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

1st Alexander Hoelzemann
Ubiquitous Computing
University of Siegen
Siegen, Germany
alexander.hoelzemann@uni-siegen.de

1st Nimish Sorathiya
University of Siegen
Siegen, Germany
sorathiyanimish9@gmail.com

2nd 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 to 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 data while a discriminator tries to distinguish between real and augmented data. These two models are training each other. As soon as the discriminator is no 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 ($X_n \in \mathbb{R}^{T*d}$) 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,

$$D_{loss}\ (X_n, y_n)\ =\ -CrossEntropy(\mathrm{LSTM}_D(X_n), y_n) \tag{1}$$

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;

$$\begin{aligned} G_{loss}\ (Z_n)\ &=\ D_{loss}(\mathrm{LSTM}_G(Z_n), 1) \\ &=\ -CrossEntropy(\mathrm{LSTM}_D(\mathrm{LSTM}_G(Z_n)), 1) \end{aligned} \tag{2}$$

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_x 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}.

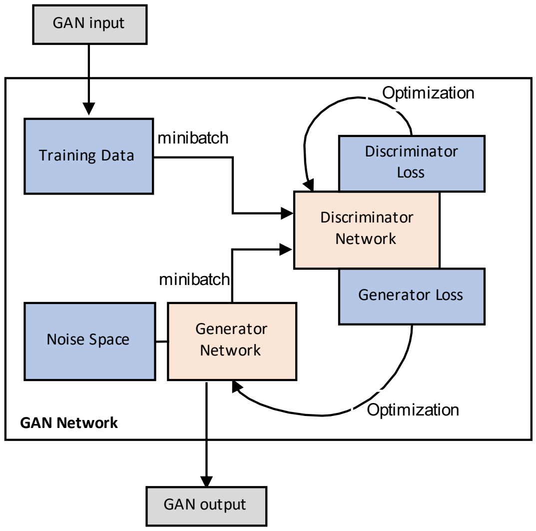

Fig. 1. GAN architecture. Both networks, Generator and Discriminator (orange boxes), consists of 100 hidden LSTM cells. Random noise from noise space is fed as input to the Generator. The trained Generator synthesizes samples as an output. Using cross-entropy loss, both networks are optimized at each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

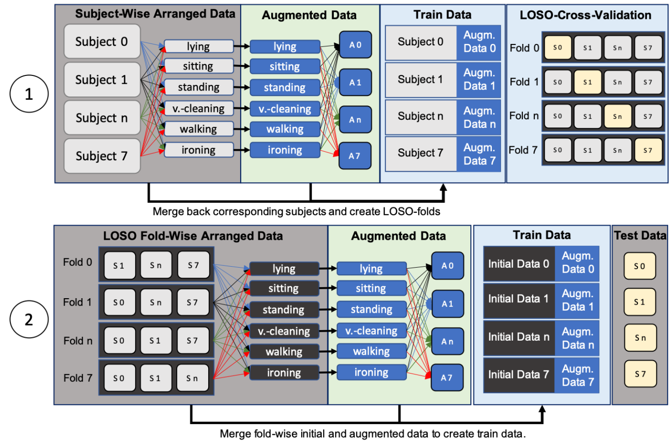

Fig. 2. Data Augmentation Strategies: Grey background represents the initial dataset, green the augmented data and blue the data after merging both (β-dataset). Yellow squares represents the test-subject for each fold. (1) Subject- & Activity-Wise augmentation and merging strategy. After augmenting the data, the augmented data is merged back into the subjects data, afterwards the LOSO-Folds will be created; (2) Fold-Wise augmentation and merging strategy. LOSO-Folds are created before the augmentation process starts. The fold-wise arranged data is then used as input for the augmentation. The synthesized data results in non-subject-specific data, it rather contains characteristics from all subjects of the fold.

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 α-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

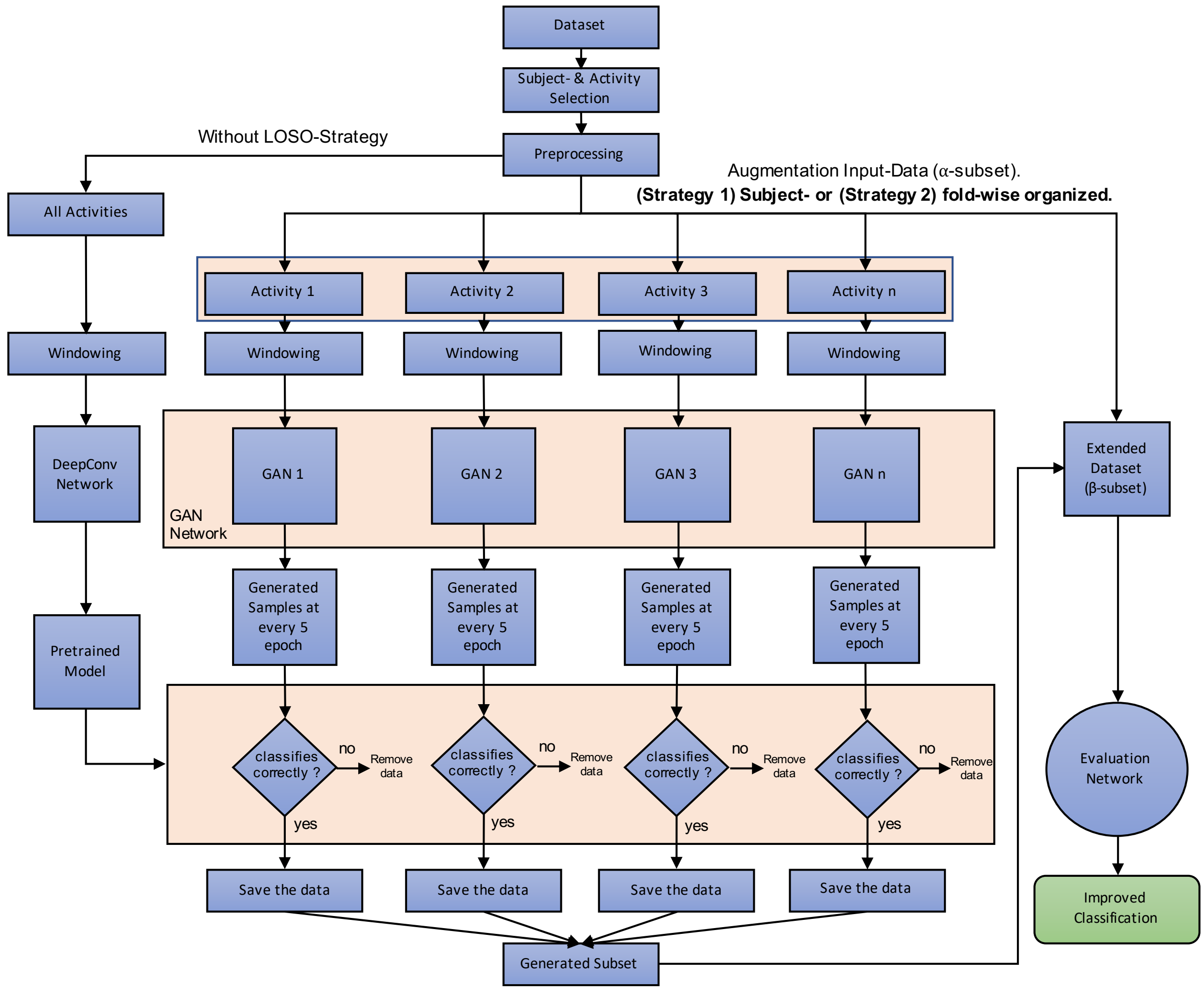

Fig. 3. Data Augmentation Process Cycle: The complete dataset is needed to train the model that monitors the quality of the augmented data. The α-subset represents the input data, the β-subset the dataset after merging the augmented data with the α-subset. For every activity, subject- or fold-wise organized data, a new GAN needs to be trained. After 5 epochs the generated data is tested, if it reaches the predefined F1-Score of 95% it will be kept. If not, the data will be discarded.

subset is called β-subset.

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%.

Fold	**0**			**1**			**2**			**3**			**4**			**5**			**6**			**7**		
Metrics	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Total avg F1-Score: 67.5	**Baseline without Data Augmentation**																							
lying	76	36	49	100	92	96	100	91	95	94	72	81	83	93	88	78	94	86	100	95	97	02	00	01
sitting	00	00	00	99	59	74	90	98	94	95	95	95	91	76	83	99	83	90	99	91	95	68	52	59
standing	00	00	00	20	11	14	97	54	70	79	80	80	50	38	43	84	36	50	88	92	90	19	02	03
ironing	31	100	48	54	95	69	84	71	77	94	93	94	78	89	83	80	89	84	89	97	93	41	49	45
vacuum cleaning	60	79	68	48	83	61	53	89	67	51	64	56	47	60	53	64	89	74	79	73	75	21	80	33
walking	82	84	83	99	50	67	95	93	94	86	88	87	83	72	77	93	95	94	96	97	96	44	01	03
weighted avg	42	50	41	70	64	63	87	83	84	84	83	83	73	73	73	83	81	80	92	92	92	33	30	24
Total avg F1-Score: 78.6	**Subject- & Activity-Wise Augmentation**																							
lying	96	95	95	100	98	99	99	98	99	99	92	95	96	97	97	95	99	97	99	100	100	01	00	01
sitting	00	00	00	99	96	97	95	100	97	98	97	98	98	93	96	100	97	98	99	99	99	85	40	54
standing	69	06	11	94	91	93	95	90	92	93	98	95	81	52	63	94	52	67	94	99	96	08	00	00
ironing	29	100	45	98	98	98	96	85	90	98	97	98	88	97	92	98	96	97	97	98	98	33	46	38
vacuum cleaning	78	89	83	84	95	89	86	96	91	89	92	90	53	89	67	67	91	77	95	90	93	28	88	42
walking	55	16	25	99	94	96	95	97	96	97	96	96	91	60	72	88	98	93	98	97	97	86	61	71
weighted avg	55	51	44	96	95	95	94	94	94	96	95	95	85	81	81	90	89	88	97	97	97	40	39	35
Total avg F1-Score: 72,6	**Fold- & Activity-Wise Augmentation**																							
lying	81	70	75	100	92	96	98	92	95	98	86	92	98	92	95	83	94	89	95	98	97	02	00	01
sitting	00	00	00	90	59	72	87	100	93	94	95	95	93	72	81	80	94	86	97	92	95	59	51	54
standing	08	00	01	74	95	83	92	67	77	85	91	88	48	42	45	83	46	59	80	97	88	43	02	05
ironing	33	100	49	93	91	92	83	57	68	97	92	94	81	87	84	94	82	88	89	97	93	35	38	37
vacuum cleaning	76	73	74	59	79	67	56	86	68	65	75	70	67	60	63	63	86	73	92	48	63	23	79	35
walking	89	86	88	97	83	90	92	93	93	91	89	90	68	91	78	92	94	93	90	97	93	84	37	52
weighted avg	49	56	49	87	84	84	85	83	83	89	88	89	76	76	75	84	83	82	90	89	88	42	35	31

TABLE I
PROCESS GUIDE TO AUGMENT DATA EXEMPLARY ON PAMAP2.

Step	**Action**	**Result**	**Pitfalls**
(1) Subject and Activity Selection	(1) Select Subjects (2) Select Activities	protocol subset	Select Activities or Subjects with insufficiant number of samples.
(2) Preprocessing	(1) Delete missing values (Optional) Normalization (2) Create Windows (3) One-Hot-Encoding of labels (4) Create LOSO-Subsets	α–subset (Preprocessed)	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.
(3a) Train Monitoring-Network	Use protocol subset to train the test-network	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 β-subsets. **Hint:** Quality-check with cross-validation on the baseline model, to see if the model is over- or underfitted.
(3b) Train baseline	Calculate the baseline by training a DeepConvLSTM with the α-subset	LOSO-Baseline	
(3c) Train GANs	(1) Select 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 α-subset	Augmented windoows of samples.	The Generator does not produce realistic samples at initial steps although discriminator loss is quite low. Introducing a new activity, results in fine-tuning the parameters at first
(4) Merge Data	(1) Merge α-subset with augmented data	β-subset	
(5) LOSO cross-validation	(1) Train/Test Model with β-subset and LOSO-Cross-Validation	Final classification results	Not choosing the correct metrics with respect to the dataset attributes.

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 variability 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 hyper parameters. 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

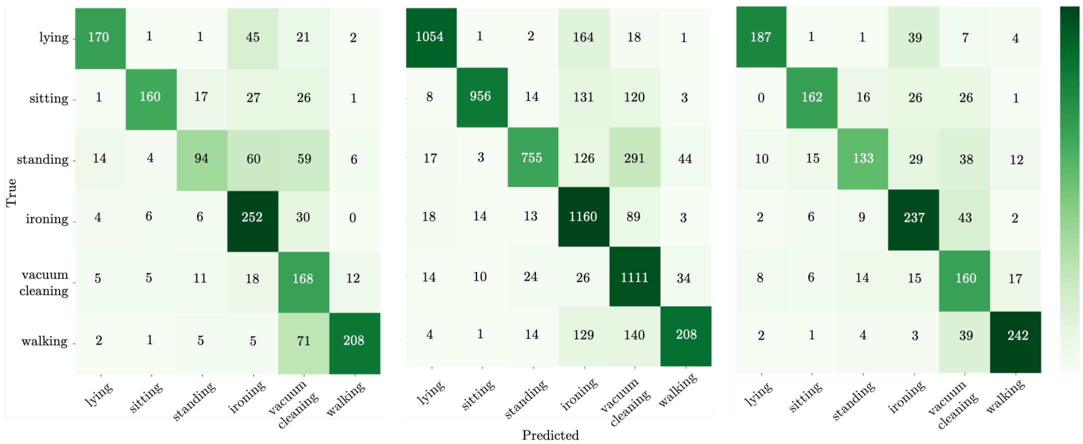

Fig. 4. Confusion Matrices of the average classification results from the loso-cross-validation. From left to right: without augmented data, with Activity- & Subject-Wise augmented data, with Fold- &Activity-Wise augmented data.

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

[1] C. Esteban, S. L. Hyland, and G. Rätsch, “Real-valued (medical) time series generation with recurrent conditional gans,” *arXiv preprint arXiv:1706.02633*, 2017.
[2] W. Koehrsen, “Overfitting vs. underfitting: A complete example,” *Towards Data Science*, 2018.
[3] K. Kunze and P. Lukowicz, “Sensor placement variations in wearable activity recognition,” *IEEE Pervasive Computing*, vol. 13, no. 4, pp. 32–41, 2014.
[4] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, “Accurate activity recognition in a home setting,” in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 1–9, 2008.
[5] A. Hoelzemann, H. Odoemelem, and K. Van Laerhoven, “Using an in-ear wearable to annotate activity data across multiple inertial sensors,” in *Proceedings of the 1st International Workshop on Earable Computing*, pp. 14–19, 2019.
[6] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Comput. Surv.*, vol. 46, Jan. 2014.
[7] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde, “Advanced interaction in context,” in *International Symposium on Handheld and Ubiquitous Computing*, Springer, 1999.
[8] K. Van Laerhoven and O. Cakmakci, “What shall we teach our pants?,” in *Digest of Papers. Fourth International Symposium on Wearable Computers*, IEEE, 2000.
[9] J. Yin, Q. Yang, and J. J. Pan, “Sensor-based abnormal human-activity detection,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008.
[10] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data,” in *International conference on pervasive computing*, pp. 1–17, Springer, 2004.
[11] M. Luštrek and B. Kaluža, “Fall detection and activity recognition with machine learning,” 2008.
[12] N. Y. Hammerla, J. Fisher, P. Andras, L. Rochester, R. Walker, and T. Plötz, “Pd disease state assessment in naturalistic environments using deep learning,” in *Twenty-Ninth AAAI conference on artificial intelligence*, 2015.
[13] F. Ordóñez and D. Roggen, “Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, p. 115, 01 2016.
[14] A. Hoelzemann and K. V. Laerhoven, “Using wrist-worn activity recognition for basketball game analysis,” in *Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction*, ACM, sep 2018.
[15] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, p. 436–444, May 2015.
[16] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.
[17] D. Brain and G. Webb, “On the effect of data set size on bias and variance in classification learning,” in *Proceedings of the Fourth Australian Knowledge Acquisition Workshop, University of New South Wales*, 1999.
[18] M. Najafabadi, F. Villanustre, T. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *Journal of Big Data*, vol. 2, 12 2015.
[19] T. DeVries and G. W. Taylor, “Dataset augmentation in feature space,” *arXiv preprint arXiv:1702.05538*, 2017.
[20] H. Inoue, “Data augmentation by pairing samples for images classification,” *arXiv preprint arXiv:1801.02929*, 2018.
[21] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, “Medical image synthesis for data augmentation and anonymization using generative adversarial networks,” in *International workshop on simulation and synthesis in medical imaging*, pp. 1–11, Springer, 2018.
[22] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, “Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks,” in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI ’17, (New York, NY, USA), p. 216–220, Association for Computing Machinery, 2017.
[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, pp. 2672–2680, 2014.
[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” Nov 2018.
[25] A. Reiss and D. Stricker, “Introducing a new benchmarked dataset for activity monitoring,” 06 2012.
[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*, pp. 56–63, 2018.
[27] N. Y. Hammerla, S. Halloran, and T. Ploetz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” *arXiv:1604.08880 [cs, stat]*, Apr 2016.