Acquisition of Unlabeled Dataset for Human Activity Recognition

Asahi Miyazaki[1,a)] Huang Tengjiu[1,b)] Tsuyoshi Okita[1,c)] Asahi Nishikawa[1,d)]

Abstract: In the paper, we recorded the device's IMU sensors in the setting of Human Activity Recognition task for the purpose of building the pretrained model by self-supervised learner. To use the data for self-supervised learning, we collected it without labels. Therefore, unlike typical human activity recognition datasets, no labels were assigned, and it is important to note this distinction. As a result, the cost of label collection was minimized, but the time required for data collection remained the same, even though the data was unlabeled. On the other hand, since the data was unlabeled, there was no need to go through the complex process of assigning labels or inputting labels through a smartphone, which meant that data collection could be carried out without any prior knowledge of human activity recognition. The dataset consists of data from three subjects, totaling 228 hours. We performed self-supervised learning on the model and evaluated its performance using other IMU datasets. While this data was originally collected for human activity recognition, we anticipate that the use of unlabeled data in self-supervised learning will become more common in the future. In such cases, this dataset should be suitable for tasks that use unlabeled IMU data.

1. Introduction

The dataset described in this paper was collected for sensor-based human activity recognition, but there is a significant distinction compared to previous approaches. Historically, human activity recognition has typically involved constructing machine learning models through supervised learning, which requires labeled sensor data (e.g., [1], [2], [5]). Consequently, in sensor-based human activity recognition, data was collected while manually assigning labels to the activities. As a result, segments where a specific activity was performed were labeled accordingly. Initially, labels were recorded manually during data collection, but in recent years, it has become common to input these labels through smartphone applications.

In contrast, the rise of large language models (LLMs) [4], [11] in recent years has brought attention to self-supervised learning (SSL) models [3] based on transformer architectures. These pre-trained models are trained on unlabeled data, which allows us to introduce a self-supervised learning method for human activity recognition, called SENvT [6]. In this self-supervised learning approach, training is performed using unlabeled data, followed by downstream tasks where labeled datasets are used for the specific task learning. Therefore, in self-supervised learning, training data can come from a domain different from the target dataset, meaning out-of-distribution (OOD) data can be used for training. This type of learning was traditionally referred to as transfer learning but is now more commonly called out-of-distribution data learning.

On the other hand, collecting data without labels literally means that no detailed information about the activities performed is recorded. As such, in the latter part of this paper, we describe our attempt to estimate labels for this data using the self-supervised learning model described above.

The structure of this paper is as follows. Section 2 outlines the data collection method, while Section 3 discusses the statistical properties of the data. These two sections primarily focus on data collection. Sections 4, 5, and 6 detail our efforts to apply the self-supervised learning model to infer labels for the activities in this dataset. Finally, Section 7 presents the conclusion.

2. Application for Sensor Data Acquisition

We used Physics Toolbox Sensor Suite [10] for data acquisition. This is a smartphone application for recording sensor data directly from the device. Also, the recorded data can be stored as csv files to a cloud such as Google Drive. The application uses all supported sensors, which are a G-force sensor, linear accelerometer, gyroscope, barometer, magnetometer, and GPS. We sampled the data at a frequency of 100 Hz, although the sampling frequency depends on the smartphone.

The participants who recorded the data put their smartphone in their breast pocket or hip pocket and recorded the

[1] Kyushu Institute of Technology
[a)] miyazaki.asahi676@mail.kyutech.jp
[b)] huang.tengjiu275@mail.kyutech.jp
[c)] tsuyoshi@ai.kyutech.ac.jp
[d)] nishikawa.asahi188@mail.kyutech.jp

sensor data. Participant 1 carried their device in their hip pocket or in their hand most of the time while Participant 2 usually carried their device in their breast pocket.

The protocol for data acquisition consists of these 5 steps: (1) the participant launches this application on their smartphone. (2) Enable all types of sensors. (3) Push the Start Recording button. (4) Push the End Recording button. (5) The participant stores the acquired data to the specific place with the specific file name. The acquired data are already csv files. Therefore, they can be used as a dataset where their file names are the date of the acquisition.

3. Size of Collected Data

The statistics of our dataset are as follows. We collected the data for about a year. It contains data from three people. Table 1 shows the length of the collected data per month and the number of people who collected the data. The total length is about 228 hours.

Table 1: The length of the collected data per month.

Year/Month	Total length (Hours)	Participants
2023/10	11.5	2
2023/11	86.7	2
2023/12	16.2	2
2024/1	55.7	1
2024/2	8.71	1
2024/3	8.05	1
2024/4	2.95	1
2024/5	9.63	1
2024/6	16.1	1
2024/7	5.04	1
2025/1	7.04	1

3.1 Range of activities

Five primary behavioral patterns were targeted to be recorded during data acquisition. These patterns include a range of daily activities.

Fig. 1 shows the rough statics of activities. Rough means that we did not record activities. Therefore, this is based on the very rough numbers that the subjects remember. For example, when the subject take the route 3 (Refer the next section about the route), this routes consists of walking and standing (=stationary behavior). However, usually we do not remember how many times we stop in the middle, we roughly calculate that we walk 3 times and stop 2 times.

Standing (=Stationary behavior) accounts for approximately 10% of the dataset, providing a baseline for comparison with more dynamic activities. Running represents about 5%, reflecting high-intensity movement data. Cycling contributes roughtly 10%, offering insights into moderate-intensity activities, while traveling by car or bus comprises approximately 5%, capturing data on vehicular motion and its unique characteristics.

3.2 Situations for data acquisition

There were 7 situations in which Participant 1's data were collected. They collected the data in and around the Iizuka campus of Kyushu Institute of Technology.

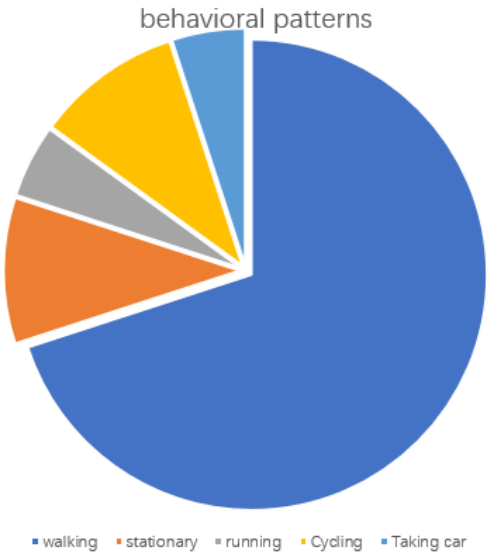

Fig. 1: Recorded Activities (Rough Statistics)

The main locations where we acquired data and the main behavioral patterns at these locations are as follows:

- **From student dormitories to university (around 30% of the all dataset):**

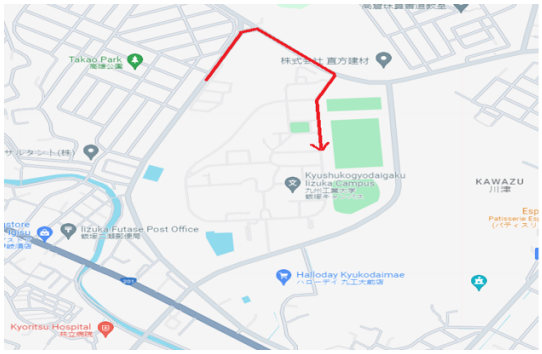

Fig. 2: Route 1

This route is the main one in my dataset. The behavior pattern consists of walking as the main part, roughly 75% of the entire process. Cycling is second about 15%. Finally, running about 5%, and standing about 5%.

- **Within university (arond 20% of the all dataset):**

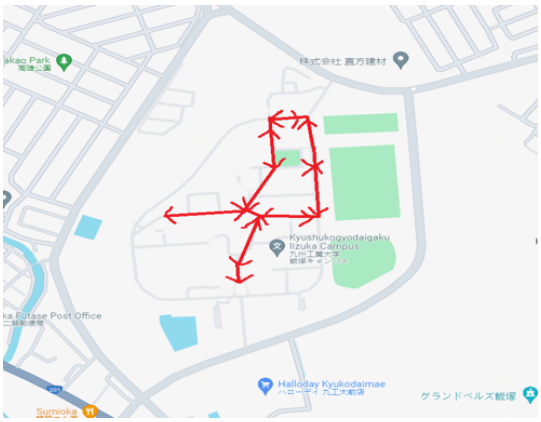

Fig. 3: Route 2

This route has the second highest amount of data acquired. In this route, the main behavioral pattern is walking about 80%, followed by standing about 20%.

- **From student dormitories to supermarkets (around 10% of the all dataset):**

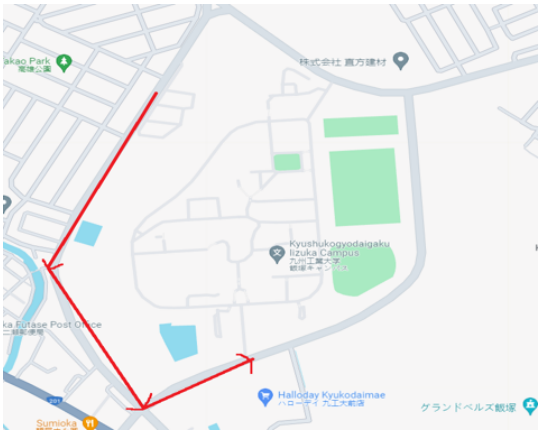

Fig. 4: Route 3

The next is from the student dormitory to the supermarket. In this route, the composition of behavior patterns is walking about 55%, cycling about 40%, and standing about 5%.

- **From school to supermarkets (around 10% of the all dataset):**

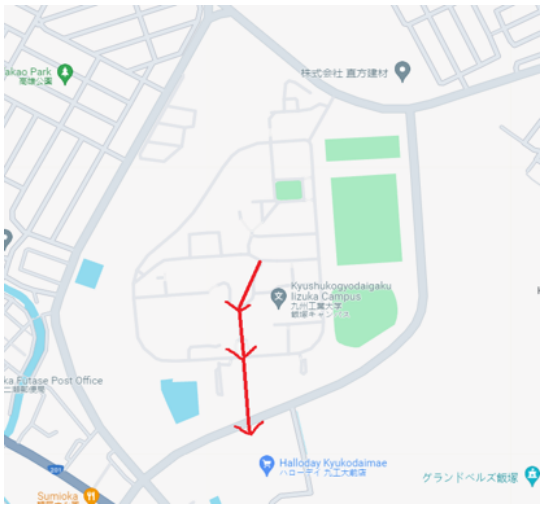

Fig. 5: Route 4

This route consists of walking about 75% and cycling about 25%.

- **From student dormitories to gym (around 15% of the all dataset)**

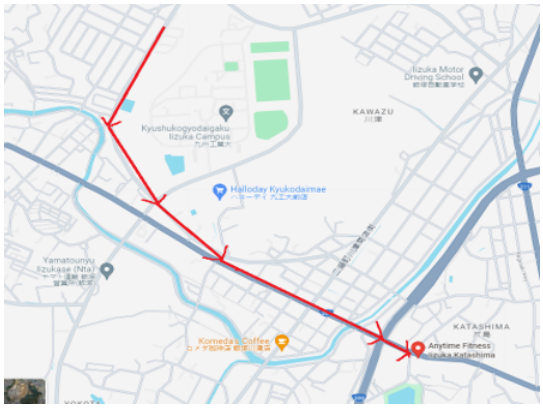

Fig. 6: Route 5

This route consists of walking approximately 60%, cycling, approximately 30%, running about 10%.

- **From school to gym (around 10% of the all dataset)**

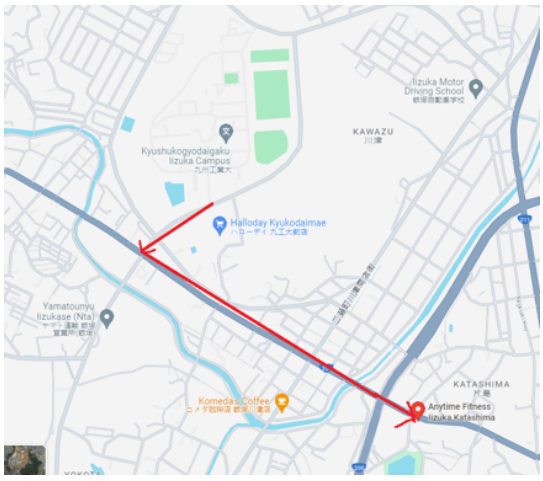

Fig. 7: Route 6

The main activities consists of walking around 80%, cycling about 15%, and running about 5%.

- **From student dormitories to station (around 5% of the all dataset):**

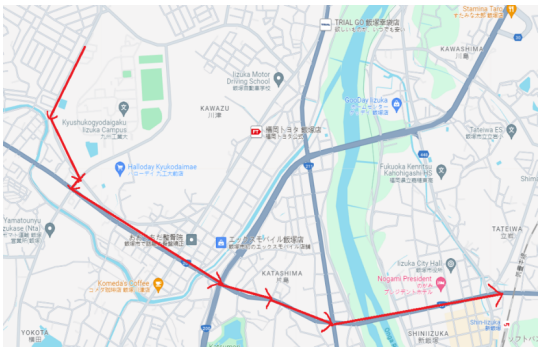

Fig. 8: Route 7

This route is used the least frequently. The proportion of taking cars or buses is the highest, which is about 90%, and cycling which is about 10%.

4. Inference of Activities

4.1 Preprocessing

The data acquired from the smartphones was recorded at an approximate 100 Hz period. From this raw data, we created 30Hz sliding windows which were 10 seconds each, as in [6]. Data acquired from the smartphones had the problem that the acquisition cycle was not constant. Most of the time, inertial data was recorded every 10 ms, but for some reason the recording was sometimes delayed. Since the model used in this study expects data to be sampled completely periodically, our dataset could not be used as was. Therefore, we pre-processed the data to make it available for training the model. We used Pandas 2.2.2 for pre-processing. The raw data included not only data from the accelerometer, but also data from magnetometers and other sources, but we used only the accelerometer data in this study.

First, the acquired data were resampled at 30 Hz. As mentioned above, our raw sensor data were not perfectly periodic and were delayed in some places. So, we filled with NaN the sections that were not recorded. This gave us the data sampled at 30 Hz evenly, although there were missing values.

Next, the data was separated by sliding windows. The size of the sliding windows was set to 300. This means that each sliding window contains 10 seconds of data. In addition, for the stride of the sliding windows, we created versions of 10, 5, and 2.5 seconds, respectively. This was done to increase the amount of dataset used in this study by reducing the stride, since the dataset used in this study was relatively small. In later experiments, we examined how this increase in stride affected the performance of the model.

Now that we obtained the sliding windows with missing values from the raw sensor data, we sorted each sliding window, the window must contain enough non-missing values. For example, if a sliding window contains only one or fewer entry, completion is not possible. Windows that cannot be completed were excluded at this point.

Finally, the NaNs contained in each sliding window were linearly interpolated. This gave us an evenly sampled inertial dataset with no missing values. Table 2 is the size of each dataset created by the above algorithm. Figure 9 is an example of a sliding window obtained by the above process. Each sliding window contains data in the X, Y, and Z axes at 30 Hz for 10 seconds.

Table 2: The numbers of the datasets of different strides. The bigger the stride is, the smaller the resulting dataset is.

Stride	Windows
2.5 seconds	98658
5 seconds	49411
10 seconds	24993

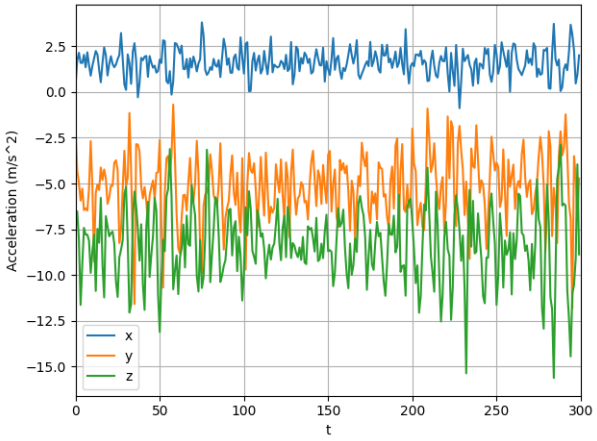

Fig. 9: An example of a preprocessed sliding window. Each sliding window contains 10 seconds of X, Y, and Z axis data at 30Hz. Its shape is 3×300.

5. Evaluation

To investigate the usefulness of the dataset created as described above, we performed human activity recognition tasks on this dataset. In this study, we used the transformer-based SENvT-u4 model [6].

5.1 SENvT-u4 model

The SENvT-u4 model is a transformer-based, self-supervised learning model. LLMs are often used in vision domain and natural language processing. In [6], they built a LLM for sensor data.

In the SENvT-u4 architecture, two learning stages take place: a first stage of self-supervised learning and a second stage of finetuning for downstream tasks.

In the first stage, the model was trained by using multiple types of pretext tasks as the purpose of self-supervised learning. The sliding windows were divided into multiple equal-sized patches, and a random pretext was used for each

patch. The pretext tasks include masked token, permutation, time warp, and rotation tasks. Since self-supervised learning was used, the dataset used in this stage does not need to contain labels.

In the second stage, the pretrained model obtained in the first stage was trained for downstream tasks by finetuning or transfer learning. Labels are needed for the dataset used in this stage.

5.2 Experimental methods

To examine the effect of the datasets acquired in this study, we compared models that were pretrained and then finetuned on the downstream datasets with those that were trained only on the datasets for downstream without pretraining.

In pretraining, we used our smartphone inertial dataset. The batch size was set to 256 and the learning rate was set to 1^{-5}. We sed AdamW as the optimizer. The maximum number of epochs was 100. A small amount of the dataset acquired in this study was used for validation, and the pretrained model with the smallest value of loss for the validation data was used for the downstream tasks.

After pretraining, we performed finetuning and transfer learning on the pretrained models. For finetuning, the batch size was set to 64, the learning rate was set to 1.25^{-5}, and AdamW was used as the optimizer. For transfer learning, the learning rate was set to 2.5^{-5}. Cross-entropy loss was used as the loss function. To address imbalances among classes in the downstream datasets, the model counted the number of instances of each class in the training dataset for the downstream task and weighted the classes. The downstream datasets and the number of classes and windows for each dataset are as shown in Table 3. The data used for training were 60% of the whole dataset. The rest was used for validation and testing, 20% each. Training for the downstream task was performed five times for one pre-trained model. For the resulting five models, performance was measured using the test data to calculate the accuracy and F1-score.

Table 3: The datasets used for the downstream tasks.

Dataset	Classes	Sliding windows
ADL [1]	5	1270
Opportunity [5]	4	8534
PAMAP2 [7]	8	5738
REALWORLD [8]	8	55992
WISDM [9]	18	24892

5.3 Results

In self-supervised learning, we consider training data for the pre-trained model and downstream task data in both In-Domain and Out-Of-Domain (OOD) scenarios.

- In-Domain Case: This refers to when both the pre-training data and downstream task data use CAPTURE-24 dataset.
- Out-Of-Domain (OOD) Case: In this case, the pre-training data uses CAPTURE-24, while the downstream task data uses datasets like ADL, Opportunity, PAMAP2, Realworld, and WISDM dataset.

Table 4 shows the results of in-domain finetuning of the SENvT-u4 model pretrained with the CAPTURE-24 dataset.

Table 5 shows the results of out-of-domain finetuning of the pretrained and non-pretrained models. It also shows the results for each stride. In every dataset, the finetuned models outperformed the non-pretrained models. Opportunity, REALWORLD, and WISDM showed the best performance at a stride of 2.5s. For ADL, the F1 score was highest at a stride of 10s, with 2.5s being the second best performing stride. PAMAP2 showed the best performance at a stride of 5s.

Table 6 shows the results when out-of-domain transfer learning was used instead of finetuning. The scores themselves are inferior to those with finetuning, but the inferiority of transfer learning over finetuning was also reported in [6]. In transfer learning, the performance improvement due to reduced stride was more significant than with finetuning. In all datasets, performance was highest at a stride of 2.5s. In particular, the REALWORLD dataset showed a large increase in f1 score, increasing by 0.1817 points. This also indicates that pretraining with our dataset resulted in improved performance.

These results indicate that pretraining with our dataset contributed to the performance improvement of the models. The smaller the stride of the sliding windows of the dataset, the higher the performance tended to be on downstream tasks, and this was especially true for transition learning. This means that small strides increased the diversity of the dataset, which resulted in higher performance in the downstream tasks.

Table 4: Results of In-Domain finetuning by the SENvT-u4 model on the CAPTURE-24 dataset. The model was pretrained with the CAPTURE-24 dataset as well. The results was not good at first sight but we understood this since we have unfortunately chosen the difficult test dataset.

Accuracy	Recall	Precision	F1
0.668	0.489	0.443	0.428

6. Labeling

The dataset built in this study has no labels. Therefore, we attempted to label this dataset using another dataset.

6.1 Methods

Our dataset include five behaviors: walking, standing, running, cycling, and driving. A very similar dataset is the SHL dataset from the Sussex-Huawei Locomotion Challenge 2023. This dataset has eight labels: still, walking, run, bike, car, bus, train, and subway. The SHL dataset contains a smartphone acceleration sensor data recorded at

Table 5: Results of Out-of-Domain (OOD) finetuning by the SENvT-u4 model on the downstream datasets, and also comparison between the pretrained models and the plain ones. It also shows the stride of the dataset used for pretraining. The highest scores are indicated as bold letters. In all datasets, pretraining with our dataset improved the performance.

Dataset	acc	f1
Pretrained (stride 2.5s)		
ADL	0.8772 ± 0.0107	0.8478 ± 0.0180
Opportunity	**0.7611 ± 0.0163**	**0.7671 ± 0.0241**
PAMAP2	0.8679 ± 0.0067	0.8618 ± 0.0070
REALWORLD	**0.9078 ± 0.0023**	**0.9175 ± 0.0024**
WISDM	**0.9001 ± 0.0027**	**0.8994 ± 0.0028**
Pretrained (stride 5s)		
ADL	**0.8929 ± 0.0107**	0.8445 ± 0.0168
Opportunity	0.7400 ± 0.0113	0.7341 ± 0.0096
PAMAP2	**0.8815 ± 0.0043**	**0.8786 ± 0.0047**
REALWORLD	0.8926 ± 0.0031	0.9041 ± 0.0029
WISDM	0.8816 ± 0.0062	0.8806 ± 0.0063
Pretrained (stride 10s)		
ADL	0.8882 ± 0.0153	**0.8539 ± 0.0147**
Opportunity	0.7375 ± 0.0290	0.7322 ± 0.0305
PAMAP2	0.8760 ± 0.0058	0.8728 ± 0.0060
REALWORLD	0.8892 ± 0.0037	0.9006 ± 0.0037
WISDM	0.8821 ± 0.0032	0.8809 ± 0.0030
Non-pretrained		
ADL	0.8630 ± 0.0242	0.8304 ± 0.0182
Opportunity	0.7424 ± 0.0156	0.7289 ± 0.0111
PAMAP2	0.8697 ± 0.0099	0.8654 ± 0.0108
REALWORLD	0.8866 ± 0.0041	0.8985 ± 0.0030
WISDM	0.8678 ± 0.0057	0.8657 ± 0.0059

Table 6: Comparison of the models built with OOD transfer learning. The highest scores are indicated as bold letters. Unlike the results of the finetuned models, the smallest stride 2.5s performed the best in every dataset.

Dataset	acc	f1
stride 2.5s		
ADL	**0.7874 ± 0.0423**	**0.6795 ± 0.0145**
Opportunity	**0.6581 ± 0.0119**	**0.6311 ± 0.0049**
PAMAP2	**0.6028 ± 0.0102**	**0.5739 ± 0.0111**
REALWORLD	**0.6708 ± 0.0020**	**0.6873 ± 0.0022**
WISDM	**0.6110 ± 0.0031**	**0.5948 ± 0.0043**
stride 5s		
ADL	0.7622 ± 0.0182	0.6427 ± 0.0083
Opportunity	0.6248 ± 0.0056	0.5827 ± 0.0066
PAMAP2	0.5523 ± 0.0031	0.5134 ± 0.0037
REALWORLD	0.5912 ± 0.0043	0.5808 ± 0.0028
WISDM	0.5508 ± 0.0027	0.5347 ± 0.0033
stride 10s		
ADL	0.7748 ± 0.0107	0.6082 ± 0.0054
Opportunity	0.5958 ± 0.0114	0.5743 ± 0.0078
PAMAP2	0.5477 ± 0.0030	0.4915 ± 0.0039
REALWORLD	0.5545 ± 0.0082	0.5056 ± 0.0053
WISDM	0.4728 ± 0.0021	0.4519 ± 0.0023

the subjects' eight different positions: body, hand, hip, and baggage. In this study, we only used the hand and body dataset in the experiment. This is because the participants who collected data in this study held their smartphones in their hands or put them in their pants or shirt pockets, so the data collection position was close to their hands or body.

For labeling, we used the SENvT-u4 model again. Firstly, we pretrained the SENvT-u4 model with our dataset whose stride is 2.5s, and then we finetuned it for the SHL dataset. For comparison, we also used the SENvT-u4 model pretrained with the CAPTURE-24 dataset [2]. The batch size was set to 128 and the learning rate was set to 2.5^{-5}. Finetuning was performed 5 times, and the model with the highest f1 score for the validation dataset was used. The results of the finetuning were as shown in Table 7. The CAPTURE-24 dataset is larger than our dataset and thus performed better. Comparing the SHL hand and body datasets, the body dataset performed better.

We used these finetuned models to infer activity labels for each sliding window in our dataset.

Table 7: The scores of the finetuned models calculated with the SHL dataset. We only used the training data of the SHL dataset. We divided that data into the training, validation, and test datasets again. The models showed in this table were finetuned with this training dataset and the f1-scores were calculated with the test dataset.

Pretrained with	Finetuned with	f1
Our dataset	SHL Hand	0.6871 ± 0.0028
CAPTURE-24	SHL Hand	0.7394 ± 0.0039
CAPTURE-24	SHL Body	0.8160 ± 0.0023

6.2 Results

Our dataset includes five behaviors: walking, standing, running, cycling, and driving. Of the total sliding windows, walking is the activity that accounts for the largest proportion of all sliding windows, accounting for more than half of the whole dataset. It is followed by standing and cycling activities, which are included in equal proportions. The activities with the smallest percentages are running and driving, which have similar percentages each other.

Table 8 shows the number of each label inferred by the finetuned models. It also shows the percentage of each of the five labels in the SHL dataset that are closest to the behaviors in our dataset.

The results of the model pretrained with our dataset and finetuned with the SHL hand dataset is at the top of Table 8. This model had the worst f1 score when its performance was measured with the SHL hand dataset. Instances inferred as running are 67.6% from the whole dataset, which is incorrect. On the other hand, instances inferred as walking are 9.42%, which is too few because the participant walked more than half of the time.

The results of the model pretrained with CAPTURE-24 and finetuned with the SHL hand dataset is at the middle of the table. This model showed the second best f1-score when its performance was measured with the SHL dataset. With this model, the number of instances inferred as the bus, train, and subway labels was very small. These activities are not included in our dataset. However, the number of instances inferred as walking was still too few and its ratio was 7.43%. The number of instances inferred as running was the largest ans the ratio was 65.0%.

The results of the model pretrained with CAPTURE-24 and finetuned with the SHL hand dataset is at the bottom of

the table. This model showed the best f1-score when its performance was measured with the SHL body dataset. With this model, the number of instances inferred as walking was the largest and the ratio was 28.3%. Also, the number of instances inferred as running was the smallest compared to the other models and the ratio was 38.3%. Although the number of instances inferred as running was still the highest, but it was the closer to the actual number of instances of running than the other models.

Also table 9 shows the number of each label inferred by the finetuned models, but adjacent sliding windows inferred to have the same label are concatenated and counted as one.

The results above demonstrate that the nature of the acceleration data measured differs depending on the placement of the smartphone. While the SHL dataset has fixed positions for the smartphone, such as on the hand or body, in our approach, the smartphone's position changes over time. As a result, we believe that the fine-tuning method using the SHL dataset failed to properly label the data.

Table 8: The number of instances of each label predicted by the finetuned SENvT-u4 models. It shows each result of the datasets used.

Activity	Windows	Ratio	Ratio (5 classes)
Our dataset + SHL Hand			
Still	528	2.67 %	2.89 %
Walking	1723	8.72 %	9.42 %
Run	12357	62.5 %	67.6 %
Bike	2314	11.7 %	12.7 %
Car	1364	6.90 %	7.46 %
Bus	503	2.54 %	-
Train	373	1.89 %	-
Subway	607	3.07 %	-
CAPTURE-24 + SHL Hand			
Still	915	4.63 %	4.63 %
Walking	1468	7.43 %	7.43 %
Run	12853	65.0 %	65.0 %
Bike	4476	22.6 %	22.6 %
Car	51	0.258 %	0.258 %
Bus	1	0.00506 %	-
Train	2	0.01011 %	-
Subway	3	0.0152 %	-
CAPTURE-24 + SHL Body			
Still	1122	5.68 %	5.93 %
Walking	5363	27.1 %	28.3 %
Run	7253	36.7 %	38.3 %
Bike	4957	25.1 %	26.1 %
Car	265	1.34 %	1.40 %
Bus	149	0.754 %	-
Train	592	2.99 %	-
Subway	68	0.344 %	-

7. Conclusions

In this study, we constructed an unlabeled accelerometer dataset using inertial sensors in ordinary smartphones.

We collected the data unlabeled in order to use it for self-supervised learning. Thus, the cost of collecting labels was low, but the time required to collect the data was the same, albeit without labels. Note that the time to assign labels when labels were available is compared to the time required for, for example, typing the current activity label with a smartphone, which is assumed to be almost negligible in terms of time. On the other hand, because the data were unlabeled, there was no need to learn the complicated process of how to apply the labels, and there was no need to input the labels with a smartphone and in this sense, the data could be collected with zero prior knowledge of human activity recognition. Our dataset consists of the data from three participants, and the total length of the data was 228 hours.

Table 9: The number of instances of each label predicted by the finetuned SENvT-u4 models. But, adjacent sliding windows inferred to have the same label are concatenated and counted as one.

Activity	Sections	Ratio	Ratio (5 classes)
Our dataset + SHL Hand			
Still	300	5.02%	6.89%
Walking	1023	17.1%	23.5%
Run	2245	37.6%	51.6%
Bike	586	9.81%	13.5%
Car	197	3.30%	4.53%
Bus	763	12.8%	-
Train	474	7.93%	-
Subway	386	6.46%	-
CAPTURE-24 + SHL Hand			
Still	254	4.20 %	4.20 %
Walking	1029	17.0 %	17.0 %
Run	2761	45.7 %	45.7 %
Bike	1950	32.2 %	32.3 %
Car	48	0.794 %	0.794 %
Bus	1	0.0165 %	-
Train	2	0.0331 %	-
Subway	3	0.0496 %	-
CAPTURE-24 + SHL Body			
Still	367	4.82 %	5.21 %
Walking	1714	22.5 %	24.3 %
Run	2826	37.1 %	40.1 %
Bike	1904	25.0 %	27.0 %
Car	229	3.00 %	3.25 %
Bus	139	1.82 %	-
Train	379	4.97 %	-
Subway	64	0.840 %	-

Furthermore, because this data was to be used in self-supervised learning, there was no need to collect labels unlike other datasets. It was limited to roughly tracing by memory what activities the participants performed. On the other hand, for this purpose, it is not necessary to seek an overview of the behavior of this data. However, we thought that if we trained a model with self-supervised learning, we would be able to obtain approximate labels. This consideration was analyzed in the latter part of this study.

We pretrained the SENvT-u4 model on this dataset and finetuned it on various other datasets, and observed performance improvements on all datasets. In addition, reducing the stride of the sliding windows of our dataset increased the diversity of the dataset and improved the performance of the model. In our attempts to label unlabeled datasets, we found that where on the user's body the sensor data was collected had a significant impact on the accuracy of activity recognition.

References

[1] Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa, Tullio Vernazza, and Renato Zaccaria, Analysis of human

behavior recognition algorithms based on acceleration data. In 2013 IEEE International Conference on Robotics and Automation, pages 1602–1607. IEEE, 2013.

[2] Matthew Willetts, Sven Hollowell, Louis Aslett, Chris Holmes, and Aiden Doherty, Statistical machine learning of sleep & physical activity phenotypes from sensor data in 96,220 uk biobank participants, Scientific Reports, 8(1):1–10, 2018.

[3] Carl Doersch, Andrew Zisserman, Multi-task Self-Supervised Visual Learning. 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2070–2079. 2017.

[4] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain et al., GPT-4 Technical Report, arXiv:2303.08774, 2023.

[5] Daniel Roggen, Alberto Calatroni, Mirco Rossi Thomas Holleczek, Kilian Forster, Gerhard Troster Paul Lukowicz, David Bannach, Gerald Pirkl, Florian Wagner, Alois Ferscha, Jakob Doppler, Clemens Holzmann+, Marc Kurz+, Gerald Holl, Walk-through the OPPORTUNITY dataset for activity recognition in sensor rich environments, 2010.

[6] Tsuyoshi Okita, Kosuke Ukita, Koki Matsuishi, Masaharu Kagiyama, Kodai Hirata, and Asahi Miyazaki, Towards LLMs for Sensor Data: Multi-Task Self-Supervised Learning, 2023 ACM, 2023.

[7] Attila Reiss and Didier Stricker, Introducing a new benchmarked dataset for activity monitoring. In Proceedings of 2012 16th International Symposium on Wearable Computers, pages 108–109. IEEE. 2012.

[8] Timo Sztyler, and Heiner Stuckenschmidt, 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. IEEE. 2016.

[9] Gary M. Weiss, Kenichi Yoneda, and Thaier Hayajneh, Smartphone and smartwatch-based biometrics using activities of daily living. IEEE Access, 7:133190–133202. 2019.

[10] Vieyra Software. Physics Toolbox Sensor Suite by Vieyra Software. https://www.vieyrasoftware.net/. (2025, January)

[11] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, Ji-Rong Wen, A Survey of Large Language Models. arXiv:2303.18223, 2023.