 doi:10.20944/preprints202202.0111.v1

Article

Wearable Sensor Based Human Activity Recognition with Transformer

Iveta Dirgová Luptáková [1], Martin Kubovčík [2] and Jiří Pospíchal [3,*]

[1] Department of Applied Informatics, Faculty of Natural Sciences, University of Ss. Cyril and Methodius, J. Herdu 2, 917 01 Trnava, Slovakia; iveta.dirgova@ucm.sk
[2] Department of Applied Informatics, Faculty of Natural Sciences, University of Ss. Cyril and Methodius, J. Herdu 2, 917 01 Trnava, Slovakia; kubovcik1@ucm.sk
[3] Department of Applied Informatics, Faculty of Natural Sciences, University of Ss. Cyril and Methodius, J. Herdu 2, 917 01 Trnava, Slovakia; jiri.pospichal@ucm.sk
* Correspondence: jiri.pospichal@ucm.sk

Abstract: This paper describes the successful application of the Transformer model used in the natural language processing and vision tasks as a means of processing the time series of signals from gyroscope and accelerometer sensors for the classification of human activities. The Transformer model is based on deep neural networks with many layers which can generalize well on signals. All measured signals come from a smartphone placed in a waist bag. Activity prediction is sequence-to-sequence, each time step of the signal is assigned a designation of the performed activity. Emphasis is placed on attention mechanisms, which express individual dependencies between signal values within a time series. In comparison with another recent result, the recognition precision was improved from 89.67 percent to 99.2 percent. The transformer model should in the future be included among the top options in machine learning methods for human activity recognition.

Keywords: Transformer; human activity recognition; time series; sequence-to-sequence prediction

1. Introduction

Human activity recognition is a very active field that seeks to identify human activities based on sensors available in everyday devices such as smartphones, tablets, or smartwatches. These devices can collect data from a wide sample of users and classify the signals using machine learning methods. Detection of human activities using mobile devices has great potential in medicine when it is possible to monitor patients with various diagnoses and control compliance with treatment procedures or to use it as prevention against performing prohibited activities. Apart from health monitoring and rehabilitation, it can be used in gaming, human-robot interaction, robotics, or sports [1].

Recently, a lot of effort was focused on human activity recognition by deep neural networks. Several types of deep neural networks are typically used for the sensor signals time series classification. Paper [2] is based on the transformation of the measured signal time series into a polar coordinate system, forming a pair of Gramian Angular Fields images. These images are then classified by a ResNet-based convolutional deep neural network. Compared to their approach, this paper focuses on the direct application of measured signals to the input of a neural network, eliminating the need for any complex pre-transformation of data. Moreover, the model in this paper is much smaller than ResNet used in [2]. The overall transformation and subsequent classification are ensured within the trained model, thanks to which it is possible to achieve higher speeds during prediction and it is not necessary to use demanding calculation models such as ResNet. Normalization is ensured so that the mean is close to 0 and the standard deviation is 1 as opposed to the min-max method used by Qin et al. [2].

 doi:10.20944/preprints202202.0111.v1

The paper [3] is primarily focused on the application of 1D CNN and LSTM. This combination can handle long time series, but it is not as effective as the Transformer model in massive parallelization of calculations. The transformer itself can process long time series at high speed without the need to combine multiple neural network approaches. Although the dataset used in [3] contained several times more examples than the dataset used here, it focused only on basic activities and transition activities, not on more complex movements such as Pick, Jump, Push-up, or Sit-up used in this paper. Moreover, a data augmentation method is introduced here for extending a dataset by manipulating existing data, making it possible to produce many new examples to supplement an existing dataset.

The paper [4] introduced a new approach - attention for learning multiscale features among multiple kernels of 1D convolution layers in HAR issues. In a similar way, the signals were preprocessed to 0 mean and 1 standard deviation, but the focus was on using special 1D convolution layers for the prediction of one label for the entire time series (window). 1D convolutional networks and recurrent neural networks, or a combination of both are among the most used approaches [5]. In this paper, the window size limit is restricted only by the memory capacity, and labels are assigned to each time step, using the previous steps in the time series. This paper also offers an alternative in the form of entirely fully connected layers without using any 1D convolution in signal processing.

The paper [6] focuses on the classification of simple as well as complex activities by widely used models like InceptionTime or DeepConvLSTM. The activities are captured using sensors in smartphones like in this paper, and smartwatches. Combined convolutional and recurrent neural networks are used for evaluation. The paper [7] focuses on the comparison of Feed Forward Neural Networks and Convolutional Neural Networks in terms of cross-validation on unseen subjects. This paper offers an alternative that directly focuses on using attention mechanisms to find connections in the time series between features.

This paper compares directly with [8], as it is based on their measurements. Each activity was recorded in a 300 time steps window width with a sampling frequency of 100Hz, which corresponds to 3 seconds of human activity. However, the data in [8] require complex pre-processing and extraction of significant features, which allow methods such as Random Forest to classify activities relatively accurately. When pre-processing signals, it is possible to use Fast Fourier Transformation, which extracts frequency-domain features from the input signal and at the same time suppresses to some extent the effect of noise on the classification [8].

This paper deals with the application of deep neural networks directly to the normalized time series of the signal from the sensors. There is an alternative approach to processing time series based purely on the attention mechanisms, called Transformer. The transformer directly focuses on using attention mechanisms to find correlations in the time series between features and allows massive parallelization of time series calculations, which is different compared to recurrent neural networks that iterate serially through a time series. Another advantage of the transformer is the longer path length between features in the time series, which allows for more accurate learning of the context in long time series [9]. Computation speed, as well as prediction accuracy, are key elements in working with human activities, where prediction can be performed directly on the mobile device. The sequence-to-sequence method is used in the prediction of activities [10], where all time steps from the Transformer output are considered and activity designations are assigned to them. In this way, it is possible to assign activity to each time step that the user has taken when measuring live values from a mobile device.

The main aim of this work is to show the suitability of the Transformer model for human activity recognition. The results fully support this objective.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0111.v1

2. Methods

2.1. Transformer

A transformer is a type of neural network, based purely on attention mechanisms, which, like recurrent or convolutional networks, typically process time series and look for correlations between features within time steps. It is frequently used to work with natural language, where it achieves higher scores than recurrent neural networks. The transformer consists of Multi-Head Attention, fully connected, normalization, and dropout layers. It also contains residual connections that help with the gradient backpropagation in a deep neural network.

Multi-Head Attention is based on the principle of mapping a query and a set of key-value pairs to an output. The output of the network is the weighted sum of values, where the weight is assigned to each value (V) based on the calculation of the compatibility function from the query (Q) and the corresponding key (K). Dot products of the query and all keys are calculated and then the softmax function is applied to normalize the obtained weights, which are multiplied by values, see eq. (1). Multi-Head Attention contains several modules called heads that have their own queries and a set of key-value pairs expressed by fully connected layers from the original queries and a set of key-value pairs fed to the input layer, see eq. (2). Subsequently, the outputs from each head are combined into one fully connected output layer, see eq. (3) and Figure 1. The advantage of using multiple heads lies in the ability to combine the different contexts found from each of the heads into one complex output.

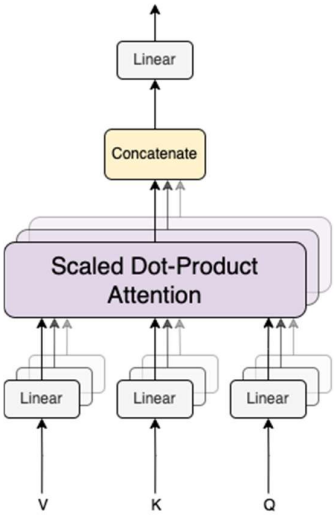


Figure 1. Multi-Head attention layer, amended from [9].

The Multi-Head attention layer is followed by the Position-wise Feed-Forward Network block, which is composed of a pair of fully connected layers linked by the nonlinear activation function RELU, see eq. (4). Typically, the number of neurons in the first fully connected layer is 4 times higher than in the following layer, where the number of neurons is equal to the number of features entering this block. The entry determining the position of the features within the series is also added to the network input, because the Transformer does not know the order of the features, for example within the processed sentence [9].

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

 doi:10.20944/preprints202202.0111.v1

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (3)$$

$$FFN(x) = \max(0, xW_1 + b_1)\, W_2 + b_2 \quad (4)$$

2.2. Vision Transformer

For image classification tasks, Transformer replaces the typically used convolutional networks with very good results. The image is divided into patches forming a sequence of features to which attention mechanisms are applied. However, a very large dataset of images of 14-300 million examples is needed for Transformer to achieve excellent results. The advantage of using a transformer in image classification tasks is its speed and scalability. In contrast to the original Transformer encoder [9], the normalization layer is applied before each block and residual connections after each block. The nonlinearity used in the Position-wise Feed-Forward Network block is GELU. The position of individual patches within the overall processed image is determined by the parameters that are adapted in the learning process together with the neural network [11].

2.3. KU-HAR dataset

The KU-HAR dataset used in this paper was published by Sikder and Nahid [8]. It was chosen here among other data sources, because it contains a lot of examples divided up into 18 classes (activities). Human Activity Recognition (HAR) data were obtained from 90 participants aged 18 to 34 years. The ratio of women to men among the participants was 1: 5. The weight range of the participants was 42.2 to 100.1 kg. The dataset contains 20,750 pre-processed examples, where each example captures 3 seconds of the performed activity, i.e., one whole time series of the signal represents just one performed activity and has only one label assigned to it. The measurements used sensors in a smartphone placed in a waist bag on each participant. The smartphone was facing left side down in the bag and the screen was pointing in the same direction as the participant. The first 11 activities in Table 1 were recorded indoors because they did not need a large space to perform. The other 4 activities were recorded outdoors. Stair-up and Stair-down activities were recorded on the staircase between the ground floor and the third floor, where there were 3 staircases between each floor. Table tennis was recorded in the common room located on the ground floor. The preprocessing consisted of deleting the part of the data recorded before the start of the performed activity because the first seconds of the recording did not correspond to the actual start of the performed activity. Similarly, an unrelated part of the records was removed at the end of the activity scan. The next step of the preprocessing was to unify the sampling frequency from all measurements to 100Hz. Because different smartphones with different computing power were used, not all measurements were identical concerning sampling frequency, and therefore one-dimensional interpolation of recorded time data was used when a particular measurement was recorded. The last step was to divide the measured activities into time series with a fixed length of 3 seconds. Each time series contains a unique portion of the original measurements [8].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0111.v1

Table 1. Description of the activity classes in the KU-HAR dataset, amended from [8].

Class name	Class ID	Performed activity	Duration or repetitions per sample	No. of extracted subsamples
Stand	0	Standing still on the floor	1 min	1886
Sit	1	Sitting still on a chair	1 min	1874
Talk-sit	2	Talking with hand movements while sitting on a chair	1 min	1797
Talk-stand	3	Talking with hand movements while standing up or sometimes walking around within a small area	1 min	1866
Stand-sit	4	Repeatedly standing up and sitting down (transition activity)	5 times	2178
Lay	5	Laying still on a plain surface (a table)	1 min	1813
Lay-stand	6	Repeatedly standing up and laying down (transition activity)	5 times	1762
Pick	7	Picking up an object from the floor by bending down	10 times	1333
Jump	8	Jumping repeatedly on a spot	10 times	666
Push-up	9	Performing full push-ups with a wide-hand position	5 times	480
Sit-up	10	Performing sit-ups with straight legs on a plain surface	5 times	1005
Walk	11	Walking 20 meters at a normal pace	~ 12 s	882
Walk-backward	12	Walking backwards for 20 meters at a normal pace	~ 20 s	317
Walk-circle	13	Walking at a normal pace along a circular path	~ 20 s	259
Run	14	Running 20 meters at a high speed	~ 7 s	595
Stair-up	15	Ascending on a set of stairs at a normal pace	~ 1 min	798
Stair-down	16	Descending from a set of stairs at a normal pace	~ 50 s	781
Table-tennis	17	Playing table tennis	1 min	458
			Total	20 750

2.4. Transformer for Human Activity Recognition

The example of the Vision Transformers has shown how effectively Transformers can replace existing recurrent and convolutional neural networks. Vision Transformer works with signals in the form of an image, which supports an assumption that it can also process 1D time series of signals from sensors such as an accelerometer or gyroscope. The Transformer for Human Activity Recognition presented further is based directly on the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0111.v1

Vision Transformer architecture [11], where, however, the signal is fed directly as input into the Encoder block along with the added information determining the position of the features within the signal time series.

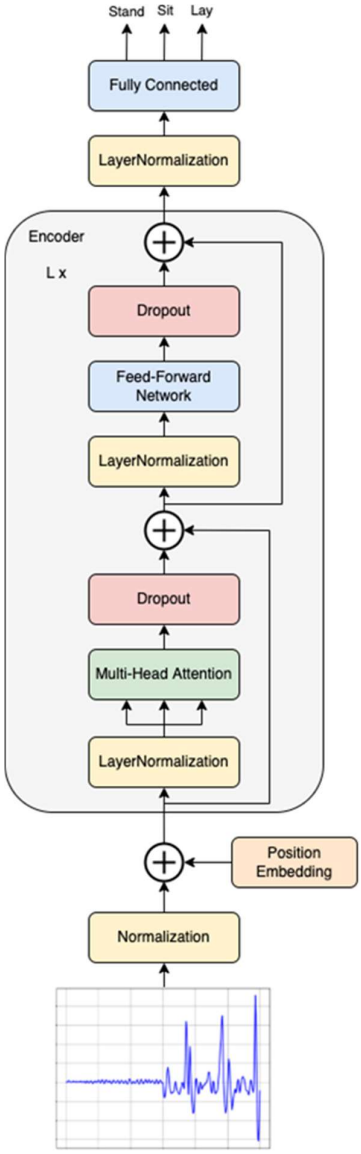


Figure 2. Transformer for Human Activity Recognition.

The Transformer for Human Activity Recognition operates in sequence-to-sequence mode and predicts the class for each time series feature, see Figure 2. The advantage is that if there are several consecutive classes in one time series, these classes can be easily identified, and the transformer is not limited to the features in the whole time series belonging to one class. All fully connected layers are initialized using the Truncated Normal distribution with a standard deviation of 0.02 as in BEIT [12]. Before the signal is fed as an input to the neural network, it passes through a normalization layer that stores the mean and variance obtained from the training data and adjusts the input to the values of 0.0 mean and 1.0 standard deviation. The advantage of this solution is that when the model is put into practical use, it already contains these calibration values, and it is not necessary to solve the signal adjustment in an external way. The model is completely ready for implementation in mobile devices, provided that the measured quantities are in the basic physical units, the accelerometer in m/s2, and the gyroscope in rad/s. The output layer of

 doi:10.20944/preprints202202.0111.v1

the model is linear, to provide a higher computational speed on less powerful devices than if the softmax function was used. In principle, the maximum value corresponding to the predicted activity can be obtained also before the application of softmax, which is only used during learning in the loss function. This form is used there due to the calculation with logarithms since it does not have negative numbers at the output, as the linear layer does.

The principal task in working with the signal time series is to find the correlations in it that will most positively affect the classification result. From the time series, the features are mutually matched by attention mechanisms, so that this Transformer ranks among the self-attention mechanisms, the same signal is applied to the input query, key, and value [9]. The advantage of using a Transformer in signal processing is again its speed and scalability, which has an impact on the usability of mobile devices and the accuracy of class predictions. Experiments show that it is a suitable alternative to recurrent and 1D convolutional networks for signal classification tasks. As with Vision Transformers, a huge number of examples are required, so a special data augmentation method has been devised here for signals expressing various human activities.

The implementation of the Transformer for Human Activity Recognition model was realized using the open-source library TensorFlow [13], which contains a rich set of tools for neural network design, their learning, evaluation, and deployment. It contained all the basic layers for Transformer creation: Normalization layers, Multi-Head Attention layer, dropout layer, fully connected (Dense) layer, up to the Position embedding layer, which was created as a custom layer by inheriting from the basic class Layer. The Encoder block is also created as an advanced custom layer from several simpler layers for easy replication. TensorFlow has also been used here for its professional deployment in many international companies and its high performance in mobile and embedded devices in the form of TensorFlow Lite.

2.5. Data augmentation

To extend the KU-HAR dataset, so that more training examples were available, an algorithm was chosen to combine pairs of activities that could follow each other in real life with a high probability. It was necessary to create all combinations of activity pairs from the original dataset, provided that identical activities were excluded. These resulting pairs had to be manually checked and their logical sense verified, see Table 2. The next step was to combine these activities into a double-length window, which needed to be transformed into a standard number of time steps used for previous training samples. The downsampling method was chosen, omitting every second step from the time series [14]. Vision Transformers were taught in a similar way, where randomly selected parts of the image were replaced by noise, and the Transformer aimed to fill these places identically to the original image [15]. For sensor data here, it is not necessary to replace the omitted time steps with noise, but the network must also process the signal and correctly identify the performed activity. As can be seen in Fig. 3, the numbers of examples in the classes substantially differ and therefore it was necessary to choose the method by which the examples of activity pairs will be generated. The smaller of the number of examples of both classes in pairs of activities was used. This avoids duplication of examples of the paired activity with fewer examples, which could cause overfit [16]. Transformer model also acquires a logical awareness of the connections between possible successive actions, which would not be possible with the coupling of pure random pairs of activities. 83,129 examples were obtained from the original 20,750 examples. The dataset was then divided into training, testing and validation sets in a ratio of 70:15:15 percent. Manipulation with the dataset was carried out by NumPy libraries [17] for working with matrices, Pandas [18] for working with CSV files, and Scikit-learn [19] for even distribution of examples according to classes per train, test, and validation dataset.

 doi:10.20944/preprints202202.0111.v1

Table 2. Newly created couples of activities.

Stand + Talk-stand	Sit + Talk-sit	Talk-Stand + Stand	Pick + Stand	Jump + Stand	Walk + Stand	Walk-backward + Stand	Walk-circle + Stand	Run + Stand	Stair-up + Stand	Stair-down + Stand	Table-tennis + Stand
Stand + Pick	Talk-sit + sit	Talk-Stand + Pick	Pick + Talk-Stand	Jump + Talk-Stand	Walk + Talk-Stand	Walk-backward + Talk-Stand	Walk-circle + Talk-Stand	Run + Talk-Stand	Stair-up + Talk-Stand	Stair-down + Talk-Stand	Table-tennis + Talk-Stand
Stand + Jump	Lay + Sit-up	Talk-Stand + Jump	Pick + Jump	Jump + Pick	Walk + Pick	Walk-backward + Pick	Walk-circle + Pick	Run + Pick	Stair-up + Pick	Stair-down + Pick	Table-tennis + Pick
Stand + Walk	Sit-up + Lay	Talk-Stand + Walk	Pick + Walk	Jump + Walk	Walk + Jump	Walk-backward + Jump	Walk-circle + Jump	Run + Jump	Stair-up + Jump	Stair-down + Jump	Table-tennis + Jump
Stand + Walk-backward		Talk-Stand + Walk-backward	Pick + Walk-backward	Jump + Walk-backward	Walk + Walk-circle	Walk-backward + Table-tennis	Walk-circle + Walk	Run + Walk	Stair-up + Walk	Stair-down + Walk	Table-tennis + Walk
Stand + Walk-circle		Talk-Stand + Walk-circle	Pick + Walk-circle	Jump + Walk-circle	Walk + Run		Walk-circle + Run	Run + Walk-circle	Stair-up + Walk-circle	Stair-down + Walk-circle	Table-tennis + Walk-backward
Stand + Run		Talk-Stand + Run	Pick + Run	Jump + Run	Walk + Stair-up		Walk-circle + Stair-up	Run + Stair-up	Stair-up + Run	Stair-down + Run	Table-tennis + Walk-circle
Stand + Stair-up		Talk-Stand + Stair-up	Pick + Stair-up	Jump + Stair-up	Walk + Stair-down		Walk-circle + Stair-down	Run + Stair-down	Stair-up + Stair-down	Stair-down + Stair-up	Table-tennis + Run
Stand + Stair-down		Talk-Stand + Stair-down	Pick + Stair-down	Jump + Stair-down	Walk + Table-tennis		Walk-circle + Table-tennis	Run + Table-tennis			
Stand + Table-tennis		Talk-Stand + Table-tennis	Pick + Table-tennis	Jump + Table-tennis							

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0111.v1

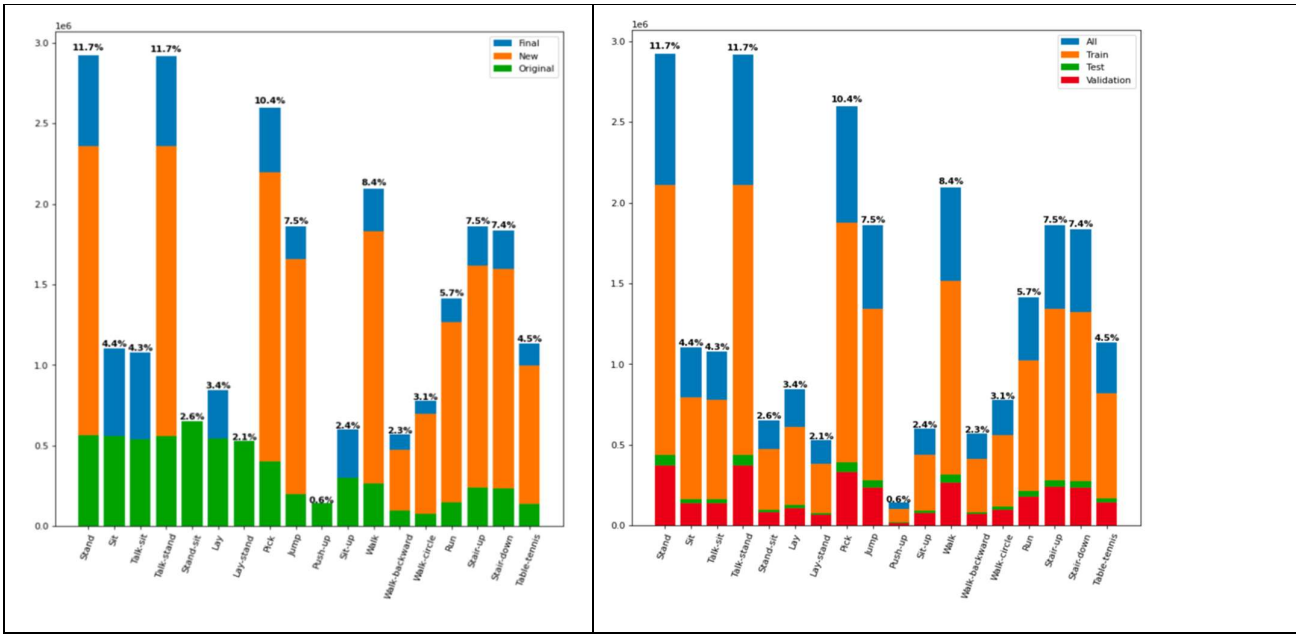


Figure 3. Distribution of examples by classes: (**a**) Distribution of examples by individual classes before and after the data augmentation process; (**b**) Distribution of examples by individual classes into training, testing validation datasets.

Figure 3 shows the distribution of examples by class. It is apparent, that the used dataset is slightly unbalanced. The Final represents the distribution of examples after applying data augmentation. New represents newly created examples from a combination of original examples and Original is a distribution of examples from the original dataset.

The Lay-stand and Stand-sit classes were omitted from pair combinations, as they represented a transition activity already composed of a pair of activities. The goal was to combine just two different activities and their involvement would create windows with up to three activities. Another completely omitted activity was Push-up, there was no suitable activity to pair it with, which would occur immediately before or after this activity. The only logically close activity was Lay, but it was "performed" on the back.

2.6. Finding optimal hyperparameters

Hyperparameters were optimized by the WanDB Sweep tool [20], which not only provides parallel coordinates chart for visualization of various settings but also offers a prediction of importance and correlation of hyperparameters against the selected metric. The used metric was the best validation accuracy obtained from the best prediction over the validation dataset during the learning process. The search method was random, which selected settings from predefined ranges of hyperparameters. Progressively, these intervals were manually adjusted to increase the accuracy of the model, and finally the most suitable combination of them was chosen, see Table 3.

Figure 4 shows how important the individual hyperparameters are for maximizing the best validation accuracy metric. The way in which they affect this metric is denoted by color. The red color indicates a negative correlation and the green color a positive correlation.

From Figure 5, it is possible to determine according to the color scale how the given value of the hyperparameter influenced the best accuracy in the predictions on the validation dataset. Yellow expresses the highest accuracy in predictions and blue the lowest.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0111.v1

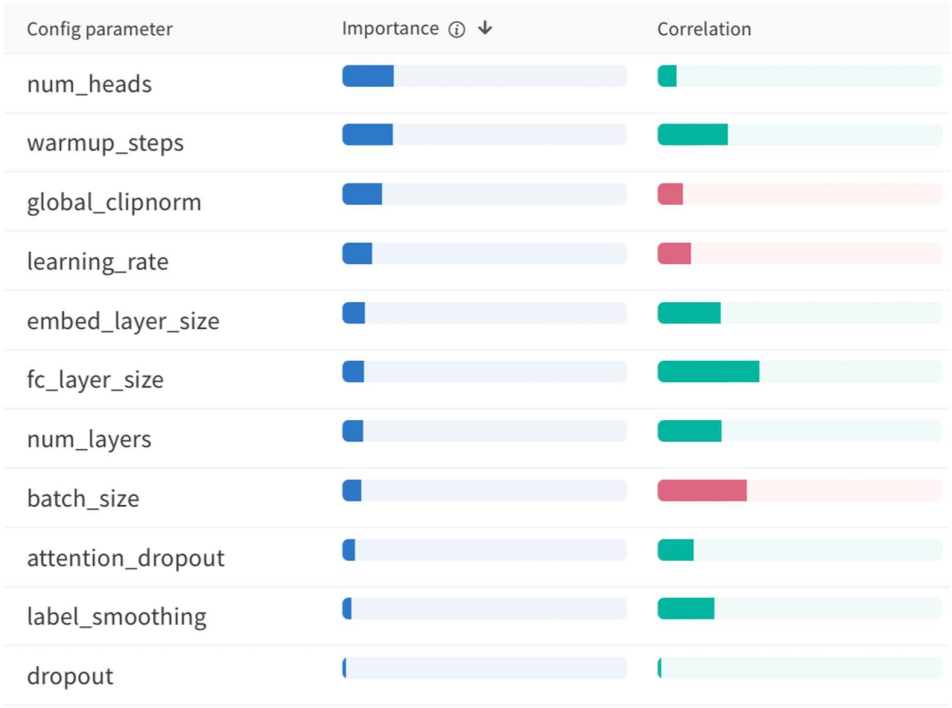


Figure 4. Parameter importance chart.

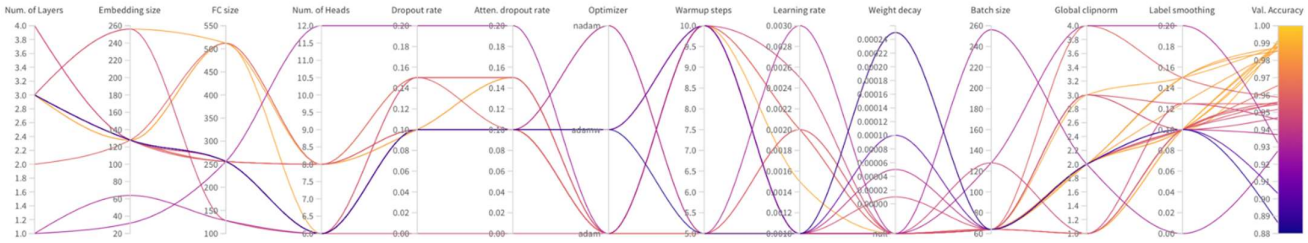


Figure 5. Parallel coordinates chart.

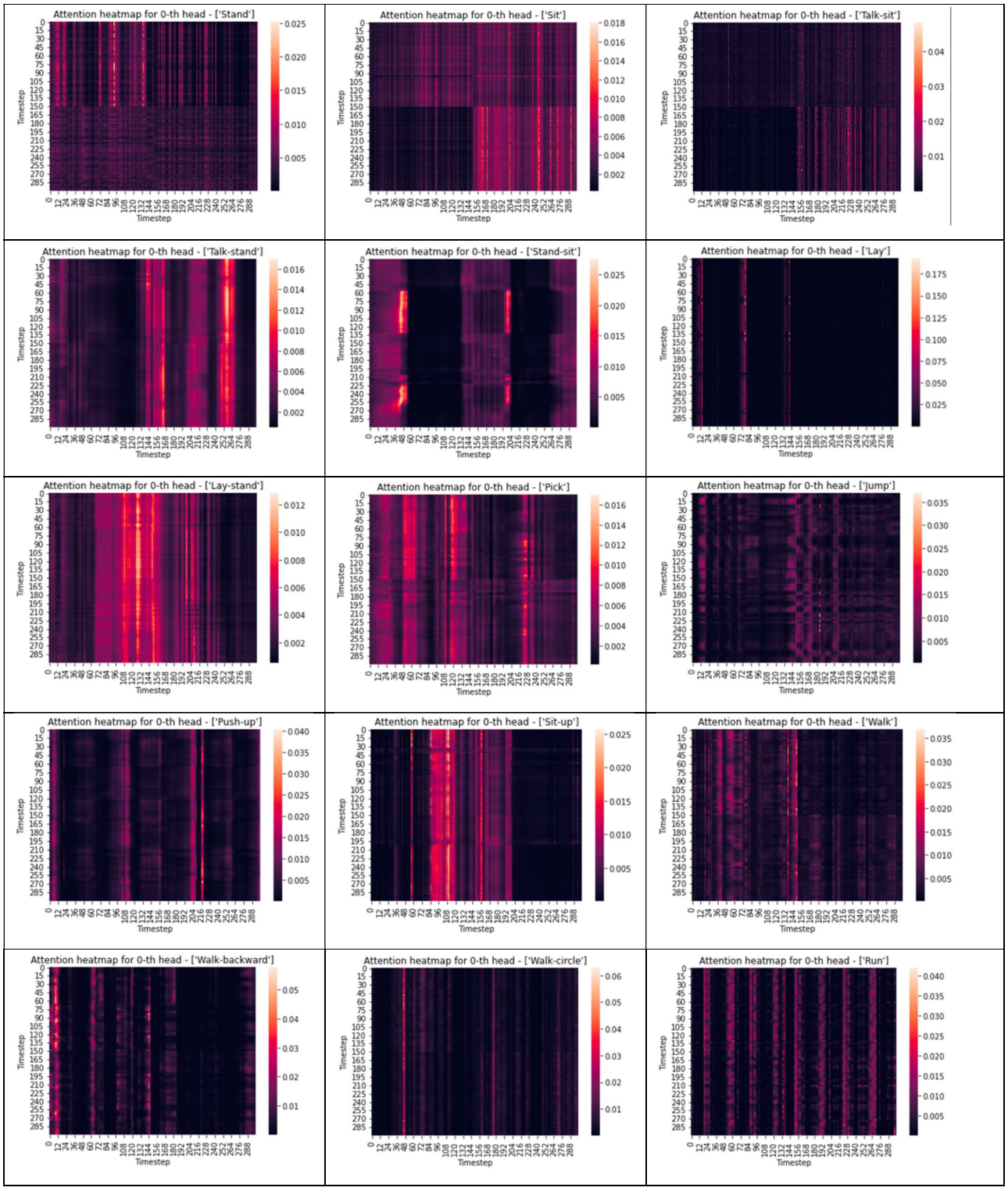
3. Results

In the testing phase, previously unseen examples from test datasets were presented to the neural network. The task was to predict activities from hitherto unseen signals with the highest possible accuracy. Fig. 6 shows the individual attention matrices from Head 1 of the Transformer expressing just one activity.

Similarly, Figure 7 shows selected examples of attention matrices from Head 1 of the Transformer expressing activity pairs. From the attention matrices, you can see the transition of activities in the exact half of the time series, when the first half belonged to one activity and the second half to another activity. It is also possible to see pairs of signal parts that are irrelevant to the correct classification of activity and, conversely, pairs that are very important. This proves the effectiveness of the Transformer algorithm in matching features from time series that contribute to correct classification.

Figure 8 shows the cosine similarities between the position embedding of the selected time step from the signal time series and all other time steps, indicating that the positions in the halves of the time series are the most similar. This phenomenon is caused by the combination of two different signals in the exact half during the data augmentation process. However, when using other ratios, there is a higher probability of losing essential information from the signal during downsampling; currently, there is a loss of half of the information from the signal from both parts of the time series.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0111.v1



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0111.v1

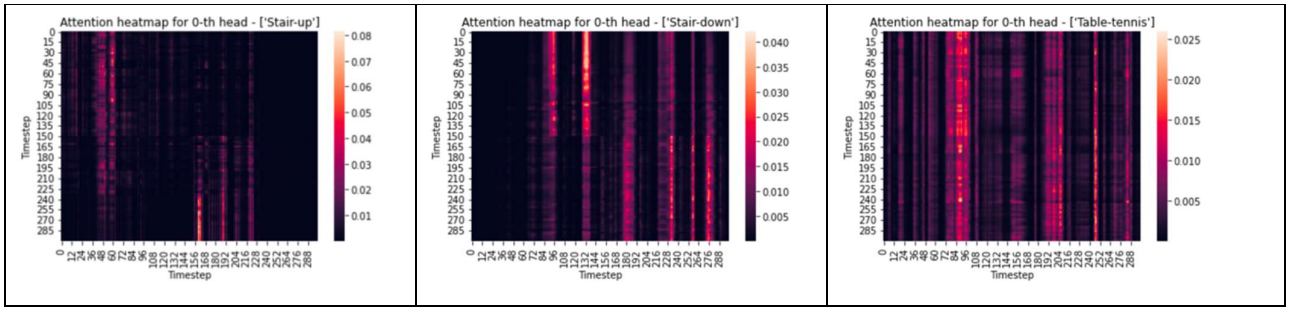


Figure 6. Attention heatmaps of different single activities.

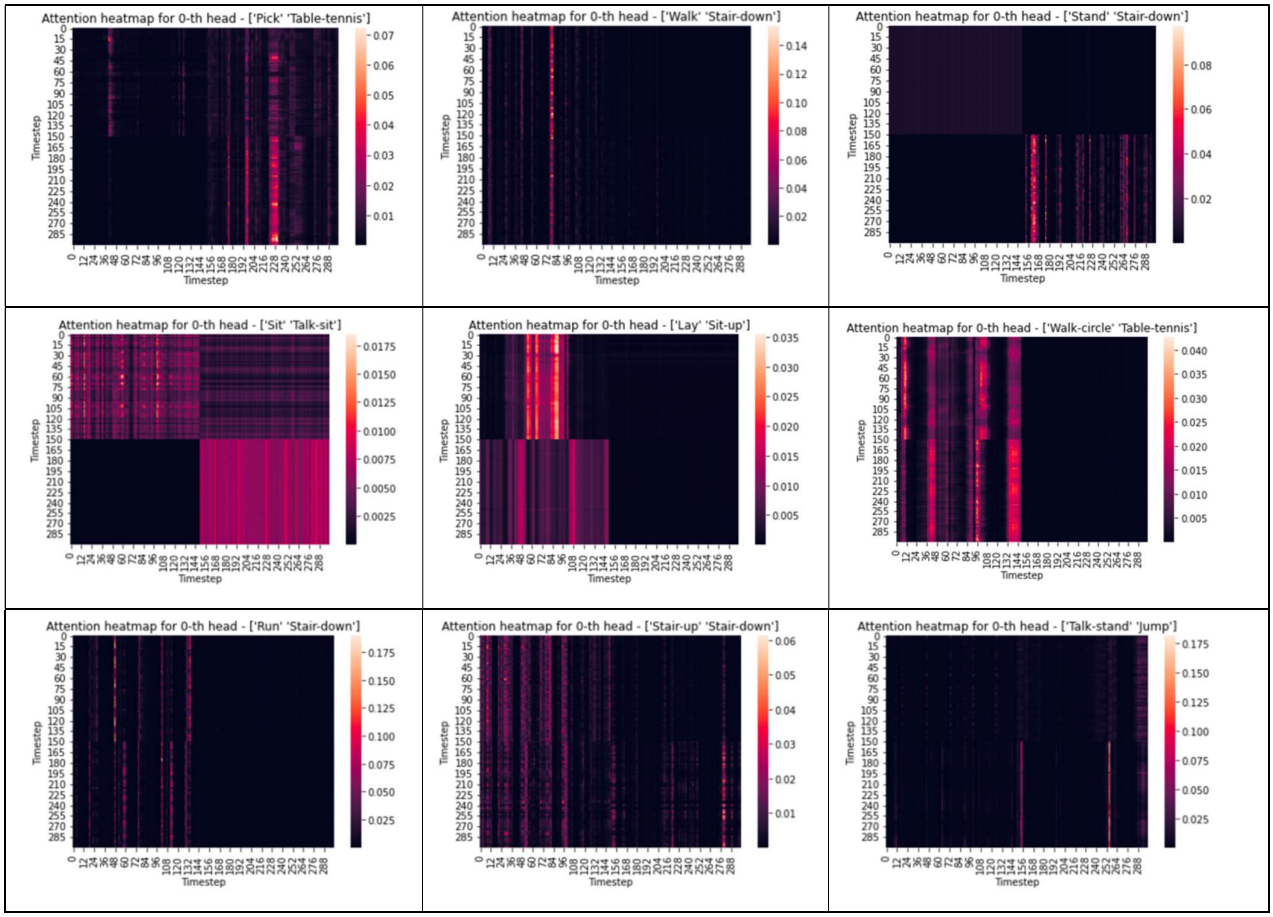


Figure 7. Attention heatmaps of different pairs of activities.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 February 2022 doi:10.20944/preprints202202.0111.v1

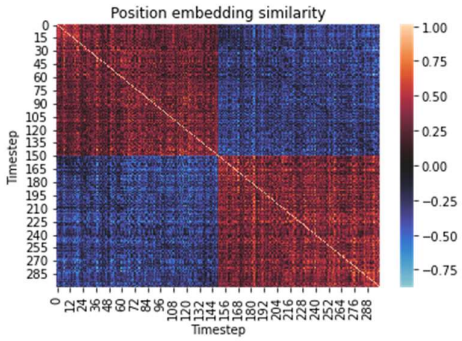
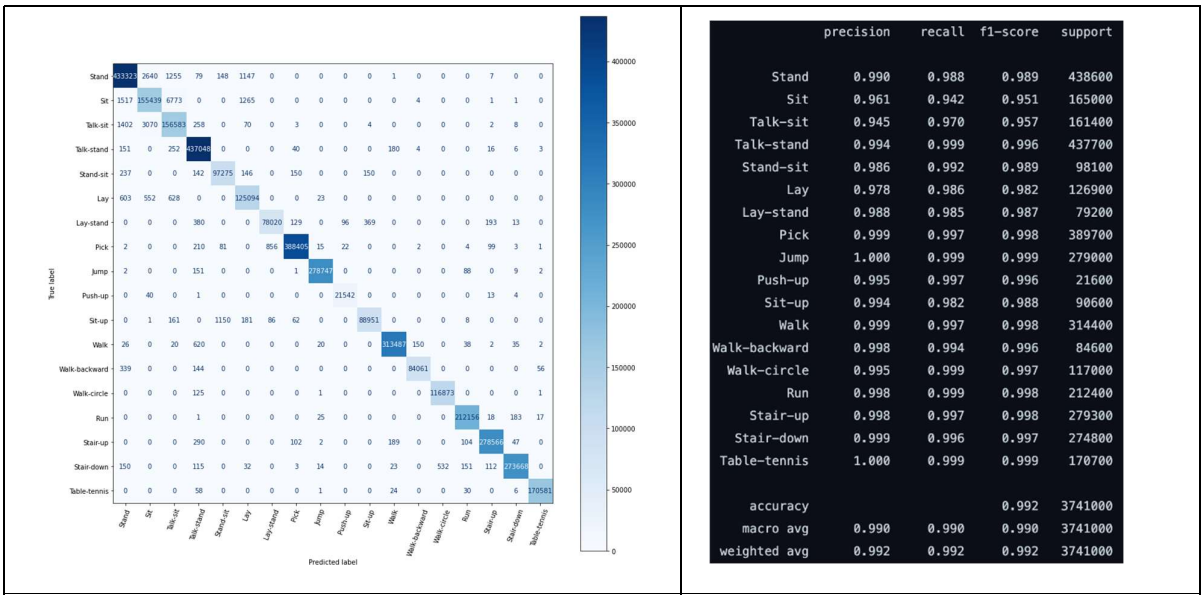


Figure 8. The cosine similarity between the position embedding of the timestep.



	precision	recall	f1-score	support
Stand	0.990	0.988	0.989	438600
Sit	0.961	0.942	0.951	165000
Talk-sit	0.945	0.970	0.957	161400
Talk-stand	0.994	0.999	0.996	437700
Stand-sit	0.986	0.992	0.989	98100
Lay	0.978	0.986	0.982	126900
Lay-stand	0.988	0.985	0.987	79200
Pick	0.999	0.997	0.998	389700
Jump	1.000	0.999	0.999	279000
Push-up	0.995	0.997	0.996	21600
Sit-up	0.994	0.982	0.988	90600
Walk	0.999	0.997	0.998	314400
Walk-backward	0.998	0.994	0.996	84600
Walk-circle	0.995	0.999	0.997	117000
Run	0.998	0.999	0.998	212400
Stair-up	0.998	0.997	0.998	279300
Stair-down	0.999	0.996	0.997	274800
Table-tennis	1.000	0.999	0.999	170700
accuracy			0.992	3741000
macro avg	0.990	0.990	0.990	3741000
weighted avg	0.992	0.992	0.992	3741000

Figure 9. (**a**) Confusion matrix; (**b**) Class-wise performance of the Transformer for Human Activity Recognition.

4. Discussion and Conclusions

For human activity recognition, the typically used convolutional neural networks, alone or combined with LSTM, find here a viable alternative, the Transformer model. The transformer is an advantageous alternative to recurrent and convolutional networks. It can scale up the model to more than 1 million parameters and can also be used on mobile devices. It can push the measured signal time series directly (after a normalization) into the neural network, without a necessity of a pre-transformation of the data. Moreover, the transformer is also well parallelized to run on GPU.

HAR achieved 99.2 percent prediction success compared to the original 89.67 percent of KU-HAR work [8]. It successfully coped with the classification of one activity contained in the whole time series as well as with the merging of two activities in one time series. The robustness of the predictions was not even affected by the omission of every second signal measurement from the time series. A new method of signal data augmentation has also been devised, focusing on the logical connections between signals and their appropriate impact to enhance the accuracy of Transformer predictions. The results of the experiments show how the attention mechanisms found correlations in the long time series

of the signal and further promoted the most important of them, which positively affected the classifications of activities.

This paper has successfully demonstrated the benefits and utility of Transformer neural networks in classifying human activities. In the future, the tests should be enlarged to use more kinds of sensor data and the results should be usefully applied, at first for models of robots, which should serve as a springboard for furthermore useful applications involving direct support for humans.

Supplementary Materials: The HAR-Transformer code and further description can be downloaded at: https://github.com/markub3327/HAR-Transformer

Author Contributions: Conceptualization, I.D.L.; methodology, M.K.; software, M.K.; validation, M. K.; formal analysis, J.P.; investigation, M.K. and J.P.; resources I.D.L.; data curation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, J.P.; visualization, M.K.; supervision, I.D.L. and J.P.; project administration, I.D.L.; funding acquisition, I.D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Cultural and Educational Grant Agency MŠVVaŠ SR, the grant number KEGA 012UCM-4/2021, and by the Slovak Research and Development Agency, the grant number APVV-17-0116.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yadav, S. K.; Tiwari, K.; Pandey, H. M.; Akbar, S. A. A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions. Knowledge-Based Systems **2021**, *223*, 106970, DOI: 10.1016/j.knosys.2021.106970.
2. Qin, Z.; Zhang, Y.; Meng, S.; Qin, Z.; Choo, K.K.R. Imaging and fusing time series for wearable sensor-based human activity recognition. Information Fusion **2020**, *53*, 80-87, DOI: 10.1016/j.inffus.2019.06.014.
3. Wang, H.; Zhao, J.; Li, J.; Tian, L.; Tu, P.; Cao, T.; An, Y.; Wang, K.; Li, S. Wearable sensor-based human activity recognition using hybrid deep learning techniques. Security and communication Networks **2020**, DOI: 10.1155/2020/2132138.
4. Gao, W.; Zhang, L.; Huang, W.; Min, F.; He, J.; Song, A. Deep Neural Networks for Sensor-Based Human Activity Recognition Using Selective Kernel Convolution. IEEE Transactions on Instrumentation and Measurement **2021**, *70*, 1-13, Art no. 2512313, DOI: 10.1109/TIM.2021.3102735.
5. TensorFlow: Time series forecasting. Available online: https://www.tensorflow.org/tutorials/structured_data/time_series (accessed 25. 1. 2022).
6. Gupta, S. Deep learning based human activity recognition (HAR) using wearable sensor data. Int. J. of Information Management Data Insights **2021**, *1*(2), 100046, DOI: 10.1016/j.jjimei.2021.100046.
7. Gholamiangonabadi, D.; Kiselov, N.; Grolinger, K. Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection. IEEE Access **2020**, *8*, 133982-133994, DOI: 10.1109/ACCESS.2020.3010715
8. Sikder, N.; Nahid, A.A.; KU-HAR: An open dataset for heterogeneous human activity recognition. Pattern Recognition Letters **2021**, *146*, 46-54, DOI: 10.1016/j.patrec.2021.02.024.
9. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. Advances in neural information processing systems **2017**, *30*.
10. MATLAB Sequence-to-Sequence Classification Using Deep Learning. Available online: https://www.mathworks.com/help/deeplearning/ug/sequence-to-sequence-classification-using-deep-learning.html (accessed 25. 1. 2022).
11. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J. An image is worth 16x16 words: Transformers for image recognition at scale. **2020**, arXiv preprint arXiv:2010.11929.
12. Bao, H.; Dong, L.; Wei, F. Beit: Bert pre-training of image transformers. **2021**, arXiv preprint arXiv:2106.08254.
13. TensorFlow. Available online: https://www.tensorflow.org (accessed 25. 1. 2022).
14. MATLAB Decrease sample rate by integer factor. Available online: https://www.mathworks.com/help/signal/ref/downsample.html (accessed 25. 1. 2022).
15. Atito, S.; Awais, M.; Kittler, J. Sit: Self-supervised vision transformer. **2021**, arXiv preprint arXiv:2104.03602.

 doi:10.20944/preprints202202.0111.v1

16. Brownlee, J. Random Oversampling and Undersampling for Imbalanced Classification. Machine Learning Mastery **2020**, Available online: https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/ (accessed 3. 2. 2022).
17. NumPy. Available online: https://numpy.org (accessed 3. 2. 2022).
18. Pandas. Available online: https://pandas.pydata.org (accessed 3. 2. 2022).
19. Scikit-learn. Available online: https://scikit-learn.org/ (accessed 3. 2. 2022).
20. WanDB: Hyperparameter Tuning. Available online: < https://docs.wandb.ai/guides/sweeps (accessed 3. 2. 2022).