Data Augmentation Strategies for Human Activity Data Using Generative Adversarial Neural Networks

Alexander Hoelzemann  
Ubiquitous Computing  
University of Siegen  
Siegen, Germany  
appliax.hoelzemann@uni-siegen.de

Nimish Sorathiya  
Ubiquitous Computing  
University of Siegen  
Siegen, Germany  
sorathiyanimish9@gmail.com

Kristof Van Laerhoven  
Ubiquitous Computing  
University of Siegen  
Siegen, Germany  
kvl@eti.uni-siegen.de

Abstract—Previous studies have shown that available benchmark datasets from the field of Human Activity Recognition are of limited use for Deep Learning applications. This can be traced back to issues in the quality, the scope, as well as in the variability of the datasets. These limitations often lead to overfitting of networks and thus results that are only conditionally generalizable. One way to counteract this problem is to extend the data by using data augmentation techniques. This paper presents an algorithm and compares two augmentation strategies: (1) user-wise augmentation and (2) fold-wise augmentation to extend the size of a dataset here shown on the PAMAP2 dataset, with an arbitrary number of synthetic samples. These synthesized data resemble the user- and activity-specific characteristics and fit seamlessly into the dataset. They are created by a recurrent Generative Adversarial Network, with both the generator and discriminator modeled by a set of LSTM cells to produce the synthetic time-series data. In our evaluation, we trained four DeepConvLSTM models with supervised learning, three times with a LOSO cross-validation: one baseline model and two times with additional data but different augmentation strategies, as well as one model without cross-validation that monitors the synthesized data quality. The compared augmentation strategies demonstrate the impact as well as the generalized nature of the augmented data. By increasing the size of the dataset by factor 5, we improved the F1-Score by 11.0% with strategy (1) and 5.1% with strategy (2).

Index Terms—Neural networks, Long short term memory, Recurrent neural networks

I. INTRODUCTION

In the context of deep learning and neural network development, having sufficient data to train and test algorithms is crucial for obtaining a high-performance classification model. However, this essential step is often hindered by the fact that not enough data is freely available for many types of applications. For this reason, methods for data synthesis have been developed in the past years. These algorithms are already widely used in computer vision and natural language processing, however, they are still in their early stages of development with regard to activity recognition from wearable inertial sensors. In order to accelerate research in the field of Deep Learning for Human Activity Recognition, it is essential to increase the scope of its public datasets in the future. We already know from other disciplines that more data can lead to more precise results and is an effective tool against overfitting. The further development of augmentation and synthesis algorithms can serve as a catalyst that will enable us to close the current gap in the available data. In this paper, we propose a neural network architecture based on [1] and further developed to generate data for an arbitrary number of samples of sensor based activity data. The network can be trained to synthesize both subject- and activity-specific characteristics. The quantitative enlargement of the dataset improves the classification potential of the neural network on the one hand, and protects it efficiently against overfitting on the other hand. Our tests show that these artificial data can be used to increase classification capabilities of a neural network model, due to an increase of variability and scope of the dataset, but the impact varies depending on how the data was synthesized and merged back into the initial dataset.

II. RELATED WORK

Related work is reviewed by focusing on three disciplines: Human Activity Recognition, Machine Learning for Human Activity Recognition and Data Augmentation for sensor based data.

HUMAN ACTIVITY RECOGNITION (HAR). Experiments with deep learning architectures and human activity data have high demands on the available datasets. These datasets must meet quality requirements to be attractive for deep learning research. The quality is mainly characterised by the scope and variability of the data. A dataset that is too small and doesn’t reach the requirements, tends to underfit the model [2]. Beside of these two very important attributes, also less obvious characteristics are decisive, such as the number of subjects, the number and type of sensors used, as well as their sampling rate and placement on the body [3], label accuracy [4], and synchronisation accuracy between sensors [5]. Regardless of the previously defined experiment protocol selection, its use defines the manner and scope of the individual activities in detail [6].

MACHINE LEARNING FOR HAR. Machine Learning for sensor based human activity recognition has a long tradition. Many published papers in the last two decades have proven its feasibility [7], [8], [9], [10], [11], [12], [13], [14]. While in the beginning most of the publications worked with classical Machine Learning approaches [8], [10], nowadays Deep Learning has replaced classical Machine Learning as the state-
of-the-art learning algorithm [15], [16], because deep learning based classifiers often outperforms classical machine learning approaches. Latest since [12], IMU sensor signals are used as input to train neural networks. However, Deep Learning models have the disadvantage that their success relies heavily on large amounts of data to be able to converge [17], [18].

**Data Augmentation for HAR.** Data Augmentation is one of the standard regularization techniques to prevent neural networks from overfitting [19] and in recent years, it has become an important focus in sensor based human activity recognition research lately. The idea to use synthesized data for training neural networks comes originally from computer vision, e.g. [20], [21]. Traditional transformations of images, e.g. scale, zoom, crop or add noise to the data were adapted and transferred to time-series data by [22]. By applying these techniques, the original data gets slightly modified. In reverse, this also means that we never generate new and unique data. Another approach introduced by Ian Goodfellow [23] to augment data is to use a Generative Adversarial Networks (GAN), like e.g. [24], [1]. Especially [1] is important for our work, since we used this architecture as a baseline architecture, on which our system is built on. The GAN published by Esteban et al. [1] consists of two neural networks. A generator model is used to augment while a discriminator tries to distinguish between real and augmented data. These two networks are trained each other. As soon as the discriminator is not longer able to detect anymore that the produced samples are synthesized, it is assumed that real appearing time-series data is generated. An advantage of this architecture is that we generate new and therefore unique data, thus increasing not only the number but also the variability.

### III. Experiment

The PAMAP2 dataset consists of 19 activities of daily living and was recorded by 9 subjects. The sampling rate of the dataset is 100Hz and the sensitivity ±16g. The sensors are placed on the chest, right ankle, and right wrist [25]. For our experiment we decided to use the protocol subset, since these data were recorded according to a fixed protocol sequence and therefore can be interpreted more uniformly. Furthermore, we limited the subset to the wrist sensor and to activities that are recorded by each subject equally. We decided to not take the null class into account, following the author’s recommendation. Under these conditions, the data is reduced to a subset that contains 8 subjects performing 6 different activities of daily living. Activities that are taken into account are: lying, sitting, standing, walking, vacuum cleaning and ironing.

Our developed approach is depicted as a process cycle and is easy to follow up, see fig. 3. This figure a total of 3 different variations of the dataset that are used or created during the augmentation process. One is the initial dataset, as earlier described. The same data, but organized as LOSO-folds is further referred as α-dataset, which consists of all selected subjects and activities. The α-subset is used to obtain the groundtruth, also called baseline, and serves as the input for synthesizing new data. Since the used protocol-subset of PAMAP2 contains data from 8 different subjects, our α-subset contains 8 folds, where in each of the subsets one subject is excluded and used as the test data. After the augmented data is merged into the α-dataset, the set is referred as β-dataset.

### A. Network Architectures

**DeepConvLSTM**

A DeepConvLSTM architecture is used, see Ordoñez et al. [13], to train four different models. The first model is trained with all subjects and all activities used in this experiment. This network monitors the quality of the generated samples by predicting the samples classes. Another model is trained by using the LOSO cross-validation with the α-subset to obtain the baseline for the final evaluation. A third and fourth model is trained with the β-subset, which contains the LOSO-folds (α-subset) as well as the synthesized data. However, the β-subset differs depending on the chosen augmentation strategy.

**Generative Adversarial Network (GAN)**

The architecture of this work is based on the network introduced as Recurrent GAN [1] and follows an architecture in which both, generator and discriminator, are LSTM-Networks instead of multi-layer perceptrons. The generator network takes random noise at the start of the training. The length of the noise-vector corresponds to the number of timesteps of the LSTM-cell. The discriminator network is used as a binary classifier, which takes the output from the generator as synthetic time series and real data at each LSTM timestep.

Training of the discriminator as a binary classifier minimizes the average negative cross-entropy between the prediction and real labels for both synthetic and real examples. Considering CE as the average cross-entropy between sequences \(X_n\) and \(y_n\), where \(X_n \in \mathbb{R}^{T \times d}\) is the matrix that comprises the output sequence \(T\) from the LSTM cells in the discriminator and \(y_n\), where \(y_n\) can be a vector, 1 or 0; the discriminator loss is given by,

\[
D_{loss}(X_n, y_n) = -\text{CrossEntropy}(\text{LSTM}_D(X_n), y_n)
\]  

(1)

This loss is used by the generator to mislead the discriminator network by producing real-like data that minimize the average negative cross-entropy between the discriminators output on synthetic data and the actual label, considering \(Z_n\) as a sequence of samples from noise space \(z\):

\[
G_{loss}(Z_n) = D_{loss}(\text{LSTM}_G(Z_n), 1) = -\text{CrossEntropy}(\text{LSTM}_D(\text{LSTM}_G(Z_n)), 1)
\]  

(2)

The generator consists of LSTM cells with 100 units as hidden layers and a linear activation function, instead of the tanh-function used by [1]. The discriminator uses the sigmoid-function as the output activation function. Both, generator and discriminator are simultaneously trained at each epoch. Considering the data generating distribution as \(p_z\) from the generative distribution as \(p_g\). After an arbitrary number of timesteps, both of the networks will hold an equilibrium condition and cannot be further improved than \(p_g = p_{data}\).
GAN Network

GAN input

Training Data

minibatch

Optimization

Discriminator Loss

Discriminator Network

minibatch

Generator Loss

Generator Network

Optimization

Noise Space

GAN output

Fig. 1. GAN architecture. Both networks, Generator and Discriminator (orange boxes), consists of 100 hidden LSTM cells. Random noise from noise space is fed as input to the Generator. The trained Generator synthesizes samples as an output. Using cross-entropy loss, both networks are optimized at each time-step. The network will take the original sample windows from \( \alpha \)-subset as an input and generate samples \( \beta \)-subset as output of the network.

B. Methodology

In our experiment, we first trained a model of the DeepConvLSTM with the complete protocol-subset of PAMAP2. This network is trained for 200 epochs and achieved a validation F1 score of 96%. This means that this network knows the characteristics of all subjects and all used activities, therefore it is able to distinguish between real and fake data, and can be used as a model to select just appropriate data generated from the GAN. Important training parameters for the GAN network are: learning-rate = 0.10, Batch-size = 20, latent-dimension (or noise space for generator input) = 10, number of time the generator and discriminator network optimized at each epoch = 5. Due to the unknown number of exactly needed training epoch of the generator and discriminator, in which the GAN starts to produce real looking synthetic data, the network needs to be trained for many epochs. Training the GAN for approximately 1000 epochs is an appropriate estimate to start the process. As soon as fitting hyperparameters and epoch are found, we are able to synthesize an arbitrary number of samples.

Since we are tuning between two networks (generator and discriminator), the discriminator often shows lower loss-values than the generator. Although the generator mislead the discriminator, the produced data does not look realistic. Therefore the synthesized data from the generator is fed for a quality-check to the DeepConvLSTM. If the F1-Score achieves \( \geq 95\% \), the data is considered to have reached the supposed quality and the data will be saved or otherwise discarded. If the required amount of data is reached, the process will be terminated.

Two different strategies, (1) Subject- & Activity-Wise Augmentation and (2) Folder- & Activity-Wise Augmentation, are developed, see fig. 2. Both strategies follow the process-cycle as shown in fig. 3, but differ in the input data of the GAN.

Therefore the augmented data show deviant characteristics.

(1) Subject- & Activity-Wise Augmentation

This strategy uses only the personal activity of a subject as input. The data generated in this way is thus subject-specific. The synthesized data is then added to the subject in the original dataset. Afterwards, the different folds for the cross-validation are generated.

(2) Fold- & Activity-Wise Augmentation

The fold-wise activity selection uses the activity-data of all subjects from a fold as input for the augmentation. The resulting data can no longer be assigned to a specific subject. Rather, it contains characteristics of each subject in the fold. Thus, the data is not assigned to the subjects, but is merged into the folds directly. In contrast, the test dataset is not enlarged as in strategy (1), instead the test subjects of the \( \alpha \)-dataset are used.

Once the subjects and activities to be augmented have been selected, the preprocessing is applied. It is important to note that the data generated by the GAN will be of the same nature as the input data. This means that if raw data has to be generated, the preprocessing must not include operations that alter the raw data itself. Our goal is to produce raw data that appears genuine. Therefore, during preprocessing, we only remove missing values from the dataset and do not apply normalization, even though normalization leads to better classification results.

Afterwards, a jumping window algorithm is applied on both subsets with a window length of 100 samples (1 second in time-domain) and without overlapping samples. The windows are labeled according to the method proposed by [13], where the assigned label of the window is identical to the last sample of the window. These labels are one-hot encoded with 0.0 or 1.0. After merging synthesized and original data. Our final
The data augmentation process itself is divided into 2 phases: (1) Generator phase and (2) Discriminator phase. Figure 3 illustrates the complete augmentation process. A separate GAN must be trained for each activity. If the generated samples cannot be distinguished from real samples anymore, they will be saved, otherwise they will be discarded. Generator and Discriminator networks train parallel on the real data. This process is repeated till the discriminator unit decides that the data looks real. Therefore we talk about this as a cycle with \( n \) iteration steps. However, due to improper parameter tuning of the generator and discriminator, it can happen that the generator mislead the discriminator, which results in unreal looking samples.

We synthesized 80000 samples per class using this method, which is approx. 5 times more than the original dataset. Table I sums up the process depicted in Figure 3 in a compact format and can be used as a progress guide to implement a data augmentation algorithm for sensor based human activity data.

IV. RESULTS

We trained our classification network with both augmentation strategies and compared the results to the baseline, the results are summed up in Table II and visualized as confusion matrices in Figure 4. As shown in the table, we are able to increase the F1-Score by about 5.1% using strategy 2. The subject-specific augmented data, strategy (1), increases the F1-Score by 11.0%. Furthermore, the cross-validation shows that the characteristics of the data of subject 0 and 7 do not seem to match those of the other study participants, emerging in lower classification results. They were only conditionally increased for the classes lying, standing, vacuum cleaning and walking (only strategy 2). The confusion, visible in Figure 4, belongs mostly to these subjects and shows that subject-specific confusion is not solvable by just increasing the number of samples, since even though the amount of samples were increased by factor 5, and the confusion remained. The baseline results of our experiment does not reach results presented in...
TABLE II
PRECISION (P), RECALL (R) AND F1-SCORE (F1) AS RESULTED FROM THE DIFFERENT AUGMENTATION STRATEGIES FOR EVERY ACTIVITY CLASS, AS WELL AS THE CALCULATED WEIGHTED AVERAGE. THE BASELINE WITHOUT DATA AUGMENTATION REACHED AN F1-SCORE OF 67.5%. BOTH AUGMENTATION METHODS IMPROVED THE CLASSIFICATION RESULTS. SUBJECT SPECIFIC AUGMENTATION METHOD IMPROVES THE TOTAL F1-SCORE TO 78.5%. THE FOLD-WISE AUGMENTATION PUSHES THE F1-SCORE TO 72.6%.

TABLE I
PROCESS GUIDE TO AUGMENT DATA EXEMPLARY ON PAMAP2.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
<th>Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Subject and Activity Selection</td>
<td>(1) Select Subjects</td>
<td>(2) Select Activities</td>
<td>protocol subset</td>
</tr>
<tr>
<td>(2) Preprocessing</td>
<td>(1) Delete missing values (Optional) Normalization (2) Create Windows (3) One-Hot-Encoding of labels (4) Create LOSO-Subsets</td>
<td>a-subset (Preprocessed)</td>
<td>If the data is normalized, be sure that the same normalization method is applied on all subsets. We recommend to skip normalization and work with raw data.</td>
</tr>
</tbody>
</table>
| (3) Train Monitoring-Network | Use protocol subset to train the net-work | trained model to test the quality of the augmented data | Over- or underfitting of the Network. If an over- or underfitting of your network already happens with the complete dataset, it will also happen with the reduced [.]. subsets. 
Valid-Data-Quality. Check with cross-validation on the baseline model, to see if the model is over- or underfitted. |
| (4) Train baseline | Calculate the baseline by training a DeepConvLSTM with the a-subset | LOSO-Baseline | 
| (5) Train GANs | Train subjects and activities, for which data should be generated. (2) Decide for an augmentation strategy (3) Train the GAN- Networks and generate augmented data using a-subset | Augmented windows of samples. | The GAN doesn't produce realistic samples at initial steps, although discriminator loss is quite low. Introducing a new activity, results in fine-tuning the parameters at first |
| (6) Merge Data | Merge a-subset with augmented data | β-subset | Not checking the correct metrics with respect to the dataset attributes. |
| (7) LOSO cross-validation | Train/Test Model with β-subset and LOSO-Cross-Validation | Final classification results | 

other papers, for instance [26], [27] that worked with similar architectures and datasets. This is due to the fact that we have limited ourselves to the wrist sensor, as well as the smaller protocol subset and have refrained from preprocessing.

V. CONCLUSION AND DISCUSSION

This paper introduces a new approach to augment sensor based human activity data. The generative part of our architecture works with a Generative Adversarial Network (GAN), which builds on the work of Esteban et al. [1] and is further developed for using inertial data. Our method synthesises data that mimics the input data characteristics. By following two strategies, we are either able to augment subject- or fold-specific activity data. The GAN is able to produce raw data, as well as preprocessed appearing signals. We argue that with the generated data we are able to increase scope and variety of a dataset, which helps to increase classification performance of a neural network and to prevent negative effects, such as over- or underfitting. This work has also shown that adding augmented data could have negative effects on certain classes and subjects. This approach is applicable on an arbitrary number of activities and subjects and can be transferred to other sensor based human activity datasets. Through the presented process cycle, we offer an easy to follow methods that help other scientists to adopt, reproduce and integrate such a method into their own experiments. The architecture of the test network can be exchanged at will and thus be adapted to individual needs. However, the further development of GAN-architecture is necessary to be able to overcome the time consuming disadvantage of choosing the correct hyperparameters. To avoid this factor in the future, we plan to extend the architecture with an independently acting search algorithm to find satisfying hyperparameters for the GAN. In further
Fig. 4. Confusion Matrices of the average classification results from the loso-cross-validation. From left to right: without augmented data, with Activity- & Subject-Wise augmented data, with Fold- & Activity-Wise augmented data.

experiments, it is important to consider how much the size of the augmented dataset influences the classification capabilities of a model. Due to space constraints, those effects were not explored in detail, but they are an important factor for real-world application of such strategies and algorithms.

**ACKNOWLEDGMENT**

This work was partially founded by the BMBF GAIIA Project 01DG20001, AfricaSign.

**REFERENCES**


