# Data Augmentation for Time Series: Traditional vs Generative Models on Capacitive Proximity Time Series

Biying Fu
biying.fu@igd.fraunhofer.de
Fraunhofer Institute for Computer Graphics Research IGD
Darmstadt, Hessen

Florian Kirchbuchner
Fraunhofer Institute for Computer Graphics Research IGD
Darmstadt, Germany
florian.kirchbuchner@igd.fraunhofer.de

Arjan Kuijper
Interactive Graphics Systems Group
Technische Universtität Darmstadt
Darmstadt, Germany
arjan.kuijper@igd.fraunhofer.de

## ABSTRACT

Large labeled quantities and diversities of training data are often needed for supervised, data-based modelling. Data distribution should cover a rich representation to support the generalizability of the trained end-to-end inference model. However, this is often hindered by limited labeled data and the expensive data collection process, especially for human activity recognition tasks. Extensive manual labeling is required. Data augmentation is thus a widely used regularization method for deep learning, especially applied on image data to increase the classification accuracy. But it is less researched for time series. In this paper, we investigate the data augmentation task on continuous capacitive time series with the example on exercise recognition. We show that the traditional data augmentation can enrich the source distribution and thus make the trained inference model more generalized. This further increases the recognition performance for unseen target data around 21.4 percentage points compared to inference model without data augmentation. The generative models such as variational autoencoder or conditional variational autoencoder can further reduce the variance on the target data.

## CCS CONCEPTS

• **Mathematics of computing** → *Variational methods.*

## KEYWORDS

Time series augmentation, Data augmentation, Human activity recognition, Capacitive Proximity Sensing, Ubiquitous Sensing

**ACM Reference Format:**
Biying Fu, Florian Kirchbuchner, and Arjan Kuijper. 2020. Data Augmentation for Time Series: Traditional vs Generative Models on Capacitive Proximity Time Series. In *The 13th PErvasive Technologies Related to Assistive Environments Conference (PETRA '20), June 30-July 3, 2020, Corfu, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3389189.3392606

## 1 INTRODUCTION

In order to train data-based inference models for especially deep learning networks, large quantity of supervised training data will be beneficial [28]. In the domain of computer vision, billions of labelled images can be easily acquired from public available databases. Thus enabling deep learning methods to build successful applications [15] such as image recognition and object localization. However, the data acquisition for human activity data is expensive and the labeling process is error prone. Compared to images, the shortage of labelled data for time series from sensory input is an impeding factor to train generalized inference models. Several minority activities such as *falling* or *running* are much more difficult to collect than *walking* or *sitting*. Therefore, data augmentation should be used to generate synthetic data, especially for such minority classes. The objective of such generative model is to learn the probabilistic distribution of the hidden representation of the activity classes and then generate more new samples similar to the true distribution of the real samples. According to Goodfellow et al. [10], generative model is a method to introduce prior knowledge into the data by transforming the data with methods that preserve the known label information.

Another inherent problem of human activity recognition is its high degree of complexity and uncertainty. It is difficult to build a robust and generalized activity recognition model which suits all individuals. This problem is called user-diversity. The goal is thus to find a robust feature representation to model this diversity with limited data. In this work, we focused on the problem of reducing user-diversity by using data augmentation methods for time series. We call the data of one user group source data, whereas our target data is another unseen user group. By applying data augmentation, we aim to make the distribution of our source distribution wide enough to cover the target distribution and thus make the inference model more powerful, performing on the unseen test data. We distinguish between traditional and generative models. The former includes domain specific methods or geometric modifying methods on the raw sensory input data. We claim that the source distribution will be enhanced by traditional data augmentation techniques on the raw signal space and the generative models such as variational autoencoder (VAE) or conditional variational autoencoder (cVAE) can be used to further decrease the variance on the target data.

Therefore, our objectives of using data augmentation on time series are

- to make the inference model more generalized
- better adaptation to unseen data by modifying the source distribution, i.e. to enrich the source distribution
- to learn a probabilistic distribution of the hidden representation of input data distribution
- use ensemble of generative models to generate more variations of output samples

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*PETRA '20, June 30-July 3, 2020, Corfu, Greece*
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7773-7/20/06…$15.00
https://doi.org/10.1145/3389189.3392606

This paper is organized as follows: we first describes the related work in section 2. Section 3 details various proposed methods separated into classical traditional data augmentation method and generative data augmentation methods such as variational autoencoder and conditional variational autoencoder. An enhancement of using ensemble method to build generative models is also provided in this section. In section 4, experimental setup is described and results are shown based on evaluation with real data from the experiment. Section 5 then discusses the challenges and limitations of the data augmentation methods on time series. Finally section 6 concludes this paper and provides future research directions.

## 2 RELATED WORK

General researches showed that deep learning applications [12] can benefit from data augmentation techniques. Various research papers such as [6] show that data augmentation can improve the generalization capability of deep neural networks, especially in many computer vision tasks such as image recognition and object localization.

Data augmentation can be performed both on the model side or on the data side. On the model itself, Dropout is a common technique to increase the model capacity to adapt to new data samples. Bouthillier et al. [3] proposed to use dropout as a kind of data augmentation technique in the input space without domain knowledge. Adding noise at the output by applying dropout can be viewed implicitly as applying transformation in the input space. Thus projecting the dropout noise back into the input space is viewed as generating augmented versions of the training data. They showed that training a deterministic network with dropout yields similar results as training the network on the augmented samples with the benefit of avoid adding significant computational cost.

Data augmentation on the data side is used as another regularization technique to train robust classifiers which can handle various different shapes of the same input data. The goal is to train classifiers that can handle slight modification of the input without affecting the output of the classifiers. By adding noise, rotating, scaling or cropping an image of a bird won't change the label information. The appearance of it still resembles an image of a bird, unless the object of interest is occluded after the applied transformations. Therefore, data augmentation for images are more intuitive than for time series data. For time series data, this is not as easy. While certain classes carry clear structure in the signal, it is hard to say which transformations will change the original data too much and thus make the label information incorrect. This could affect the inference model in a negative way.

### 2.1 Data Augmentation on Image Data

Generative adversarial networks (GANs) are commonly used to generate synthetic images from unsupervised training data. Frid-Adar et al. [7] compares traditional versus GANs to enlarge the training set with the intention to increase the classification accuracy. The traditional methods are assumed to increase the data size, while GAN network is supposed to increase the diversity to the original data pool. Both methods were applied on a strongly limited amount of computed tomography (CT) images of 182 liver lesions. The classification accuracy of only using classical data augmentation achieved 78 %, while synthetic data augmentation with GAN further increased the classification accuracy to 85.7 %.

To generate samples not randomly, but from predetermined classes, we need to further incorporate conditions in the synthetic generating process. Those approaches are for example conditional generative adversarial networks (cGANs) or conditional variational autoencoders (cVAEs). Mishra et al. [16] uses cVAE to generate images for zero shot learning tasks conditioned on attribute vectors. Evaluating on four benchmark dataset, they demonstrated that their model outperformed the state of the art. Wang et al. [26] leveraged conditional GANs to synthesize high-resolution photo-realistic images conditioned on semantic labels.

### 2.2 Data Augmentation on Time Series

Current researches on data augmentation for time series are most common in feature spaces. Oversampling or under-sampling are popular cases to balance the class distribution. However the former could easily lead to over-fitting since it only duplicate data without modulation and the later will change the input distribution and thus lead to loss of useful information. Synthetic Minority Over-sampling Technique (SMOTE) introduced in year 2002 [5] for synthetic feature generation based on k-nearest neighbours are used to deal with the aforementioned problems by carefully generating new synthetic samples instead of copying.

SMOTE is commonly applied directly in the feature space. According to Wong et al. [27] the classification results can benefit more if the data augmentation technique is applied directly in the original space instead of feature space. The author evaluated on two data augmentation techniques, by creating additional samples with data warping method in the data space and synthetic oversampling in features space. They evaluated the performance on standard MNIST handwritten digit dataset over a range of classifiers. Those include a convolutional backpropagation-trained neural network, a convolutional support vector machine and a convolutional extreme learning machine classifier. Both data augmentation techniques yield an increase in classifier performance. However, it is shown that it is more beneficial to perform the data augmentation in data space, as long as label preserving transformations are possible.

Alzantot el al. [1] used a generative model of stacked LSTM cells combined with a Mixture Density Network (MDN) to synthesize sensory data. To introduce more variability on the generated synthetic data, the MDN network is applied on top of the LSTM architecture. Their objective is to generate fake samples to deceive the discriminator and making it unable to distinguish between generated synthetic samples and the real samples. But their objective is to protect user privacy data by replacing the real data set through a synthetic set and not focus on the task of improving the classification accuracy.

Our goal is to investigate data augmentation techniques on time series in a specific domain with respect to classification performance. The objective of using data augmentation for time series is to be considered as a regularization technique to build more generalized inference model with the ability to cope on unseen target data. To adapt to real application, the data augmentation is used once to train the classifier. The trained model is used in

inference time on new data samples. To preserve the hidden inherent data distribution, we choose variational autoencoder and the conditional variational autoencoder structure as our generative model. The hidden probability distribution of the input data will be approximated by a known function. This approximate function is sampled to reconstruct new samples which are subject to the same inherent data distribution.

## 3 METHODS

In this work, we use data labeled and collected from our previous works [8, 14]. *ExerTrack* is a capacitive proximity sensing system that is able to detect non-stationary exercises. It consists of a regular exercise mat enhanced with eight capacitive proximity sensors attached underneath to perform exercises recognition. The sensing principle is based on active capacitive sensing where the sensor generates a low-frequent quasi-static electric field. The presence of a non-conductive body, such as the human body will affect this static electric field. This modulation in electric field strength caused by body movement with respect to the sensing electrode is measured by the sensing entity.

All exercises corresponding to its raw sensory input time series are depicted in Figure 1. These exercises are periodic in movement that causes periodic modulation in capacitive time series. Opposed to pressure-based sensing technology, no direct contact to the sensing mat is required in our proposed system. It can measure up to an interaction distance of 15 cm. This property of remote sensing enables us to catch more fine-grained body actions close to the sensing device even without contact. The sampling frequency of the operating system is 20 Hz, corresponds to 50 ms each sample. This operation frequency is fast enough to cover the targeted velocity of these eight chosen activities. The time window for each activity is selected to 6 s based on the classification performance from our previous investigation in [14]. This results in a windowed input sample of the dimension 120x8, meaning 20x6 $s = 120$ discrete time samples for all eight sensor channels.

Capacitive time series defines sensor data collected from low-frequent capacitive measurement. In contrast to acceleration data, capacitive data are less noisy and there occurs less abrupt changes in the time-series. Signal curvature is more smooth compared to high frequent signals (e.g. acoustic signals). Thus this type of signal is easy to model, predict, and has less anomaly. Capacitive technology has been widely used in the field of human activity recognition, such as posture detection [25] for healthcare applications, appliances for smart home [4, 9] and wearable applications in form of flexible textiles [19, 21]. However, publicly available databases for capacitive data with respect to human activity recognition is still a niche. Most prominent works collected their own database for prototyping without sharing them with the research community. Thus, our work intends to present first results on the possibility of using simple data augmentation techniques on capacitive time series to increase the diversity of input data in the data space. This method should not only generalize to capacitive time series, but also generalize for other types of signals whose property resembles the capacitive measurements with respect to the signal smoothness and differential in time.

### 3.1 Traditional Data Augmentation

Due to the symmetric system layout of 2x4 sensor placement, one simple method is a domain specific technique. As users are not constrained in which direction the exercises should be performed, the original windowed time series data can be flipped in several ways as depicted in Figure 2. The amount is increased by a factor of four by applying this method.

Other traditional methods implement magnitude and/or time warps. These methods directly modulate the form of the raw input time signal. Magnitude warping multiplies smoothly varying factors with the time series, which can be modeled with piece-wise cubic polynomial functions around 1 as proposed by [24]. We extended this approach to time warping, interpreting the polynomials around 1 as time intervals: if we accumulate them, we receive new indices which can be used to resample the original data and interpret them as samples from the original indices again. This results in smooth warps along the time axis. We can model coarse variations with lower degree polynomials and vice verse. This method is depicted in Figure 3, where a fine time warp is followed by a coarse time warp and subsequently a fine and coarse magnitude warp. The gray dashed line indicates no warp –, where the area above leads to stretching and the area below indicates the squeezing of the time series. These manipulation is applied to the whole input time series before the windowing process to preserve the smoothness of the overall signal. In Table 1 the traditional augmentation methods we evaluate for different classifiers are listed. The $\sigma$ controls the variance of the polynomials around 1, i.e. higher values lead to higher factors and consequently greater distortions.

To visualize the training sample distribution, ISOMAP [2] is used to reduced the high dimensional time series to two dimensional feature distribution space. Isomap is a non-linear dimensionality reduction method for computing a quasi-isometric, low-dimensional embedding for a set of high-dimensional data points. It learns the internal manifold of the high dimensional input data based on a rough estimate of each data point's geodesic distance. We trained the ISOMAP on the original time series and mapped the augmented time series in to the same feature space. In Figure 4 the training data distribution is shown. Different colors represent the individual participants. The '+' indicates the original samples and '*' indicates the augmented samples. Clearly visible is the effect of domain specific augmentation, as it mirrors the original distribution in various ways. Also the general warping shows impressively the effect of enhancing the input distribution.

### 3.2 Generative Models

One of the generative approaches we used is based on the method of variational autoencoder (VAE) proposed by Kingma et al. [11] and Rezende et al. [18]. It is a powerful generative model used to generate fake images [17] or purely synthetic music [22] based on the modeled statistical knowledge about the input space. It can be viewed from the perspective of unsupervised representation learning. By gaining a deep knowledge about the input space even without labels, we will be able to alter, or explore variations on already existing data and in a desired, specific way.

VAE tries to build a model of the input space $p(\vec{X})$ based on the observations of input training data $\vec{X} = \{\vec{x}^{(1)}, \vec{x}^{(2)}, ..., \vec{x}^{(N)}\}$. Here,

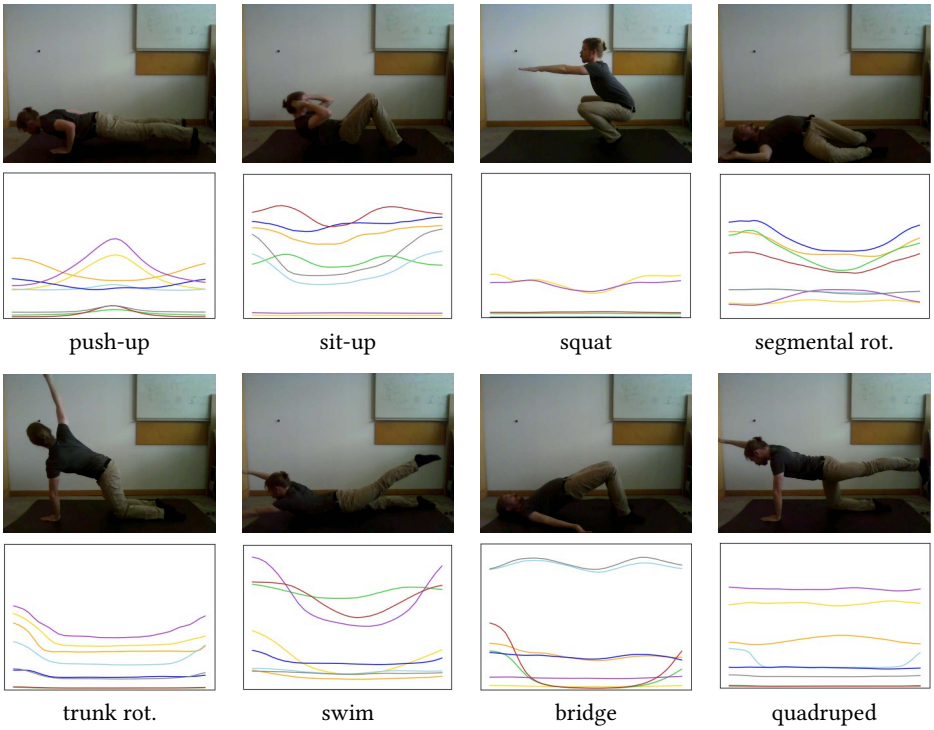



**Figure 1: Figure shows the sport exercises we collected and the capacitive time series with respect to each exercise is shown below [14].**

**Table 1: The tested data augmentation methods for which the evaluation will be carried out. $r \in \{0, \ldots, 3\}$ denotes the original orientation and the three flips. $M_f, M_c, T_f, T_c$ represent fine and coarse magnitude and time warps. The $\sigma$ controls the variance of the polynomials around 1 [14].**

| Augmentation method | Applied transformations | Data increase |
|---|---|---|
| domain | $X_r$ with $r \in \{0, \ldots, 3\}$ | 1 : 4 |
| general | $M_f(M_c(X, \sigma = 0.2), \sigma = 0.1)$ | 1 : 4 |
| | $T_f(T_c(M_f(M_c(X, \sigma = 0.1), \sigma = 0.2), \sigma = 0.2), \sigma = 0.2)$ | |
| | $T_f(T_c(M_f(M_c(X, \sigma = 0.3), \sigma = 0.6), \sigma = 0.6), \sigma = 0.6)$ | |

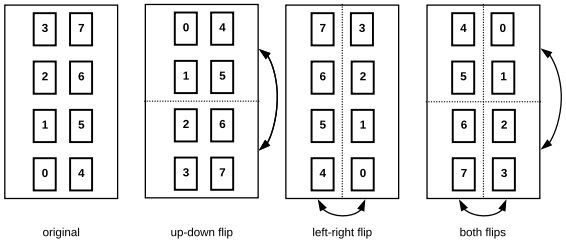



**Figure 2: Illustration of flipping the sensor data, which corresponds to rotating the user on the mat [14].**

the input data are the windowed time series, while the observations are the discrete sampled values. The objective of the VAE is to generate realistic sensor time series similar to samples from the true hidden distribution of the input space. Since the true hidden distribution $p(x)$ is intractable, the task of VAE is to approximate a simple known distribution $q(z)$ which can approximate the true distribution. Then sample from this approximation $q(z)$, we can reconstruct realistic new samples subject to the true sample distribution. The easiest way to do this is by applying the maximum likelihood estimation method (MLE). A latent variable $\vec{z}$ is introduced to marginalize over to get the same distribution as shown in Eq. 1. The symbol $\theta$ represents a possible set of the networks hyper-parameters.

$$\mathcal{L} = \sum_{i=1}^{N} log\, p_\theta(\vec{x}_i) = \sum_{i=1}^{N} log \int p_\theta(\vec{x}_i, \vec{z}) dz \tag{1}$$

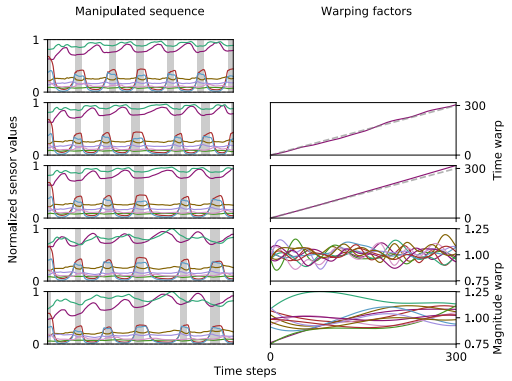



**Figure 3: The process of applying multiple label preserving warps to the same sequence is depicted. The gray areas on the left denote breaks between exercise repetitions. For *time warp* a polynomial is accumulated and interpreted as time intervals to modify all sensor channels [14]. The dashed gray line indicates no modification.**

In general the dimension of the latent variable $\vec{z}$ is smaller than the input variable $\vec{x}$ space. The process of compressing the information from the input variable space $\vec{x}$ to latent representation $\vec{z}$ is called the encoder stage. We can further express the distribution over the input space by applying the Bayes theorem, with the Equation 2.

$$p_\theta(\vec{x}) = \int p_\theta(\vec{x}, \vec{z})dz = \int p_\theta(\vec{x})p_\theta(\vec{x}|\vec{z})dz \tag{2}$$

This allows us to generate new samples conditioned on the latent distribution $z$. If we have the prior probability $p_\theta(\vec{z})$, we can now sample value $\vec{z}$ from this distribution in order to generate the value $\vec{x}$ from the posterior probability $p_\theta(\vec{x}|\vec{z})$ by using a generator network. This is called the decoder stage. To optimize the encoder and decoder stage, we have to minimize the following objective in Eq. 3.

$$p_\theta(\vec{z}|\vec{x}) = \frac{p_\theta(\vec{x}|\vec{z})p_\theta(\vec{z})}{p_\theta(\vec{x})} \tag{3}$$

The loss function of the original VAE is given in Eq. 4. It consists of two separate parts, the first term presents the log-likelihood function which refers to the reconstruction error. The second term is to make the approximated posterior distribution $q(\vec{z}|\vec{x})$ and the model prior $p_\theta(\vec{z})$ more similar to each other. It is called the Kullback-Leibler divergence[13].

$$L(q) = \mathbb{E}_{\vec{z}\sim q(\vec{z}|\vec{x})} log\, p_\theta(\vec{x}|\vec{z}) - D_{KL}(q(\vec{z}|\vec{x})||p_\theta(\vec{z})) \tag{4}$$

As we have very limited training data for deep learning approach, we used a very shallow model with limited capacity. The architecture is illustrated in Figure 5. We reshape the input shape of 120x8 of a windowed input sample to a sequence of the dimension 1x960. In the encoder stage, the input dimension will be reduced in two successive fully connected (fc) layers, each accompanied with a batch-normalization layer, reLu activation layer and one dropout layer. The dropout rate of 0.3 is used to regularize the network from overfitting.

After the latent variable space is learned, the output is generated using the decoder stage. Similar to the encoder stage, we successively increase the dimension of the latent variable $z$ by using two fc layers to generate new input samples. This version of VAE works very fast and takes the overall time series interconnections into consideration. Samples of the generated output of $\vec{x}$ are depicted in Figure 6. The architecture should be adapted to the input space. In case of a more complex input domain, a deeper network structure is to be applied. In order to enrich the input distribution and not just memorizing, we further manipulated the latent space variable before the output generation. Assuming samples surrounding the true sample have similar structures and labels, we added a small normalized Gaussian perturbation on to the multidimensional hidden variable $z$. Other possibility is to set a probabilistic value $p$ to select which dimension should be perturbed by the normalized Gaussian noise. Since the output of the VAE is *shaky*, a smoothing filter in form of a moving average filter with the kernel size of 3 samples was applied to the reconstruction. The width of the smoothing filter is yet another hyper-parameter that can be adjusted according to the generated output data.

In case of conditional variational autoencoder (cVAE), we further included the class label as conditional constraints in the encoder and decoder stage. The cAVE is an extension of VAE. In case of VAE, we trained the generative model in an unsupervised way, therefore we have no control over the data generation process. The objective function for VAE is given in Eq. 5.

$$\begin{aligned} &logP(X) - D_{KL}[Q(z|X)||P(z|X)] = \\ &E[logP(X|z)] - D_{KL}[Q(z|X)||P(z)] \end{aligned} \tag{5}$$

The encoder $Q(z|X)$ is only using training data $X$ without its label information. The decoder $P(X|z)$ is also dependant solely on the latent variable $z$. As a solution, cVAE proposed by Sohn et al. [20] enables to generate new samples with specific attributes. This allows the generative model to generate samples conditioned on specific attributes such as the class labels. The objective function extended to include the class label of the input sample as given in Eq. 6.

$$\begin{aligned} &logP(X|c) - D_{KL}[Q(z|X,c)||P(z|X,c)] = \\ &E[logP(X|z,c)] - D_{KL}[Q(z|X,c)||P(z|c)] \end{aligned} \tag{6}$$

In this way, for each given class label $c = y$, the network is trying to model its own distribution $Q(z|X, c = y)$. Therefore it is a good way to incorporate labels to VAE, if available.

The model structure is similar to VAE illustrated in Figure 5, with the exception, that we now also include the class label $c$ as one-hot vector to the encoder and decoder stage. We merge the one-hot vector with another small dense layer to previous input vector $x$. The latent vector size $z$ remains the same as in case of VAE. But we also merge the class label $c$ to the latent variable $z$ in the reconstruction stage for the new generated output $\hat{x}$. Samples of the generated example outputs of the cVAE model is in Figure 6. On the first two rows, the original signal of the 8 different classes is depicted. Compared to the original data, the next two sets are both generated from VAE and cVAE models based on the same original input with minor modification on the latent space before the reconstruction stage.

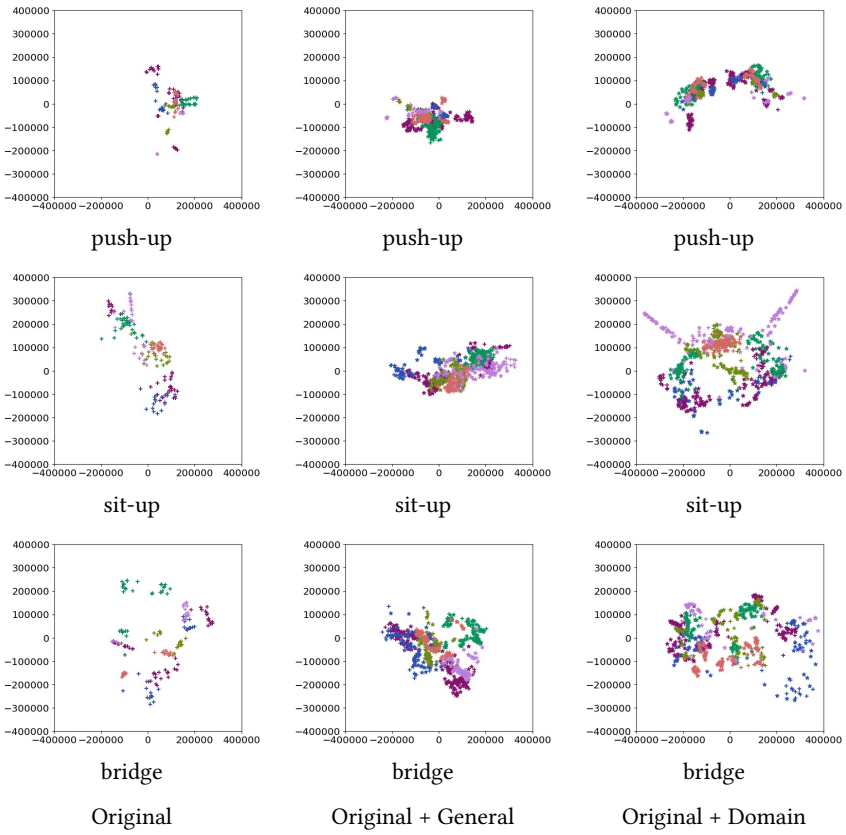



**Figure 4: Two dimensional feature space is visualized for three of the eight sample exercises. The color represents the 6 individuals used for training. The + is the original sample and * indicates the augmented sample. Domain specific data augmentation reflects the assumption of adding flip information by mirroring the sample distribution. Magnitude and Time warp further introduced a more diverse feature distribution.**

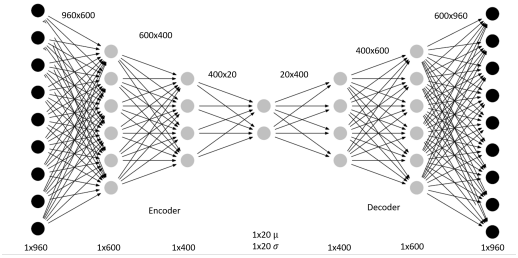



**Figure 5: Pipeline of the variational autoencoder using two successive fully connected (fc) layers is depicted here. Each hidden layer in the encoder and decoder stage is consisted of a batch-normalization, relu activation, and a dropout layer to regularize the VAE network. The latent space is learned in the encoder stage.**

In this network architecture, we try to use Maximum Mean Discrepancy (MMD) [23] loss instead of the Kullback-Leibler divergence. The reason why MMD is favoured to KL divergence is that the MMD is a non-parametric measure. Thus no estimate of parameter from the underlying distribution is required. We can work on dataset directly. MMD represents the distance between two probability distributions by a distance between the means of the embeddings and its variance in the feature space. The equation of measuring the MMD distance is given by Eq. 7,

$$F = \mathbb{E}_{p(z),p(z')}[k(z,z')] + \mathbb{E}_{q(z),q(z')}[k(z,z')] - 2\mathbb{E}_{p(z),q(z')}[k(z,z')] \tag{7}$$

and thus the overall loss function for a MMD-VAE transferred into Equation 8 from the original VAE loss function in Eq. 4.

$$\mathcal{L}_{\text{MMD-VAE}} = \text{MMD}(q_\phi(z)\|p(z)) + \mathbb{E}_{p_{data}(x)}\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \tag{8}$$

We identify that for training each individual model, the performance with MMD loss is higher compared to KL divergence loss. According to [29], the MMD-VAE is always preferred especially for small dataset. Because for small dataset, KL divergence tends to over-fit and generates poor samples, while MMD loss still can generate reasonable samples.

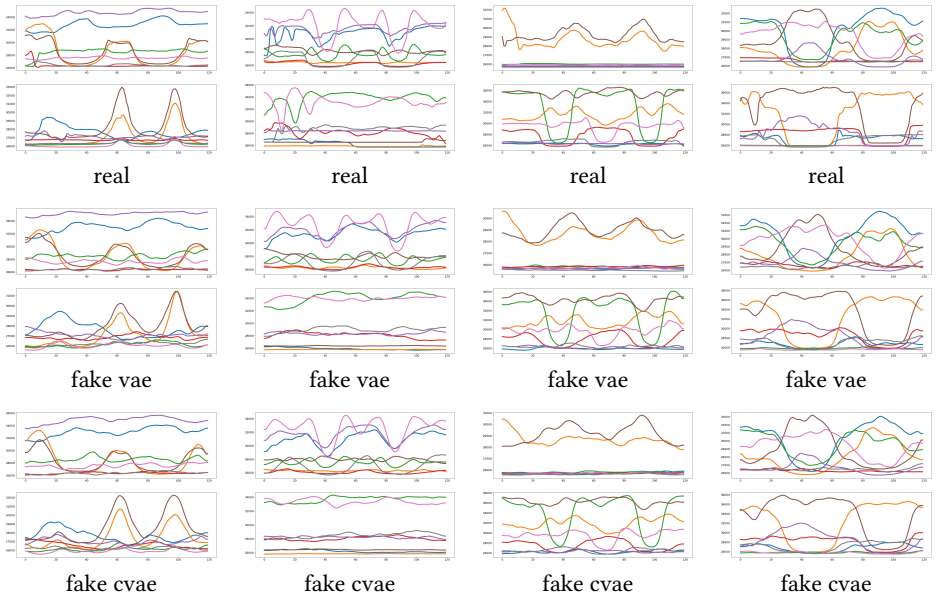



**Figure 6: Sample outputs from VAE and CVAE compared to the real output for the 8 individual exercise classes are shown. The first two rows depict the original samples.**

### 3.3 Ensemble of Generative Models

To wisely composite both characteristics of VAE and cVAE models, we further explore the architecture by combined training of both generative models together in one network. We extended the objective to optimize both loss functions from both models. The combined loss function is presented in Eq. 9.

$$L(q) = L_{rec,CVAE} + L_{KL,CVAE} + L_{rec,VAE} + L_{KL,VAE} \tag{9}$$

The loss function consists of the reconstruction loss of both generative networks and the KL divergence loss to minimize the distance between the probabilistic distribution of the generative output and true input distribution. We trained for 1000 epochs, with the objective to reduce this combined loss function. For combined training of both generative models, the KL divergence converges better than MMD and therefore is a more suitable choice here. The network structure of VAE and cVAE are introduced in the previous section. They have separated structures, but the back-propagation is performed on the ensemble loss function.

The new generated output samples are given by the weighted combination of both generative outputs as in Eq. 10. This weight factor can easily affect the generated signal output by the underlying model.

$$\hat{x}_{rec} = \beta \cdot x_{rec,VAE} + (1 - \beta) \cdot x_{rec,CVAE} \tag{10}$$

In Figure 7, we observe the generated samples with varying weight factor from 0 to 1.0 with a step size of 0.1 each. This effect corresponds to a shift from favoring the VAE at the beginning to favoring the CVAE model at the end. Thus ensemble method can produce more variability and are intended to further increase the classification performance. It also induces more variance in the inference model. Therefore the enhanced performance comes with a price of increased variance. Proper design is required to train the ensemble model and to combine the generative outputs. Researchers should weigh the advantages and disadvantages before applying the ensemble model.

### 3.4 The Evaluation Classifier Architecture

The evaluation architecture used for the classification task is a simple convolutional neural network (CNN) model. We are training inference model from an end-to-end network to avoid handcrafted feature generation. The process of handcrafting features and forming discriminative features to increase the classification performance is strongly related to domain expert knowledge. Thus the model generalization ability is not always guaranteed and is thus out of scope for this paper. Here, we intend to prove the assumption that generating new time series by using data augmentation techniques in data space, can make the inference model perform better on unseen input samples. Convolution filters have proven its superior ability in extracting useful features from each sensor channels. By modifying the convolution filter forms such as adding dilation, we are further able to capture features from across the sensor channels.

The architecture for the evaluation network can be seen in Figure 8. To avoid over-fitting, the L2 regularization is used for the weights in each convolutional layer. Batch-normalization and Dropout Layer with a dropout rate of 0.2 are added after each convolutional layer. The output is a global average pooling (GAP) layer with the number of outputs equal to the classes of activity plus the none class. The none class includes all transitions between useful activity and none activity. The model uses a fixed batch size of 200 and a fixed number of training epochs of 40 for all different experimental setups. The learning rate is fixed at 0.001.

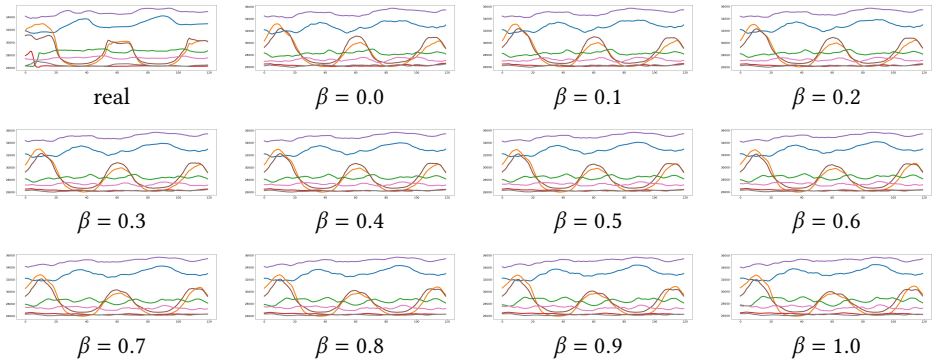



**Figure 7: Sample outputs of the ensemble generative model for the combination with different weight parameter are depicted.**

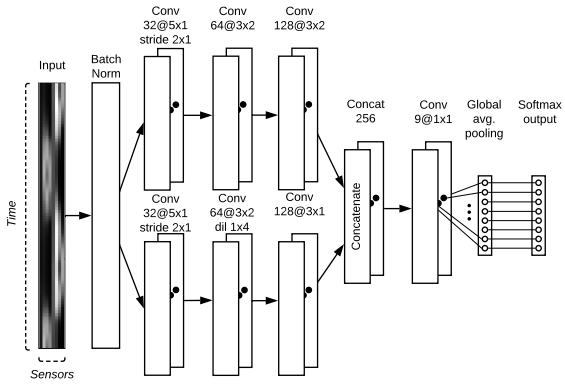



**Figure 8: CNN model used to perform the classification task in the evaluation.**

**Table 2: Table contains the different setups for the conducted experiments.**

| Setting | Training | Test |
|---|---|---|
| 1 | Original (Baseline) | Original |
| 2 | Original + Domain | Original |
| 3 | Original + General | Original |
| 4 | Original + Domain + VAE | Original |
| 5 | Original + Domain + CVAE | Original |
| 6 | Original + Domain + Ensemble | Original |

## 4 EXPERIMENTS AND EVALUATION

To evaluate the proposed methods, we collected data from 9 individuals, each containing two sessions of the 8 different exercises. We randomly selected all sessions from 6 individuals as training set and keep the sessions of the other 3 individuals as holdout set to evaluate the performance of the trained inference model. We used 5-fold cross validation to fine-tune the inference model. The holdout test set is used to measure the performance of the trained model and it's variance on this unseen data. The 6 different setups are listed in Table 2.

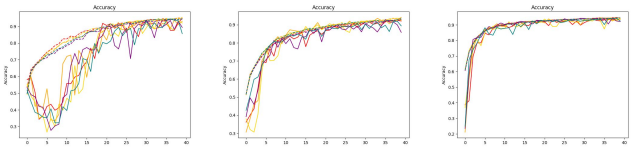
In the second setting, we only use original dataset to include its three flip versions as given in Table 1. This results in a four times augmentation from the original data amount. As in the third setting, the original dataset is warped in time and magnitude with a coarse and dense warp as shown in Table 1 and thus also results in a four times increase. The overall class distribution for the samples remains the same as in the original case.

For the generative model, we used the original dataset including the domain specific augmented samples to train the generative models. In case of VAE, the training is conducted in an unsupervised fashion without the labels of the training data. Whereas in case of conditioned VAE, the training data label is used to condition the generated output signal. The label is included both in the encoder and decoder stage. In this way the generative output is conditioned on the input label. To train the ensemble model, we simultaneously combine the training of both generative models in parallel branches. The training objective is thus to reduce the reconstruction error on both models simultaneously to minimize the Equation 9. The final output generation is concatenated with a weighting factor $\beta = 0.5$ as stated in Equation 10. This can control how strongly each individual generative model will affect the generated output signal.

Now, in Table 3 we list the evaluation results on the holdout test set using the trained inference models based on 5-fold cross validation on the training data. We have chosen the weighted F-Measure (F1 score) to illustrate the classifier performance since the training dataset is not fully balanced. This evaluation measure provides a geometric mean of sensitivity and specificity for each class. As expected, the weighted F1 score on the holdout set is increased by using data augmentation in general. Both warping in the *general* case and integrating additional domain expert knowledge based on the symmetrical system design, have increased the performance by more than 20 percentage points compared to baseline. Further it is interesting to note, that the expected model variance on the holdout set can actually be reduced by using further generative models such as VAE or cVAE models. The highest performance is shown by leveraging ensemble of generative models. This could be explained that each individual generative model captures something unique of the hidden representation space and are able to generate multi-variate output spaces in combination.

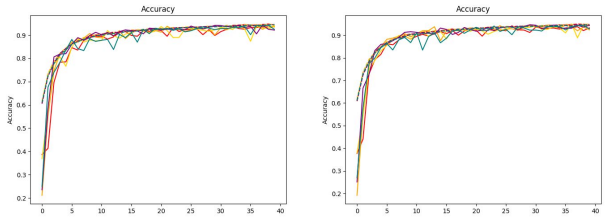
**Table 3: Table contains the F1-score on the unseen test data from the inference models trained using the 5-fold cross validation on the training samples with different settings.**

| Setup | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | $\sigma^2$ |
|---|---|---|---|---|---|---|
| Original (Baseline) | 55.8 % | 55.9 % | 59.5 % | 65.6 % | 60.3 % | 3.59 |
| Original + Domain | 86.3 % | **87.0** % | 85.6 % | 82.0 % | 86.9 % | 1.84 |
| Original + General | 84.2 % | 82.5 % | 84.8 % | **86.7** % | 86.4 % | 1.53 |
| Original + Domain + VAE | 87.8 % | **88.6** % | 88.4 % | 87.1 % | 86.9 % | **0.68** |
| Original + Domain + CVAE | 87.2 % | 86.8 % | 86.8 % | **88.7** % | 88.4 % | **0.81** |
| $(\beta = 0.5)CVAE + (1 - (\beta = 0.5))VAE$ | **88.7** % | **89.0** % | **89.9** % | 83.0 % | 85.0 % | 2.65 |




Original Original + General Original + Domain

**Figure 9: Results show 5 fold cross-validation on training samples from different settings. Increased performance can be seen on models with data augmentation techniques.**




Original + Domain + VAE Original + Domain + CVAE

**Figure 10: Results show 5 fold cross-validation on training samples from different settings. Variance is reduced for the training samples with respect to original data set or data set augmented with traditional data augmentation only.**

Similar results can be seen on the performance curve for the training set. The weighted F1 score on the training and validation set are visualized in the Figure 9 and 10.

## 5 DISCUSSION AND LIMITATION

Labelled data of time series from sensory input is limited and the acquisition process is tedious and expensive. It requires extensive manual labeling. Especially for the task of human activity recognition, not every activity can be equally easily collected. For example it is way difficult to collect as much real falling activities as general activities, such as walking or sleeping. In addition, time sequence cannot be easily shuffled, rotated or permuted without destroying the sequence information. That is why not much research has been conducted on time series augmentation than on data augmentation for images. However, several researches already proved that deep learning can clearly benefit from large quantity and diversity of labeled training data. Therefore it is of interest to confront the problem of time series augmentation.

We identify that special care has to be taken into account by designing network architectures for augmenting time series data. A distorted bird image can much likely still to be recognised as a bird by the human visual than a largely distorted time signal. The degree of distortion is therefore an important hyper-parameter to optimize while augmenting time series to still preserve the original label. Similarly it is the case for the generative models. To generate new realistic samples, we reconstruct slightly modify samples in the learned hidden feature space. How large can the manipulation of the new sample be apart from the mapped original sample in the latent space, without the risk of generating a wrong sample.

Therefore, a careful design is required to apply data augmentation for time series. It could have a negative effect, if the traditional data manipulation is too strong or the generative model is not properly designed. In general, we observe a positive effect by applying traditional data augmentation on time series to enhance the input data distribution and thus increase the model generalization ability on unseen holdout set. The generative models can further reduce the variance on the holdout set. The setup of the ensemble model by wisely fusing both generative models can further increase the performance on the holdout set. The fusion process is considered to combine the learning characteristics of both generative models on the latent space.

## 6 CONCLUSION AND OUTLOOK

Our objective is to show that data augmentation on time series is needed to train more generalized inference models. The data augmentation process is only used in the training phase and thus does not add additional overhead in the inference stage. As for most applications in the human activity recognition with sensory inputs, the amount of labeled data is strongly limited. Data augmentation in data space enables to increase the amount of training data. Traditional methods, such as mirroring the input data or warping the signal, can easily increase the number of samples. Generative methods further adds more diversity to the input data space. The trained classifier thus inherits the ability to adapt to new samples.

To evaluate, we separate our multi-user data in training and holdout set to train and evaluate the inference model. The training set are augmented with various approaches. The holdout set are constructed such that all sessions from these individuals are not included in the training phase. We applied 5-fold cross-validation to train and finetune the inference models. The results show improved

F1 score using the previously trained inference models with the data augmentation methods, either domain specific or by directly manipulating the raw sensory input form. A performance increase of 21.4 and 21.1 percentage points compared to baseline can be achieved with data flipping and warping methods respectively. Thus, our objective that data augmentation is required to better adapt the inference model to unseen data is proven.

Several prominent research works have been introduced for data augmentation with respect to images and natural speeches. Our ideas are strongly inspired by these contributions in the field of computer vision, but are extended to adopt to specific domain related issues. The novelty of our proposed approach in the field of generative data augmentation is to use cVAE to generate time series conditioned to its class affiliation. Existing methods apply data augmentation mostly in an unsupervised way. We also combined both generative methods to increase the diversity by fusing alternative information from both models.

We prove that generative models such as VAEs or cVAEs can be used to learn the hidden representation of the input data distribution and therefore generate fake samples realistic enough to mimic the original data drawn from the same source distribution. Therefore, by adding samples generated from the generative models, we could further decrease the variance of the inference models on the unseen target data distribution without loosing performance. Finally, training ensemble models of generative models can further improve the performance given that the weight parameter on the combination for the reconstruction is chosen properly. However, ensemble model introduces slightly more noise to the inference model for the benefit of increased performance.

In the future, we further investigation on the cost and benefit of using data augmentation for time series is planned. Generalization ability on larger data sources will be considered. Data augmentation can be a valuable solution for limited labeled training data, especially for human activity recognition tasks where the data acquisition process is expensive and the labeling process is difficult.

## REFERENCES


[1] Moustafa Alzantot, Supriyo Chakraborty, and Mani B. Srivastava. 2017. SenseGen: A Deep Learning Architecture for Synthetic Sensor Data Generation. *CoRR* abs/1701.08886 (2017). arXiv:1701.08886 http://arxiv.org/abs/1701.08886

[2] Mukund Balasubramanian and Eric L. Schwartz. 2002. The Isomap Algorithm and Topological Stability. *Science* 295, 5552 (2002), 7–7. https://doi.org/10.1126/science.295.5552.7a arXiv:https://science.sciencemag.org/content/295/5552/7.full.pdf

[3] Xavier Bouthillier, Kishore Konda, Pascal Vincent, and Roland Memisevic. 2015. Dropout as data augmentation. *arXiv preprint arXiv:1506.08700* (2015).

[4] Andreas Braun, Henning Heggen, and Reiner Wichert. 2011. CapFloor–a flexible capacitive indoor localization system. In *International Competition on Evaluating AAL Systems through Competitive Benchmarking*. Springer, 26–35.

[5] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

[6] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2018. Data augmentation using synthetic data for time series classification with deep residual networks. *ArXiv* abs/1808.02455 (2018).

[7] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. 2018. Synthetic data augmentation using GAN for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 289–293.

[8] Biying Fu, Lennart Jarms, Florian Kirchbuchner, and Arjan Kuijper. 2020. ExerTrack—Towards Smart Surfaces to Track Exercises. *Technologies* 8, 1 (2020), 17.

[9] Nan-Wei Gong, Steve Hodges, and Joseph A Paradiso. 2011. Leveraging conductive inkjet technology to build a scalable and versatile surface for ubiquitous sensing. In *Proceedings of the 13th international conference on Ubiquitous computing*. 45–54.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

[11] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12)*. Curran Associates Inc, USA, 1097–1105. http://dl.acm.org/citation.cfm?id=2999134.2999257

[13] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *Ann. Math. Statist.* 22, 1 (03 1951), 79–86. https://doi.org/10.1214/aoms/1177729694

[14] A. Kuijper L. Jarms, B. Fu. 2018. *CapMat for Sport Exercise Recognition and Tracking*. Technical Report. Technische Universit at Darmstadt, Fraunhoferstrasse 5, 64283 Darmstadt. 93 pages.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444. https://doi.org/10.1038/nature14539

[16] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. 2018. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2188–2196.

[17] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. 2016. Variational Autoencoder for Deep Learning of Images, Labels and Captions. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2352–2360.

[18] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Eric P. Xing and Tony Jebara (Eds.), Vol. 32. PMLR, Bejing, China, 1278–1286. http://proceedings.mlr.press/v32/rezende14.html

[19] Gurashish Singh, Alexander Nelson, Ryan Robucci, Chintan Patel, and Nilanjan Banerjee. 2015. Inviz: Low-power personalized gesture recognition using wearable textile capacitive sensor arrays. In *2015 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE, 198–206.

[20] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 3483–3491. http://papers.nips.cc/paper/5775-learning-structured-output-representation-using-deep-conditional-generative-models.pdf

[21] Horia-Nicolai Teodorescu. 2013. Textile-, conductive paint-based wearable devices for physical activity monitoring. In *2013 E-Health and Bioengineering Conference (EHB)*. IEEE, 1–4.

[22] Alexey Tikhonov and Ivan P. Yamshchikov. 2017. Music generation with variational recurrent autoencoder supported by history. *CoRR* abs/1705.05458 (2017). arXiv:1705.05458 http://arxiv.org/abs/1705.05458

[23] Ilya O Tolstikhin, Bharath K. Sriperumbudur, and Bernhard Schölkopf. 2016. Minimax Estimation of Maximum Mean Discrepancy with Radial Kernels. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 1930–1938. http://papers.nips.cc/paper/6483-minimax-estimation-of-maximum-mean-discrepancy-with-radial-kernels.pdf

[24] Terry Taewoong Um, Franz Michael Josef Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data Augmentation of Wearable Sensor Data for Parkinson's Disease Monitoring using Convolutional Neural Networks. (2017), 216–220. https://doi.org/10.1145/3136755.3136817

[25] Miika Valtonen, Lasse Kaila, Jaakko Mäentausta, and Jukka Vanhala. 2011. Unobtrusive human height and posture recognition with a capacitive sensor. *Journal of Ambient Intelligence and Smart Environments* 3, 4 (2011), 305–332.

[26] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8798–8807.

[27] Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. 2016. Understanding data augmentation for classification: when to warp?. In *2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, 1–6.

[28] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. https://arxiv.org/abs/1611.03530

[29] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017. InfoVAE: Information Maximizing Variational Autoencoders. *CoRR* abs/1706.02262 (2017). arXiv:1706.02262 http://arxiv.org/abs/1706.02262