Contrastive Self-supervised Learning for Sensor-based Human Activity Recognition

Bulat Khaertdinov
Maastricht University
Maastricht, Netherlands
b.khaertdinov@maastrichtuniversity.nl

Esam Ghaleb
Maastricht University
Maastricht, Netherlands
esam.ghaleb@maastrichtuniversity.nl

Stylianos Asteriadis
Maastricht University
Maastricht, Netherlands
stelios.asteriadis@maastrichtuniversity.nl

Abstract

Deep Learning models, applied to a sensor-based Human Activity Recognition task, usually require vast amounts of annotated time-series data to extract robust features. However, annotating signals coming from wearable sensors can be a tedious and, often, not so intuitive process, that requires specialized tools and predefined scenarios, making it an expensive and time-consuming task. This paper combines one of the most recent advances in Self-Supervised Leaning (SSL), namely a SimCLR framework, with a powerful transformer-based encoder to introduce a Contrastive Self-supervised learning approach to Sensor-based Human Activity Recognition (CSSHAR) that learns feature representations from unlabeled sensory data. Extensive experiments conducted on three widely used public datasets have shown that the proposed method outperforms recent SSL models. Moreover, CSSHAR is capable of extracting more robust features than the identical supervised transformer when transferring knowledge from one dataset to another as well as when very limited amounts of annotated data are available.

1. Introduction

In recent years, numerous advances have been made in the field of deep learning, which are a result of supervised models that rely on massive amounts of annotated data. Nonetheless, a lack of labeled data sets a major challenge for the supervised learning paradigm. Besides, data labeling is a time-consuming and expensive process. These facts have inspired researchers to propose solutions where, at least, feature extraction algorithms can be effectively trained in an unsupervised manner. This paradigm is generally known as self-supervised learning (SSL).

The process of training a model within a self-supervised framework normally consists of two stages, namely a pretext task and fine-tuning. In the pretext task, the encoder is trained on unlabeled data while, during the fine-tuning process, only the output layers of a model are trained on features extracted by the encoder with frozen parameters. Such a protocol of training is particularly useful when only a small part of data is labeled.

Recently, SSL methods exploiting a contrastive learning concept have shown outstanding performance on computer vision tasks [3, 9, 21, 22]. This paradigm can be described as metric learning applied to instance-level classification. The main idea is to train a model to match different views crafted from the same data instance during the pre-training stage by contrasting them with views of other instances.

Various deep learning methods and approaches have been used to address time-series classification tasks over the recent years. In particular, sensor-based Human Activity Recognition (HAR) can also be considered as multivariate time-series classification if input data is collected using such sensors as accelerometers and gyroscopes. Sensor-based HAR has found applications in areas of pervasive computing [18], ambient assisting living [2, 16] and automation in manufacturing industry [20].

Data labeling is a major challenge in sensor-based HAR as well. For instance, it is almost impossible for a person to precisely label time-series data coming from accelerometer and gyroscope sensors without using corresponding videos. Moreover, it is even more challenging, expen-

978-1-6654-3780-6/21/$31.00 ©2021 IEEE
978-1-6654-3780-6/21/$31.00 ©European Union

sive and time-consuming than creating labels for videos, since there should be a specific tool where a data annotator can match time-series data and videos. There is a limited number of works applying self-supervised learning to the sensor-based HAR problem [6, 7, 8, 17], although this paradigm might significantly decrease the amount of labeled data needed for training a robust HAR model.

In this paper, motivated by the issues of data labelling and the success of SSL in other domains, we propose a self-supervised learning framework which combines powerful encoder architectures for sensor-based HAR with the recent advances in contrastive self-supervised learning, in order to address the problem of sensor-based HAR. The contributions of this study are listed as follows:

- We introduce a Contrastive Self-supervised learning approach to Sensor-based Human Activity Recognition (CSSHAR) which is based on the SimCLR framework [3] and uses a transformer-based architecture with one-dimensional CNNs as encoder.
- We propose using random compositions of up to five simple time-series augmentations within a random augmentation module in order to obtain a rich variety of views crafted from the initial time-series instances during the pretext task.
- Extensive experiments were conducted on three open-source datasets, namely MobiAct, UCI-HAR and USC-HAD, to compare the proposed framework to its supervised version and other SSL approaches applied to sensor-based HAR. The experiments including baseline activity recognition, transfer learning and a scenario with limited labeled data demonstrate the robustness of features extracted by the suggested CSSHAR topology.

2. RELATED WORK

Contrastive Learning. Self-supervised learning can be considered as a methodology to train feature extractors or encoders without using data labels. There are various families of self-supervised learning approaches which are different in a pretext task, i.e. the first stage of training where an encoder is trained on a complementary task without using ground-truth labels. In this paper, the emphasis is made on contrastive learning approaches. The idea of contrastive SSL is to train an encoder to match different representations of the same instance using distance-based loss computed for pairs of representations. In 2018, Oord et al. [22] introduced contrastive predictive coding (CPC) and applied it to audio, image and text data. In the case of audio signals, they made use of the temporal nature of data and proposed training a model to identify representations of the same instance at different timesteps. This work was later followed by contrastive multiview coding where authors suggested training an encoder to match two and more views (e.g. luminance, chrominance and depth) of the same image [21]. Finally, Chen et al. [3] proposed a so-called SimCLR framework which crafts two representations for each instance within a mini-batch using simple augmentations and feeds them into a Siamese network trained using NT-Xent (normalized temperature-scaled cross entropy) loss.

Sensor-based HAR and SSL. While plenty of modern deep learning methods have been used for sensor-based HAR in a supervised manner [24, 5, 15, 28, 26, 11, 13], only a few works have been focused on self-supervised learning. In [6], authors compared different types of autoencoders, including Convolutional Autoencoder (CAE), in terms of their ability to learn feature representations. Later, masked reconstruction models were suggested in [7] in order to use the temporal nature of sensory data. Specifically, the authors masked sensory signals to zero at certain timesteps and trained their model to reconstruct initial signal values on these timesteps. In 2019, Saeed et al. [17] adapted a concept of transformation networks to the sensor-based HAR task by creating a multi-task self-supervised approach. During the pretext task, a model was trained to identify the type of transformation (augmentation) applied to input data instances. Finally, in 2020, Haresamudram et al. [8] applied contrastive predictive coding (CPC) to the problem and outperformed all the methods described before. Comparing to the CPC approach, the proposed method does not need an autoregressive model on top of the encoder and the encoder is trained on short time windows, not long time sequences.

3. METHODOLOGY

3.1. Definitions

The Human Activity Recognition problem can be considered as multivariate time-series classification. Specifically, at timestamp t, input signal $\boldsymbol{x_t} = [x_t^1, x_t^2, \dots, x_t^S] \in \mathbb{R}^S$ consists of S values where each is obtained using a separate sensor channel. These multichannel signals are aggregated into a matrix $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_T}] \in \mathbb{R}^{T \cdot S}$ over T timestamps. Finally, the goal is to correctly assign a label $y \in Y$ to the vector $\boldsymbol{X}$ which is associated with a certain activity from a set of activities Y present in a dataset.

In this paper, the self-supervised learning paradigm is exploited in order to pre-train an encoder using unlabeled data, while class annotations are only used for fine-tuning the model with the frozen encoder in different scenarios. In this case, the encoder trained during the pretext task can be considered as a function $f : \mathbb{R}^{T \cdot S} \to \mathbb{R}^D$ which maps initial time windows into embeddings of size D. These embeddings are later passed through the MLP-based model $g : \mathbb{R}^D \to \mathbb{R}^Y$ at the fine-tuning stage.

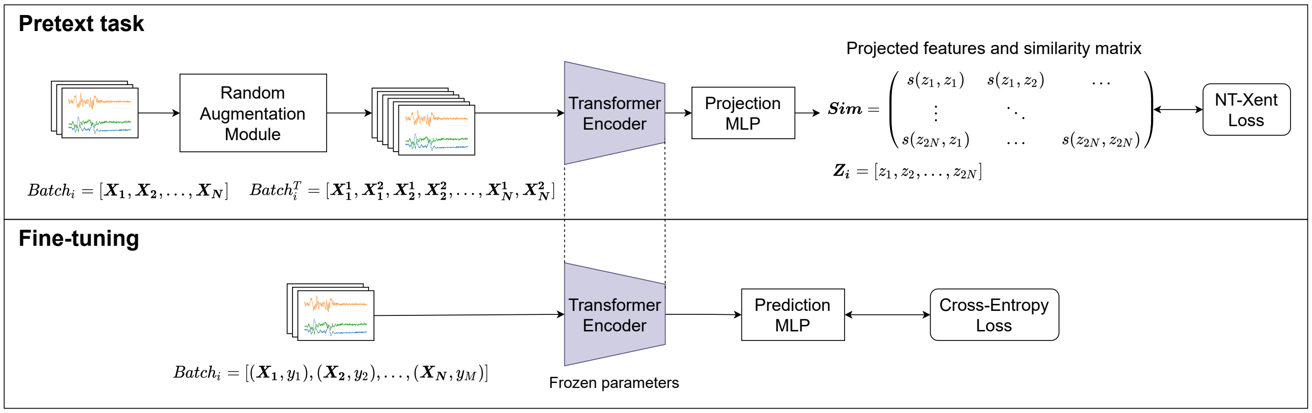

Pretext task
Random Augmentation Module
Transformer Encoder
Projection MLP
Projected features and similarity matrix
$Sim = \begin{pmatrix} s(z_1,z_1) & s(z_1,z_2) & \dots \\ \vdots & \ddots & \\ s(z_{2N},z_1) & \dots & s(z_{2N},z_{2N}) \end{pmatrix}$
NT-Xent Loss
$Batch_i = [\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_N}]$
$Batch_i^T = [\boldsymbol{X_1^1}, \boldsymbol{X_1^2}, \boldsymbol{X_2^1}, \boldsymbol{X_2^2}, \dots, \boldsymbol{X_N^1}, \boldsymbol{X_N^2}]$
$\boldsymbol{Z_i} = [z_1, z_2, \dots, z_{2N}]$
Fine-tuning
Transformer Encoder
Prediction MLP
Cross-Entropy Loss
$Batch_i = [(\boldsymbol{X_1}, y_1), (\boldsymbol{X_2}, y_2), \dots, (\boldsymbol{X_N}, y_M)]$
Frozen parameters

Figure 1: The proposed CSSHAR framework. **Pretext task**: a batch of signals is passed through the random augmentation module which generates 2 transformed views for each example. The encoder and projection head produce a representation vector $\boldsymbol{Z_i}$ which is later used for computing a view similarity matrix and the loss. **Fine-tuning:** labeled data is passed through the encoder with frozen weights and only the prediction MLP parameters are optimized using cross-entropy loss.

3.2. Proposed Framework

The CSSHAR framework presented in this study is shown in Figure 1. It is based on the SimCLR approach [3] which was introduced for representation learning in computer vision tasks. Given a batch of instances without labels, SimCLR, first, generates two views of each sample in the batch by applying two transformations. Two views obtained from the same instance form positive pairs, whereas views transformed from different instances are negative pairs. Second, this approach aims to match feature embeddings of positive pairs among features corresponding to negative pairs in the batch. The metric which is normally used as a matching score between two feature representations z_i and z_j is cosine similarity:

$$s(z_i, z_j) = \frac{z_i^T z_j}{||z_i||_2 ||z_j||_2}, \tag{1}$$

where $||\cdot||_2$ is the l_2 normalization operator. In other words, the goal of the SimCLR approach is to maximize similarity between features extracted for positive pairs of views and minimize similarity scores for the negative ones. The blocks of the proposed framework are described in detail in the following sections.

3.2.1 Random Augmentation Module

According to Chen et al. [3], compositions of data augmentations enhance the quality of learnt embeddings within the SimCLR framework. In this paper, we propose using random data transformations which consist of several simple time-series augmentations. Given a set of simple augmentations $A = \{a_1, a_2, \dots, a_K\}$, all augmentations are applied to an input signal one by one with a probability p. The simple augmentations used in this study are listed as follows:

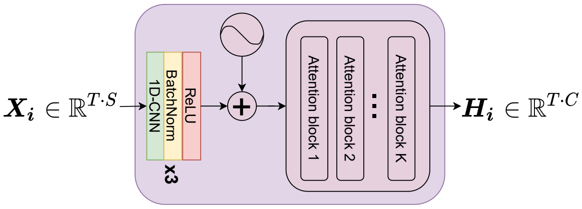

$\boldsymbol{X_i} \in \mathbb{R}^{T \cdot S}$
1D-CNN
BatchNorm
ReLU
x3
Attention block 1
Attention block 2
Attention block K
$\boldsymbol{H_i} \in \mathbb{R}^{T \cdot C}$

Figure 2: The architecture of the encoder. It takes input time-series consisting of T timestamps and S channels. The features extracted by the encoder also correspond to T step, while the number of channels C is a hyperparameter of the one-dimensional CNN. The feature representations are latter flattened into a one-dimensional vector of size $TC = D$ before feeding into projection or prediction MLPs.

- **Jittering.** Adds random Gaussian noise to signals.
- **Scaling.** An augmentation which multiplies input signals with values sampled from normal distribution.
- **Channel shuffle.** Randomly shuffle channels of multivariate time-series data.
- **Rotation.** Inverts the signs of the randomly selected values in signals.
- **Permutation.** Splits input signals into a certain number of intervals and randomly permutes them.

In order to exclude cases when no augmentations were applied to an instance, jittering is selected as a base augmentation and applied to each sample with $p = 1$. Different sets of these augmentations were used to find the best combination for each dataset.

3.2.2 Transformer-like Encoder

Recent works on sensor-based Human Activity Recognition exploit attention mechanisms in order to adaptively focus on the most important parts of input signals [13, 26]. In this work, we use a combination of a one-dimensional CNN with a transformer encoder as a backbone model as shown in Figure 2. Specifically, input signals are passed through the one-dimensional CNN with three layers including batch normalization [10] and ReLU activation [14]. We also apply reflective padding in order to preserve the initial length of time-series signals. The output of the CNN is then passed through positional encoding and the transformer encoder containing multiple self-attention blocks. The encoded features are then flattened in order to pass them through MLP models.

3.2.3 Pretext task

In SSL, the pretext task is a procedure of pre-training an encoder without using target labels. While some methods craft labels from data itself, contrastive approaches aim to match different views of the same instance using metric learning objective functions.

The pretext task implemented in this study is shown in the upper part of Figure 1. As can be seen from the figure, given a batch of N instances, two transformations are applied to each example using the random augmentation module (introduced in Section 3.2.1). Hence, the batch consists of $2N$ transformed views. Then, these views are consecutively passed through the encoder (described in 3.2.2) and the projection MLP which consists of 2 fully connected layers. Finally, in order to minimize cosine similarity between feature representations z_i and z_j corresponding to two augmented views of the same instance by contrasting it with all the other views in the batch, we use NT-Xent loss defined as follows [3]:

$$l(i,j) = -log \frac{exp(\frac{s_{(}z_i,z_j)}{\tau})}{\sum_{k=1}^{2N} \mathbb{I}_{[k\neq i]} exp(\frac{s_{(}z_i,z_k)}{\tau})}, \qquad (2)$$

where τ is a temperature parameter and $s_{(}z_i, z_j)$ is a cosine similarity (Equation 1) between representations z_i and z_j returned by the projection MLP. The loss function for the whole batch can be written as follows:

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} (l(z_{2k}, z_{2k-1}) + l(z_{2k-1}, z_{2k})). \qquad (3)$$

In the equation above, the loss value is computed for all positive pairs present in a batch in a symmetric way. According to Equation 3, for representations z_{2K} and z_{2K-1} forming a positive pair, losses $l(z_{2k}, z_{2k-1})$ and $l(z_{2k-1}, z_{2k})$ are computed. The difference between them is in the negative pairs similarity scores present in the denominator of Equation 2, i.e negative pairs formed with representation z_{2k} are used in $l(z_{2k}, z_{2k-1})$, whereas for $l(z_{2k-1}, z_{2k})$ negative pairs include embedding z_{2k-1}.

3.2.4 Fine-tuning routine

Fine-tuning is the last stage when embeddings generated by a pre-trained encoder are used in order to train a simple classification model with available labeled data. The fine-tuning routine exploited in this work is shown in the lower part of Figure 1. The projection model is dropped and the encoder is used for feature extraction. Notable, encoder parameters are frozen during the fine-tuning stage. The only model which is trained is a prediction MLP which takes features from the encoder and outputs. In our study, the prediction MLP consists of three layers with ReLU activation [14] and dropout [19].

4. EVALUATIONS

4.1. Datasets and Pre-processing

Three datasets which were exploited in previous works on SSL for sensor-based HAR, namely MobiAct [23], UCI-HAR [1] and USC-HAD [27], are used in this study. The pre-processing steps applied to these datasets are also based on the related works [7, 8, 17]. Initially, raw accelerometer and gyroscope signals are downsampled to 30Hz and segmented into 50% overlapping time-windows of 1 second length. Then, the training, validation and test splits are created based on subjects. Finally, signals are normalized to have zero mean and unit variance per channel based on training data.

MobiAct. This dataset was collected via a smartphone located in a pocket and consists of accelerometer, gyroscope and orientation (ignored in this work) signals. The second version of the dataset containing 11 activities performed by 61 subjects was used in this study. As proposed in [17], 20% of subjects are randomly sampled for the test split, while 20% of the remaining subjects are selected for the validation set.

UCI-HAR. Collected using a smartphone fixed on waists, the dataset consists of gyroscope and accelerometer signals corresponding to 12 activities and 30 users. As in [8], we used 6 main activities from the dataset. Splitting routine for UCI-HAR is the same as for the MobiAct dataset.

USC-HAD. The USC-HAD dataset consists of accelerometer and gyroscope data for 12 activities performed by 14 participants. The splitting protocol for this dataset is adapted from [7] and supervised HAR works: sensor data from subjects 11 and 12 was used for validation, while data from subjects 13 and 14 - was used as a test set.

4.2. Implementation Details

Augmentations. The random augmentation module applies each augmentation from a pre-defined set with probability $p = 0.5$. In order to avoid views without any transformation applied, we used jittering augmentation as a mandatory transformation with $p = 1$. We tested possible combinations of augmentation sets from applying jittering only and composing transformation using all five augmentations in order to find the best combination for each dataset. For the MobiAct dataset the best set is $\{$Jittering, Scaling, Rotation$\}$, for USC-HAD and UCI-HAR – $\{$Jittering, Scaling, Permutation$\}$.

Encoder properties and pretext setup. Since the used datasets have different sizes, we presume that different sets of hyperparameters would lead to the best performance. For the one-dimensional CNN encoder, we fixed the kernel size to 3, while the number of output channels was changed over $[32, 64, 128]$ (better for UCI-HAR and USC-HAD) and $[64, 128, 256]$ (MobiAct) for three layers of the network. The number of heads in multi-head attention was set to 8 while the number of attention blocks was tuned over 6 (best for MobiAct), 8 (UCI-HAR) and 10 (USC-HAD).

Pretext setup. The encoder is pre-trained within the proposed contrastive learning framework using LARS optimizer [25] for 200 epochs. This optimizer is suggested in [3] in order to stabilize training with large batch sizes which were set to 256 for UCI-HAR and USC-HAD and 512 for MobiAct before obtaining transformed views. The projection MLP consists of 1 layer with ReLU activation and output layer which returns projected representations for loss calculation. The number of neurons in both layers are equal and tuned over 256 (optimal value for UCI-HAR), 512 (USC-HAD) and 1024 (MobiAct).

Fine-tuning. The prediction model consists of 2 hidden layers of 256 and 128 neurons with ReLU activation and dropout ($p = 0.2$) and output layer with softmax activation. The model parameters are optimized with Stochastic Gradient Descent with Adaptive Moment Estimation (ADAM) with the default parameters ($\epsilon = 10^{-8}$,$\beta_1 = 0.9$, $\beta_2 = 0.999$) [12].

4.3. Baseline Activity Recognition

In order to evaluate the quality of extracted embeddings, the baseline sensor-based HAR scenario is employed. Specifically, the proposed CSSHAR encoder is first trained on unlabeled data within the pretext task. Later, the encoder parameters are frozen and only the prediction model is fine-tuned on the whole training set with the original activity labels. We compare the performance of our SSL model in terms of mean F1-score against models reported in [8].

Additionally, we evaluated the identical transformer architecture trained in a supervised end-to-end manner. The supervised transformer model is composed of the same blocks as the suggested SSL model, namely the encoder containing CNN and transformer and the prediction MLP model. The encoder of the supervised model was not pre-trained via the pretext task and its parameters were not frozen while training on labeled data.

		Mean F1-Score		
Method	Type	MobiAct	UCI-HAR	USC-HAD
DeepConvLSTM [15]	Sup.	82.4	82.83	44.83
Transformer (ours)	Sup.	83.92	95.26	60.56
Multi-task SSL [17]	SSL	75.41	80.2	45.37
CAE [6]	SSL	79.58	80.26	48.82
Masked Reconstruction [7]	SSL	76.81	81.89	49.31
CPC [8]	SSL	80.97	81.65	52.01
CSSHAR (ours)	SSL	**81.13**	**91.14**	**57.76**

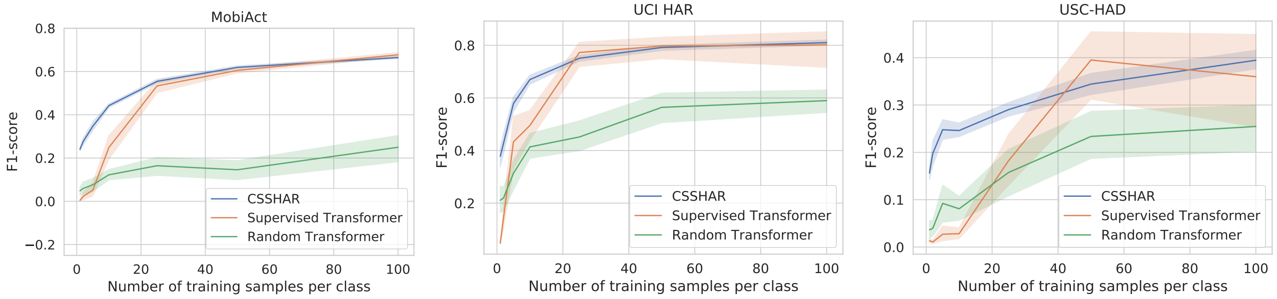
Table 1: F1-scores for the baseline activity recognition task.

The mean F1-scores for the models are aggregated in Table 1. In the table, sup. values in the type column refer to supervised methods. As it can be seen from the table, the proposed model outperforms all the previous SSL approaches. While for the MobiAct dataset the performance is comparable to the CPC approach, improvements on the UCI-HAR and USC-HAD datasets are more significant and make up approximately 9% and 5.75%, respectively. While comparing the suggested model to supervised models, it is clear that the supervised transformer is more powerful feature extractor in the scenarios when huge amounts of annotated data are available and shows performance higher by about 3-4% on all the datasets comparing to CSSHAR. It is worth mentioning that the proposed SSL model significantly outperforms DeepConvLSTM trained in a supervised manner on the UCI-HAR and USC-HAD datasets. The obtained results demonstrate that the proposed CSSHAR framework is capable of extracting robust feature embeddings without using data labels.

4.4. Semi-supervised Scenario

As it was mentioned before, sensory data labeling is an expensive and time-consuming process. That is why we implement a semi-supervised learning scenario when very limited annotated data is available. For this purpose, similarly to [7, 8, 17], we conduct a series of experiments when $k \in \{1, 2, 5, 10, 25, 50, 100\}$ labeled examples per class are randomly sampled from the training set and used to train the supervised and SSL models and compare their performance. The proportion of training data available varies within the following rates: 0.0149% – 1.49%, 0.0314% – 3.14% and 0.0342% – 3.42% for the MobiAct, UCI-HAR and USC-HAD datasets, respectively.

In this scenario, the CSSHAR encoder is frozen after the pretext task and only the prediction MLP model is fine-tuned on the selected instances. In case of the supervised model, it is trained on the sampled signals in an end-to-end manner without pre-training. For each k, the experiment is



(a) MobiAct (b) UCI-HAR (c) USC-HAD

Figure 3: Average F1-scores with 95% confidence intervals for the semi-supervised learning scenario.

repeated 10 times. The average F1-score with the 95% confidence intervals are drawn in Figure 3. We also illustrate the performance of a random transformer with the same architecture. In this case, sampled instances are passed through the randomly initialized and frozen transformer in order to obtain features. These features are then used to train the prediction model. The random transformer can be considered as CSSHAR without the pretext task.

What can be seen from the figure is that both supervised transformer and the SSL transformer (CSSHAR) show about the same performance when $k \geqslant 25$. However, from confidence intervals for the average value of F1-score, it is also clear that the supervised transformer is much more volatile and, hence, less robust than the proposed self-supervised model. For example, for the UCI-HAR dataset, the lowest F1-score value for the supervised transformer when $k = 100$ is 44.43%, while for CSSHAR it is 77.9%. Furthermore, for the USC-HAD dataset, confidence intervals of random and supervised models intersect almost for all the values of k. It is also crucial to mention that, unlike CSSHAR, the supervised transformer model completely fails when only very limited data ($k < 10$) is available and for all the datasets performs at about the same level as the random encoder.

4.5. Transfer Learning

The final evaluation scenario exploited in this study is transfer learning. In this experiment, encoders pre-trained on one dataset are evaluated on other datasets. As in [6], we pre-train the SSL encoder on the MobiAct dataset within the pretext task and fine-tune the prediction model on the remaining datasets. The performance of the SSL models was also compared to the supervised transformer model. In case of the supervised models, the encoder is trained in an end-to-end manner on the MobiAct dataset and frozen. Hence, only the prediction model is trained on the remaining datasets. The F1-scores for our SSL and supervised models and the models presented in [7] obtained in the transfer learning scenario are shown in Table 2.

		Mean F1-Score	
Method	Type	UCI-HAR	USC-HAD
DeepConvLSTM [15]	Sup.	73.68	25.57
Transformer (ours)	Sup.	86.62	39.8
Multi-task SSL [17]	SSL	73.89	31.35
CAE [6]	SSL	84.15	51.66
Masked Reconstruction [7]	SSL	81.37	46.19
CSSHAR (ours)	SSL	88.26	48.73

Table 2: F1-scores for the transfer learning scenario.

According to Table 2, CSSHAR outperforms all SSL models for the UCI-HAR dataset and performs worse only than the CAE model on USC-HAD. What can also be clearly seen is that the proposed model demonstrates better performance than the identical transformer-based model pre-trained on the MobiAct dataset in a supervised manner on both UCI-HAR and USC-HAD datasets by about 1.5% and 9%, respectively. These findings illustrate a high potential of the CSSHAR model to learn effective feature representations on unlabeled datasets and transfer its knowledge to new unseen data.

5. Conclusions and Future Work

In this paper, the powerful encoder based on transformer architecture is combined with the modern approach to the contrastive self-supervised learning approach in order to address the sensor-based HAR problem. The findings of our experiments demonstrate that the proposed model outperforms previous SSL methods and shows more robust performance in the semi-supervised and transfer learning scenarios compared to the identical supervised model.

There is a criticism in the literature regarding negative pairs in contrastive approaches since they might be obtained from the same class [4]. As for future work, it might be interesting to adapt methods that do not rely on negative pairs or suggest a technique that could decrease the effect of negative pairs coming from the same class.

References

[1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, page 437–442, 2013.

[2] J. Camps, A. Samà, M. Martín, D. Rodríguez-Martín, C. Pérez-López, J. M. Moreno Arostegui, J. Cabestany, A. Català, S. Alcaine, B. Mestre, A. Prats, M. C. Crespo-Maraver, T. J. Counihan, P. Browne, L. R. Quinlan, G. Laighin, D. Sweeney, H. Lewy, G. Vainstein, A. Costa, R. Annicchiarico, Àngels Bayés, and A. Rodríguez-Molinero. Deep learning for freezing of gait detection in parkinson's disease patients in their homes using a waist-worn inertial measurement unit. *Knowledge-Based Systems*, 139:119–131, 2018.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[4] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[5] N. Y. Hammerla, S. Halloran, and T. Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 1533–1540. AAAI Press, 2016.

[6] H. Haresamudram, D. V. Anderson, and T. Plötz. On the role of features in human activity recognition. In *Proceedings of the 23rd International Symposium on Wearable Computers*, pages 78–88, 2019.

[7] H. Haresamudram, A. Beedu, V. Agrawal, P. L. Grady, I. Essa, J. Hoffman, and T. Plötz. Masked reconstruction based self-supervision for human activity recognition. *Proceedings - International Symposium on Wearable Computers, ISWC*, pages 45–49, 2020.

[8] H. Haresamudram, I. Essa, and T. Plötz. Contrastive predictive coding for human activity recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(2), June 2021.

[9] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[11] B. Khaertdinov, E. Ghaleb, and S. Asteriadis. Deep triplet networks with attention for sensor-based human activity recognition. In *2021 IEEE International Conference on Pervasive Computing and Communications, PerCom 2021, Kassel, Germany, March 22-26, 2021*. IEEE, 2021.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[13] S. Mahmud, M. T. H. Tonmoy, K. K. Bhaumik, A. K. M. M. Rahman, M. A. Amin, M. Shoyaib, M. A. H. Khan, and A. A. Ali. Human activity recognition from wearable sensor data using self-attention. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain*, pages 1332–1339. IOS Press, 2020.

[14] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[15] F. Ordóñez and D. Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, Jan 2016.

[16] M. Panwar, D. Biswas, H. Bajaj, M. Jobges, R. Turk, K. Maharatna, and A. Acharyya. Rehab-net: Deep learning framework for arm movement classification using wearable sensors for stroke rehabilitation. *IEEE Transactions on Biomedical Engineering*, PP:1–1, 02 2019.

[17] A. Saeed, T. Ozcelebi, and J. Lukkien. Multi-task Self-Supervised Learning for Human Activity Detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2):1–30, 2019.

[18] P. Skocir, P. Krivic, M. Tomeljak, M. Kusek, and G. Jezic. Activity detection in smart home environment. *Procedia Computer Science*, 96:672–681, 12 2016.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[20] W. Tao, Z.-H. Lai, M. C. Leu, and Z. Yin. Worker activity recognition in smart manufacturing using imu and semg signals with convolutional neural networks. *Procedia Manufacturing*, 26:1159–1166, 2018. 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA.

[21] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 776–794, Cham, 2020. Springer International Publishing.

[22] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[23] G. Vavoulas., C. Chatzaki., T. Malliotakis., M. Pediaditis., and M. Tsiknakis. The mobiact dataset: Recognition of activities of daily living using smartphones. In *Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health - Volume 1: ICT4AWE, (ICT4AGEINGWELL 2016)*, pages 143–151. INSTICC, SciTePress, 2016.

[24] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 3995–4001. AAAI Press, 2015.

[25] Y. You, I. Gitman, and B. Ginsburg. Large batch training of convolutional networks. *arXiv: Computer Vision and Pattern Recognition*, 2017.

[26] M. Zeng, H. Gao, T. Yu, O. J. Mengshoel, H. Langseth, I. Lane, and X. Liu. Understanding and improving recurrent networks for human activity recognition by continuous attention. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ISWC '18, page 56–63, New York, NY, USA, 2018. Association for Computing Machinery.

[27] M. Zhang and A. A. Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 1036–1043, New York, NY, USA, 2012. Association for Computing Machinery.

[28] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang. Deep residual bidir-lstm for human activity recognition using wearable sensors. *Mathematical Problems in Engineering*, 2018:7316954, Dec 2018.