolutional networks shift-invariant again.