IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 March 2023, accepted 21 April 2023, date of publication 1 May 2023, date of current version 31 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3271409

RESEARCH ARTICLE

ADL-GAN: Data Augmentation to Improve In-the-Wild ADL Recognition Using GANs

APIWAT DITTHAPRON[1], ADAM C. LAMMERT[2], AND EMMANUEL O. AGU[1]
[1]Computer Science Department, Worcester Polytechnic Institute, Worcester, MA 01609, USA
[2]Biomedical Engineering Department, Worcester Polytechnic Institute, Worcester, MA 01609, USA

Corresponding author: Emmanuel O. Agu (emmanuel@wpi.edu)

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Agreement FA8750-18-2-0077, and in part by the High-Performance Computing System acquired through the National Science Foundation (NSF) Major Research Instrumentation Program (MRI) under Grant DMS-1337943.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board at Worcester Polytechnic Institute (WPI).

ABSTRACT The types of Activities of Daily Living (ADL) a person performs or avoids, and underlying patterns can provide insights into physical and mental health, making passive ADL recognition from smartphone sensor data important. However, as people perform ADLs unequally in real life, ADL datasets collected in the wild can be extremely imbalanced, which presents a challenge to Machine Learning (ML) ADL classification. Prior solutions to mitigating imbalance, such as oversampling and instance weighting, reduce but do not completely eliminate the problem. We instead propose ADL-GAN, which utilizes translation Generative Adversarial Networks (GANs), to synthesize smartphone motion and audio sensor data to improve ADL classification performance. ADL-GANs augment the minority ADL of subject *A* by translating real samples from either 1) other ADLs where subject *A* has adequate data in *Context-transfer ADL-GAN* or 2) other subjects with adequate ADL data in *Subject-transfer ADL-GAN*. ADL-GANs utilize multi-domain and contrastive loss functions to perform many-to-many translations between ADL classes and subjects, respectively. Subject-transfer ADL-GAN outperformed baselines and improved balanced accuracy (BA) on an in-the-wild ADL dataset by 27.9 %, while context-transfer ADL-GAN performed best on a scripted dataset, improving the BA of baselines by 9.58 %. The augmented samples from ADL-GANs were shown to be more realistic and diverse than conditional GAN.

INDEX TERMS Activity of daily living, imbalanced class, GAN, data augmentation, smartphones.

I. INTRODUCTION

Activities of Daily Living (ADLs) are routine tasks of daily life that healthy adults can perform without assistance. Examples include walking, sleeping, and bathing. Measuring and analyzing ADL patterns, including their frequency, duration, and regularity, can be utilized for health assessment, especially for the elderly who live independently [1]. When patients are unable to perform basic ADLs, alternative living arrangements (e.g., hospitalization or nursing homes) may be considered [1]. To accurately evaluate the performance of ADLs, the ADL and activity recognition research communities have explored using mobile devices for collecting pervasive sensor data that can be collected passively while users live their lives. Moreover, data collection using mobile devices already owned by users does not require the user to carry an additional device or wear uncomfortable sensors, which facilitates the capture of natural user behavior [2], [3].

Artificial Neural Networks (ANNs) have become a popular method for automatic ADL recognition, Human Activity Recognition (HAR), and Human Context Recognition (HCR).[1] Previous work achieved state-of-the-art ADL recognition accuracy, mainly by employing ANN classifiers [2], [3], [4]. ANNs have demonstrably outperformed traditional machine learning that classifies hand-crafted sensor features

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

[1]Human context usually includes the user's activity but also other attributes such as phone placement and location.

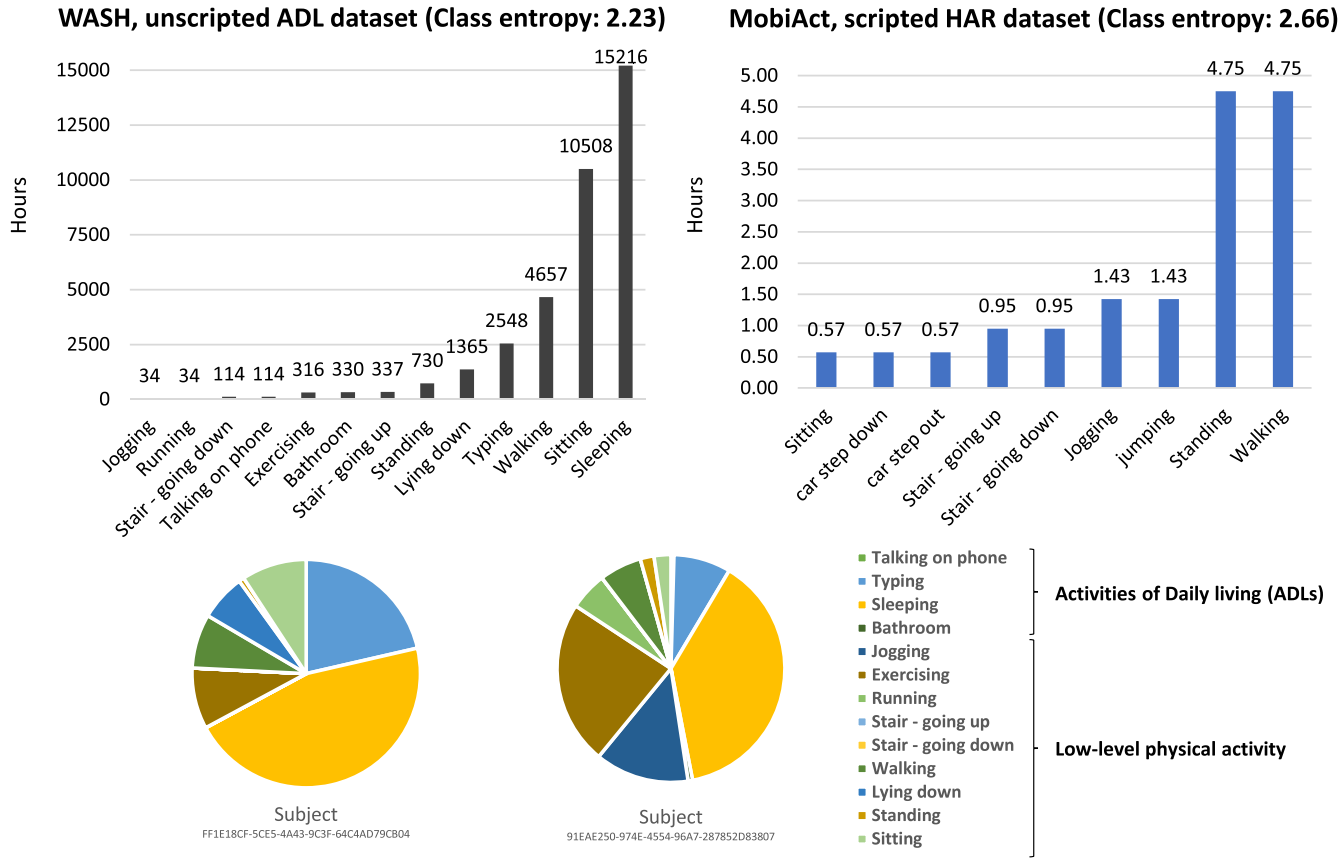


FIGURE 1. **Top: Distribution of the ADL classes in the MobiAct and WASH datasets. Bottom: An example of ADL class labels that may be rare in some subjects, but common in other subjects.**

using algorithms such as Support Vector Machine (SVM) and Random Forest (RF). While early ANNs for ADL recognition also analyzed handcrafted sensor features [2], [5], more recent work [4], [6] analyzed raw data directly using powerful ANNs such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs), which obviate the need for feature engineering step that is error-prone and time-consuming. While significant progress has been made, some in-the-wild ADL recognition challenges remain unsolved, including class imbalance. Unequal distribution of ADL classes is a major challenge in ADL recognition, especially in in-the-wild (realistic) datasets, which is the focus of this paper. Training a classification model on a dataset with unequal numbers of instances per class creates the model that learns to predict the majority class and seldom the minority class [7]. Unlike the other problems caused by errors during data collection, the class imbalance problem is inevitable in realistic ADL datasets as people perform various ADLs at different frequencies. For instance, while some people jog often, others never jog. This extreme imbalance can be observed in Fig. 1, which summarizes the distribution of ADL labels in an HCR dataset gathered in the real world.

To mitigate class imbalance using ANNs, prior work has either included class weights in the loss function [3], [6] or applied data augmentation, such as Synthetic Minority Over-sampling Technique (SMOTE) and Generative Adversarial Network (GAN) [8], [9], [10], [11]. Vaizman et al. [3] and Ge and Agu et al. [6] addressed imbalanced in-the-wild HCR dataset by introducing class weights, which were inversely proportional to the class distribution in order to reduce the attributions of the majority classes. However, weighting class attribution in the loss function frequently causes the HCR model to overfit to the minority class [12].

In this paper, we propose ADL-GAN, a data augmentation method that utilizes GANs to address class imbalance. GANs are able to model the true distribution of the training dataset and generate new samples that improve the decision boundary of the ADL classifier [13]. This is not possible with traditional data augmentation techniques that linearly interpolate or transform real samples [13], [14]. GANs have previously been used to address the class imbalance in various domains, including computer vision [13], speech/audio [15], [16], and sensing [17], achieving noteworthy improvements in performance. For sensor data, various GAN architectures have been proposed to synthesize new samples from a random variable [9], [10], [11], but no prior work explored image-to-image translation GANs that are capable of augmenting higher fidelity samples of multiple classes using a single

GAN [18] whereas previously proposed GANs [9], [10], [11] were only able to augment sensing data of a single activity class, requiring a separate GAN model for each activity class. Moreover, image-to-image translation GANs learn the relationships between two classes instead of learning the complete data distribution of an image class, which is a less complex task and produces a more realistic image than vanilla GAN and ACGAN. Since an image is utilized as source input during inference, the generator model no longer needs to memorize fine details but the high-level representation of the image.

Inspired by image-to-image translation GAN, we propose two ADL-GANs that augment low-level smartphone accelerometer, gyroscope, and audio features corresponding to minority ADL classes. Giving a scenario where subject A has insufficient data for ADL I (minority ADL class), the two ADL-GANs generate synthetic ADL I data for subject A as follows:

- **Context-transfer ADL-GAN** uses data from other contexts/ADLs for which subject A has sufficient data. For example, the *jogging* class of subject A can be augmented using data from the *walking* class of subject A.
- **Subject-transfer ADL-GAN** uses ADL data from other subjects that have sufficient data for that ADL. For example, the *jogging* class of subject A can be augmented using data from the *jogging* class of subject B.

Image-to-image translation GAN was originally proposed to transfer facial expressions such as happy, angry and sad, from one image to another while preserving the identity of faces and the background [18]. This study draws parallels between image-to-image translation and ADL synthetic data generation by considering the subject's context and identity as components of the signals that should be transferred. ADL-GAN learns to transform high-level components of ADL signals such as amplitude, peak values, and frequency in the deep layers of GAN while the shallow layers learn the fine details of sensor data. Specifically, high-level components corresponding to ADL class are transformed by context-transfer ADL-GAN, while the transformation of individual movement patterns, smartphone placement and the recording device's characteristics is learned by subject-transfer ADL-GAN, as visualized in Fig. 3. To translate the signal from one domain to another domain, context-transfer ADL-GAN utilizes one-hot encoding as a conditional input to the GAN, while subject-transfer ADL-GAN utilizes a subject embedding vector obtained from a separate embedding model that learns a representation of the subject domain using a contrastive loss function.

Using ADL-GAN, we answer fundamental questions on whether deriving synthetic data from related real data (subject-wise or ADL-wise) when available, achieves better results than from random noise with condition [17], [19], [20] and without condition [9], [10], [11]. By utilizing a GAN for data translation instead of data generation, the complexity of the task of generating synthetic is significantly reduced and less training data is required compared to previously proposed GANS for sensor data [9], [10], [11]. For instance, subject-transfer ADL-GAN can augment *jogging* class of subject A even though *jogging* class data is missing, deriving the *jogging* class from other subjects that have more samples. We evaluated the two ADL-GANs in terms of their capability to augment realistic and diverse sensor data, and how much the synthetic data they generate improves ADL recognition. Our evaluation explored three state-of-the-art HCR models [2], [3], [6] on two in-the-wild HCR datasets and one scripted ADL dataset collected from smartphone [2], [3]. The two in-the-wild HCR datasets contain activity classes that can be directly utilized for ADL assessment, such as sleeping, bathroom, talking on the phone and typing in unscripted WASH dataset and eating, driving, bathroom, doing laundry, cleaning, working on computer and watching TV in the UCSD ExtraSensory dataset [3] while other activity classes that indicate physical activity levels, such as walking, running, jogging, are also beneficial to the assessment of ADL [21], [22]. To demonstrate the impact of data augmentation on the class imbalance problem, both ADL-GANs were evaluated on datasets with different degrees of class imbalance, as characterized by Shannon entropy [23], a widely used metric to measure information impurity. The impact of data augmentation on each smartphone sensor and the importance of each feature were also examined in this study.

The remainder of the paper is organized as follows: Section II presents background knowledge on GANs. Section III describes prior work related to ADL-GANs. Section IV presents our ADL-GAN approach including training methodology. Section V describes our evaluation of ADL-GAN including datasets, metrics and the ADL recognition models we used with results presented in section VI. Section VII discusses our main findings and section VIII concludes the paper.

II. BACKGROUND

Generative Adversarial Networks (GANs) are generative models that utilize adversarial training of generator ($\mathcal{G}$) and discriminator ($\mathcal{D}$) networks [24]. The generator aims to produce a rich vector $\hat{x}$, typically from a random variable z, $\hat{x} = \mathcal{G}(z)$, in a way that $\hat{x}$ is indistinguishable from observed data x, by the discriminator. This optimization is often referred to as vanilla GAN.

Although traditional GANs can generate raw data from a random variable, it is not possible to control the class generated data corresponds to. ACGAN [25] was proposed to tackle this issue by including an additional variable to control the class of augmented sample. pix2pix, another GAN, considers an image as conditional input in $\mathcal{G}$ and learns the transformation between a pair of images from two different domains, often referred as *image-to-image translation GANs*. However, pix2pix requires samples in both domains to have the same image structure, which is not feasible in many real-world applications. This limitation was overcome in CycleGAN and StarGAN using cycle consistency:

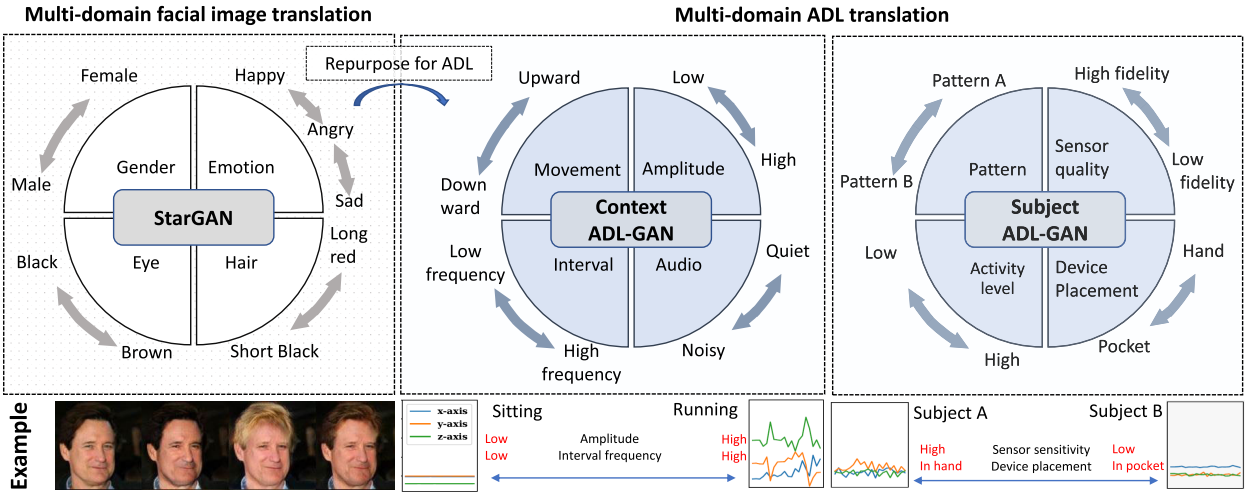


FIGURE 2. **Top: Multi-domain image translation GANs such as StarGAN are repurposed for ADL translation for the purpose of data augmentation by teaching two ADL-GANs to transfer components within the sensor signal from one ADL domain to another. Bottom: Examples of facial image translation using StarGAN [18] (left to right are original, mustache, blond hair and brown hair) and sensing data translations between ADL classes and subjects.**

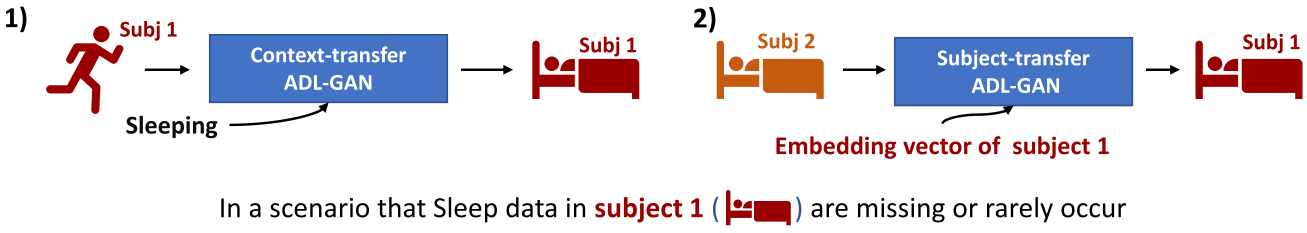


FIGURE 3. ***1) Context-transfer ADL-GAN:*** **One-hot encoding of the target ADL class sleeping is fed into the ADL-GAN to transfer the ADL class.** ***2) Subject-transfer ADL-GAN:*** **An embedding vector of the target subject is fed into the ADL-GAN to transfer the subject.**

A. CycleGAN

CycleGAN [26] introduced a cycle consistency loss function that optimizes parameters in the generator network ($\theta_{\mathcal{G}}$) using $\mathcal{G}_{forward}$ to transfer an image x from the source domain to a target domain with a cycle-transfer back to its original domain using $\mathcal{G}_{backward}$ and image x' from target domain to source domain, as shown in Eq. 1. Cycle consistency is able to capture domain characteristics rather than local feature transfer. Consequently, CycleGAN is considered one of the baselines.

$$\min_{\theta_{\mathcal{G}}} \|\mathcal{G}_{backward}(\mathcal{G}_{forward}(\mathbf{x})) - \mathbf{x}\|_1 \times \|\mathcal{G}_{forward}(\mathcal{G}_{backward}(\mathbf{x}')) - \mathbf{x}'\|_1 \quad (1)$$

B. StarGAN

Image translation using CycleGAN is limited to two specific classes. StarGAN [18] was then proposed for translating images from multiple source domains to any target domain by incorporating a target domain label in the form of a one-hot encoding vector along with the picture that defines the target domain, as an auxiliary input. ADL-GANs use StarGAN's translation concept to transfer the signal between ADL classes in our context-transfer ADL-GAN, and between subjects in subject-transfer ADL-GAN.

III. RELATED WORK

A. CLASS IMBALANCE

Biases caused by imbalance negatively impact ADL recognition [3], [4]. To learn robust representations from the minority class while reducing the influence of the majority class, previous work has proposed several techniques to address class imbalance, which can be categorized into dataset-level and algorithm-level approaches [27] that are expounded on below.

1) DATASET-LEVEL APPROACHES

Balancing class distribution of a dataset can be accomplished by either sampling a small portion from the majority class (undersampling), duplicating samples from the minority class (oversampling), or combinations of both approaches (hybrid). These approaches are effective when using non-complex model that does not require a large number of training samples [7]. Otherwise, overfitting may occur. As machine learning algorithms have become more complex, larger training

sets are required to avoid overfitting. Instead of gathering more data, data augmentation techniques were proposed as a cost-effective, alternative method to increase the training set size for deep learning.

To avoid the overfitting problem caused by oversampling, which duplicates minority class samples, data augmentation is generally applied on new samples to prevent the ANN from learning noise or fine details in the training set. Data augmentation methods are fundamentally categorized into data manipulation methods and deep learning approaches [28]. Data manipulation methods for sensor data include kernel filters, random erasing, signal transformation, and space transformation [29], [30]. These methods aim to prevent the model from learning fine details within the signal. Alternatively, Synthetic Minority Oversampling Technique (SMOTE) [8] was proposed to augment a sample at the feature-level by randomly selecting two real instances from a minority class, which are within the k-nearest points. Interpolation is used to generate a new synthetic point data point between them. However, even though the samples generated by data manipulation methods are unique, they are highly correlated as they transform low-level features. More complex transformation functions are required to prevent the ANN from learning the data augmentation function and to train an ANN to learn important features from the augmented data [27].

Deep learning has achieved remarkable performance on various tasks including data augmentation. Recently, GANs have been proposed as an alternative and powerful generative model. Although the generator in the GAN creates a new image from a random vector that follows Gaussian distribution, which is similar to the autoencoder, the GAN's latent vector can be in any form of data representation, e.g., image (image-to-image translation [26]). This makes a GAN a robust generative model. GAN-based data augmentation has been shown to improve ADL recognition in various works including an increase from 86% to 98% using SVM [9], from 93% to 96% using a CNN-LSTM model [10], from 96% to 98% using a logistic regression model [11]. However, the aforementioned GAN-based data augmentations for HAR were extended from a vanilla GAN that does not have a controllable parameter and, hence, requires a dedicated GAN model for each ADL class.

In this paper, we investigate GAN-based data augmentation of smartphone sensor data using ADL class information as auxiliary inputs, an approach that has not been explored in previous ADL data augmentation work [9], [10], [11]. Moreover, no prior work has investigated GANs to augment in-the-wild datasets that have severe class imbalances and noisy labels.

2) ALGORITHM-LEVEL APPROACHES

In contrast to dataset-level approaches that address class imbalance prior to model training, algorithm-level methods aim to remove the prior class probability during training. In previous HAR and HCR studies, instance-weighting was adopted to balance the cost attributed to each activity class and subject in the objective function [3], [6]. The instance-weighting method was shown to reduce the classification error of class-sensitive metrics. However, prior work did not compare instance-weighting to other approaches.

B. THE USE OF TRANSLATION GANs FOR DATA AUGMENTATION

Translation GANs were previously used to augment or to generate data in many signal-processing domains including audio and speech. However, to the best of our knowledge, this paper is the first work that uses translation GANs to augment in-the-wild sensor data for the ADL recognition task. Similar to image translation GANs, signal translation GANs learn to map input signals from one class to another class using a high-level representation between classes. The development of ADL-GAN was inspired by the following studies. Shahnawazuddin et al. used Parallel-data-free voice conversion [31], a CycleGAN-based method, to augment child speech derived from adult speech for the Automatic Speech Recognition (ASR). By mapping adult speech (majority class) to child speech (minority class), the word recognition error in the child's ASR was reduced by 21.1% [16]. Esmaeilpour et al. proposed an unsupervised learning model for environmental sound classification using a GAN and reported that unsupervised clustering using augmented Mel-spectrograms from CycleGAN improved the results of clustering, outperforming state-of-the-art supervised models [15]. To augment smartphone sensor (accelerometers, gyroscopes, and magnetic sensor) data for road surface assessment, [19] proposed a GAN that synthesized sensor features from a random vector separately for each class. Similarly, [17] trained a Wasserstein GAN separately on each minority class to augment accelerometer features for diagnosing faults in rotating machinery. Both augmentation approaches achieved higher detection accuracies on minority classes.

Our ADL-GAN synthesizes minority class samples to match the number of samples in the majority class. The novelty of ADL-GAN lies in the use of a single GAN to synthesize multiple ADL classes. Prior methods trained separate GANs for each class, which is computationally expensive as they do not adequately exploit relationships between classes, especially for ADL recognition or HCR with a large number of classes. Moreover, we employed a contrastive loss function in the subject-transfer ADL-GAN in order to learn a better representation of subjects. In contrast, StarGAN uses the cross-entropy loss to learn a deterministic mapping of classes, neglecting to exploit the similarity between classes and requiring data from all subjects to be in the training set. In contrast to the image-to-image translation that is performed on two-dimensional data in order to capture spatial information, our ADL-GAN performs in the temporal domain and utilizes sensor data as input, making the data one-dimensional.

TABLE 1. Architecture of generator network.

Layer	Input	Output
Downsampling		
CNN(32,7,1,3),IN,LReLU	(300,19)	(300,32)
CNN(64,4,2,1),IN,LReLU	(300,32)	(150,64)
CNN(128,4,2,1),IN,LReLU	(150,64)	(75,128)
CNN(128,4,2,1),IN,LReLU	(75,128)	(37,128)
Backbone		
Residual-CNN(128,3,1,1),IN,LReLU	(37,128+c)	(37,128)
Residual-CNN(128,3,1,1),IN,LReLU	(37,128+c)	(37,128)
Residual-CNN(128,3,1,1),IN,LReLU	(37,128+c)	(37,128)
Self-attention	(37,128)	(37,128)
Upsampling		
De-CNN(128,7,1,3),IN,LReLU	(37,128)	(75,128)
De-CNN(64,4,2,1),IN,LReLU	(75,128)	(150,64)
De-CNN(32,4,2,1),IN,LReLU	(150,64)	(300,32)
De-CNN(19,7,1,3)	(300,32)	(300,19)

The configuration of CNN and De-CNN are in a (number of filters, filter size, stride, padding) format.

TABLE 2. Architecture of discriminator network.

Layer	Input	Output
CNN(32,4,2,1),LReLU	(300,19)	(150,32)
CNN(64,4,2,1),LReLU	(150,32)	(75,64)
CNN(64,4,2,1),LReLU	(75,64)	(37,64)
CNN(128,4,2,1),LReLU	(37,64)	(18,128)
Maxpooling(5)	(18,128)	(4,128)
Self-attention, Flatten	(4,128)	(512)
FC(64),LReLU	(512)	(64)
FC(c)	64	c

IV. ADL-GAN

A. CONTEXT-TRANSFER ADL-GAN

We propose context-transfer ADL-GAN to augment training data (x_p^c) of the minority ADL class c within the same subject p, i.e. sensor data $x_p^{c_i}$ is transferred to $\hat{x}_p^{c_j}$ where i and j denote the majority and minority classes respectively. The input and output are visualized in Fig. 4. To accomplish this goal, a shallow ADL recognition model in the form of $\mathcal{D}_{cls}$ is included into the loss function $\mathcal{L}_{cls}$ in order to learn multi-class context transfer, together with an adversarial loss $\mathcal{L}_{adv}$. The main components in our context-transfer ADL-GAN are as follows.

1) GENERATOR

Our data generator contains three residual Convolution Neural Network (CNN) blocks with self-attention at the last layer. After each convolution operation, the generator included Instance Normalization (IN) and LeakyReLU (LReLU). Table 1 details the network's architecture, which is grouped into downsampling, backbone and upsampling. The downsampling network uses CNN to learn important features across sensors, which is then fed into the backbone network to extract high-level features. To include the target ADL vector, the vector c was appended before the residual network at the end of the feature dimension. The upsampling part reconstructs the signal in reverse order using a De-Convolution Neural Network (De-CNN) [32], which increases the spatial dimensions of the tensor from the backbone network to match the dimensions of sensing data x. Our generator $\mathcal{G}$ utilizes c_j in the backbone network by concatenating c_j with the output from the previous layer ($[h_p^{c_i}, c_j]$, where h is the output from the hidden layer) and synthesized smartphone sensor data $\hat{x}_p^{c_i} = \mathcal{G}(x_p^{c_i}, c_j)$.

2) DISCRIMINATOR AND ADL CLASSIFIER

Our discriminator and ADL classifiers have the same network architecture but were trained independently. As indicated in Table 2, four layers of CNN, max-pooling, and self-attention are utilized. In the final layer, c is defined as one in the discriminator and defined as the ADL vector size in the ADL classifier with a Softmax activation function. The discriminator $\mathcal{D}_{src}$ is trained to distinguish fake and real samples on a combined set of real and fake samples. Simultaneously, the ADL classifier $\mathcal{D}_{cls}$ learns to classify ADL classes on the same combined set of real and fake samples.

3) ADVERSARIAL TRAINING OBJECTIVE FUNCTION

The fake/real discriminator and generator are optimized using the adversarial loss function ($\mathcal{L}_{adv}$) computed using Eq. 2. For the ADL classifier, parameters in $\mathcal{D}_{cls}$ are updated by optimizing $\mathcal{G}$ and $\mathcal{D}_{src}$ as $\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^r$ and $\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^f + \lambda_{rec}\mathcal{L}_{rec}$, where λ_{cls}, λ_{rec}, and λ_{gp} are hyperparameters that balance the domain classification loss, reconstruction loss and gradient penalty in the Wasserstein GAN.

$$\begin{aligned}\mathcal{L}_{adv} = {} & \mathbb{E}_{\mathbf{x_p}}[\log \mathcal{D}_{src}(\mathbf{x_p^{c_i}})] \\ & + \mathbb{E}_{\mathbf{x_p},c_j}[\log(1 - \mathcal{D}_{src}(\mathcal{G}(\mathbf{x_p^{c_i}}, c_j))] \\ & - \lambda_{gp}\mathbb{E}_{\hat{\mathbf{x}_p}}(\|\nabla_{\hat{\mathbf{x}_p}}\mathcal{D}_{src}(\hat{\mathbf{x}})\|_2)^2 \end{aligned} \tag{2}$$

$$\mathcal{L}_{rec} = \mathbb{E}_{\mathbf{x_p},c_j,c_i}[\|\mathbf{x_p} - \mathcal{G}(\mathcal{G}(\mathbf{x_p^{c_i}}, c_j), c_i)\|_1] \tag{3}$$

B. SUBJECT-TRANSFER ADL-GAN

The subject-transfer ADL-GAN transfers the subject p's activity style to subject q. Specifically, this method augments $x_q^{c_i}$ where c_i is a minority ADL class. To this end, we introduce a subject embedding vector (v_q) into $\mathcal{D}_{cls}$ analogous to speaking style-transfer [33]. Subject embedding vectors are typically applied to speech features in order to recognize subjects in short utterances [34] and, more recently, to aid speech processing applications such as a speaker-aware applications [35]. Representing a subject as an embedding vector makes the transformation applicable to any subject, not just

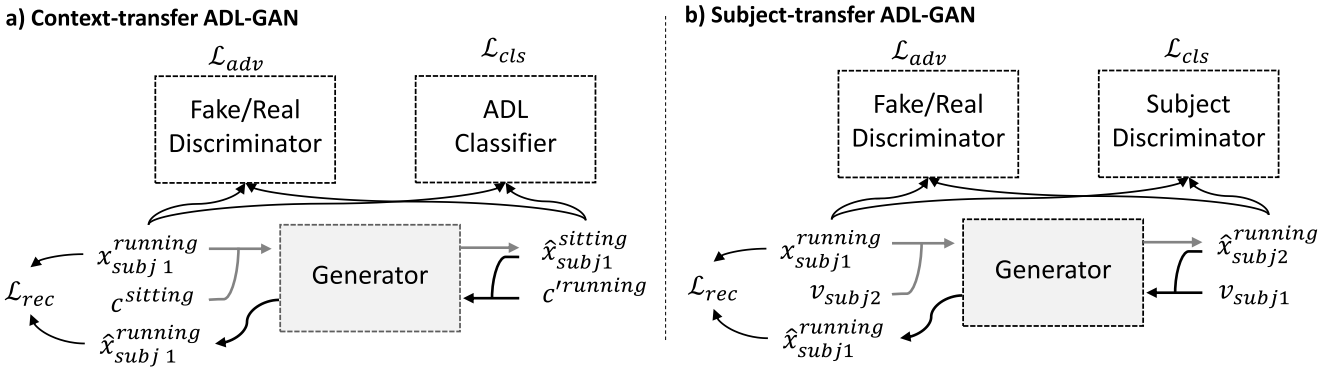


FIGURE 4. Networks in ADL-GAN: a) Context-transfer ADL-GAN, b) Subject-transfer ADL-GAN.

on the training subject as [31] that used one-hot embedding to represent the subject. In addition to MFCC, our subject embedding also considers accelerometer and gyroscope in order to capture body movement features.

Adversarial Training Objective Function: We used the same generator and fake/real discriminator as the context-transfer ADL-GAN. The ADL classifier model was replaced by the subject classifier model and the d-vector v was used as a label instead of c. The D-vector was originally proposed to represent speaker identity for the speaker recognition task, using a latent vector from the last hidden layer of an ANN that was trained on the speaker's speech [36]. We trained the d-vector as a subject embedding vector and used it to control the subject identity of augmented sensing data. Previously, one-hot encoding was used to represent a speaker in StarGAN-based voice transfer [37]. However, considering each speaker as a distinct class does not produce a model that generalizes well between the training and testing sets. Alternatively, the distance between speaker representations was found to better represent the speaker [38]. In this study, GE2E loss function (Eq. 6), an improved contrastive loss [38], was used along with a cosine distance as classifier loss. The loss is defined as in Eq. 4 for the real sample, where v' represents the input subject and Eq. 5 for the generated sample. λ denotes a hyperparameter that allows the learning of embedding vector.

$$\mathcal{L}^{r}_{cls} = \mathbb{E}_{\mathbf{x},v'}[1 - \cos(\mathcal{D}_{cls}(\mathbf{x}), v')] + \lambda\mathcal{L}_{GE2E}(v') \quad (4)$$

$$\mathcal{L}^{f}_{cls} = \mathbb{E}_{\mathbf{x},v}[1 - \cos(\mathcal{D}_{cls}(\mathcal{G}(\mathbf{x}, v), v)] + \lambda\mathcal{L}_{GE2E}(v) \quad (5)$$

The contrastive loss function ($\mathcal{L}_{GE2E}$) uses two learnable parameters, positive weight w and bias b, to scale the cosine similarity between all centroids c. V_{ji} denotes the subject embedding of sample i from subject j. To ensure that $\mathcal{D}_{cls}$ functions properly, samples in a mini-batch must have the same number of samples from each subject j.

$$\mathcal{L}_{GE2E}(v) = \sum_{j,i} 1 - \sigma(w \cdot \cos(v_{ji}, \frac{1}{M-1}\sum_{m,m\neq i} v_{jm}) + b) + \max_{k,k\neq j}\sigma(w \cdot \cos(v_{ji}, c_k + b)) \quad (6)$$

C. DATA OVERSAMPLING USING ADL-GAN

To increase the number of samples in the minority ADL class c of each subject p in the training set, an oversampling strategy was utilized, which resamples each class such that each class has an equal number of samples. Within each minority class, a random sample was chosen to be augmented using two different ADL-GANs as shown in Algorithm 1. ADL-GAN created augmented samples to yield equal distributions of ADL classes and avoid repeated samples in the training set.

V. EVALUATION METHOD

Context-transfer and Subject-transfer ADL-GANs were evaluated as a data augmentation method for ADL recognition

Algorithm 1 ADL Augmentation Using ADL-GAN

Input: ADL feature $\mathbf{x}^{ADL}_{subject}$, ADL label $\mathbf{y}$, ADL class list C, subject class list P and Generator network $\mathcal{G}$ from either context-transfer or subject-transfer ADL-GAN
Output: Augmented feature $\hat{x}$ and ADL label $\hat{y}$
$N \leftarrow \max_{i,j}(||\ x^i_j\ ||)$
$\hat{x}, \hat{y} \leftarrow$ new List, new List
foreach *i in C* **do**
  foreach *j in P* **do**
    for $k \leftarrow 0$ **to** *N-len*($\mathbf{x}^i_j$) **do**
      $x_{real} \leftarrow randomSelect(\mathbf{x}_j - \mathbf{x}^i_j)$
      if $\mathcal{G}$ *is context-transfer ADL-GAN* **then**
        Append $\mathcal{G}(x_{real}, onehot(i))$ to $\hat{x}$
        Append class i to $\hat{y}$
      else if $\mathcal{G}$ *is subject-transfer ADL-GAN* **then**
        $v \leftarrow Dvector(\mathbf{x}_j)$
        Append $\mathcal{G}(x_{real}, v)$ to $\hat{x}$
        Append class i to $\hat{y}$
return $\hat{x}$,$\hat{y}$

randomSelect($\mathbf{x}$) randomly yields one sample x from sensor dataset.
onehot(i) encodes ADL class i into one-hot numeric array.
Dvector($\mathbf{x_j}$) yields a subject embedding vector representing subject j.

with the main objective of solving class imbalance. We augmented accelerometer, gyroscope and audio signals in three corpora that contain smartphone sensor data for HCR and ADL recognition. ADL-GAN was compared against baseline augmentation methods, including dataset-level and algorithm-level, in terms of ADL recognition improvement and synthetic data quality. Furthermore, we investigated the impact of data augmentations on the ADL datasets with different class entropy values and on a subset of smartphone sensors.

A. DATASETS

1) UNSCRIPTED WASH DATASET

Unscripted WASH is an in-the-wild and unscripted HCR dataset gathered as part of the Warfighter Analytics using Smartphones for Health (WASH) project. Unscripted context data were collected from 100 healthy adults aged 18-65 years old at Worcester Polytechnic Institute who gave informed consent for an IRB-approved study. Participants installed a data gathering app on their smartphone, which continuously recorded sensor data from their phone's accelerometer, gyroscope, magnetometer, pressure sensor, light sensor, GPS, WiFi access points, WiFi location, battery life, soft sensor, and audio in natural settings. Periodically, participants were prompted to report their current activities in the mobile application, which were considered as ADL labels in this study. A complete list of label and data distribution are shown in Fig. 1. The WASH dataset also contained phone placement labels but they were not considered in this study.

2) UCSD ExtraSensory DATASET

The ExtraSensory dataset [5] continuously gathered context features from 60 subjects (34 female and 26 male) at the University of California San Diego (UCSD) in-the-wild using 34 iPhone devices and 26 Android smartphones and smartwatches. Participants periodically reported their current context as labels for the context features. The frequency with which ExtraSensory participants and reported various context labels differed significantly between subjects, ranging between 685 and 9,706 samples for one subject. The average duration of participation was 7.6 days, with a standard deviation of 3.2 days. The extraSensory dataset had 51 HCR labels, which includes activity, phone placement, location and processed labels. Since our focus is on ADLs, only 29 activity labels including 7 ADL classes (eating, driving, bathroom, doing laundry, cleaning, working on computer and watching TV) were considered as labels in the evaluation of ADL-GAN.

3) MobiAct DATASET

The MobiAct dataset [2] collected scripted ADL from 50 subjects (42 male and 15 female) ages between 20 and 47 (means: 26) years at the Technological Educational Institute of Crete. Each subject performed the following sequence of ADLs, standing (5m), walking (5m), jogging (30s, 3 trails), jumping (30s, 3 trails), stairs up (10s, 6 trails), stairs down (10s, 6 trails), sit (6s, 6 trails), car step in (6s, 6 trails), car step out (6s, 6 trails), in a specific order that ensured that the dataset was accurately labeled. The MobiAct dataset collected accelerometer and gyroscope data from the same smartphone device, located in the trousers' pocket at random orientations, with a sampling rate of 20 Hz. The MobiAct dataset was included in order to investigate the utility of ADL-GAN on scripted vs. unscripted datasets.

B. DATA PRE-PROCESSING AND FEATURE EXTRACTION AND SELECTION

This study excluded soft features such as time of day and battery percentage features from the ADL-GAN evaluation. Instead, we focused on the tri-axial accelerometer, tri-axial gyroscope, and 13 MFCCs of audio, which were aggregated into a feature vector of 19. Each input sample to the ADL-GAN and ADL recognition models was created to contain 3 seconds of ADL features with a 50% overlap of consecutive windows, according to [6] that found this setup to deliver the best HCR performance. All features were re-sampled to 100 Hz, resulting in a sample size of (300,19). At the subject level, features were normalized using z-score normalization, defined as $(x_p^i - \mu_{X_p})/\sigma_{X_p}$ where μ_{X_p} and σ_{X_p} are mean and standard deviation of all data from subject p. Figure 5 shows our data pre-processing and evaluation pipeline.

C. ADL CLASSIFICATION MODELS

1) ExtraSensory MULTI-LAYER PERCEPTRON (ES-MLP)

ES-MLP is a state-of-the-art multi-sensors HCR, achieving an average Balanced Accuracy (BA) of 77% [3]. The MLP model consists of two hidden layers with 16 units. While the authors extracted a total of 78 handcrafted features from smartphone sensors, this study only considered the subset of their features that can be extracted from accelerometers (26 features), gyroscopes (26 features), and audio MFCCs (26 features). For accelerometers and gyroscopes, 17 features were extracted from the magnitude of 3-axis time series and 3 features from each of the axis, and mean and standard deviation were extracted for each MFCC coefficient as listed in Table 3.

2) MobiAct-MLP (MA-MLP)

Vavoulas et al. rigorously investigated the best feature set for the ADL recognition task. The Optimal Feature Set (OFS) was constructed by recursively eliminating unimportant accelerometer and gyroscope sensor features, and outperformed other feature sets explored for MLP. A total of 90 features were extracted over 5 seconds of signals with 80% overlapping, including those listed in Figure 3 with slop and tilt angle as additional features. A classifier model containing three MLP layers was applied to the OFS. To compare MA-MLP with the other two methods, we extended the OFS to include MFCC features from the ES-MLP.

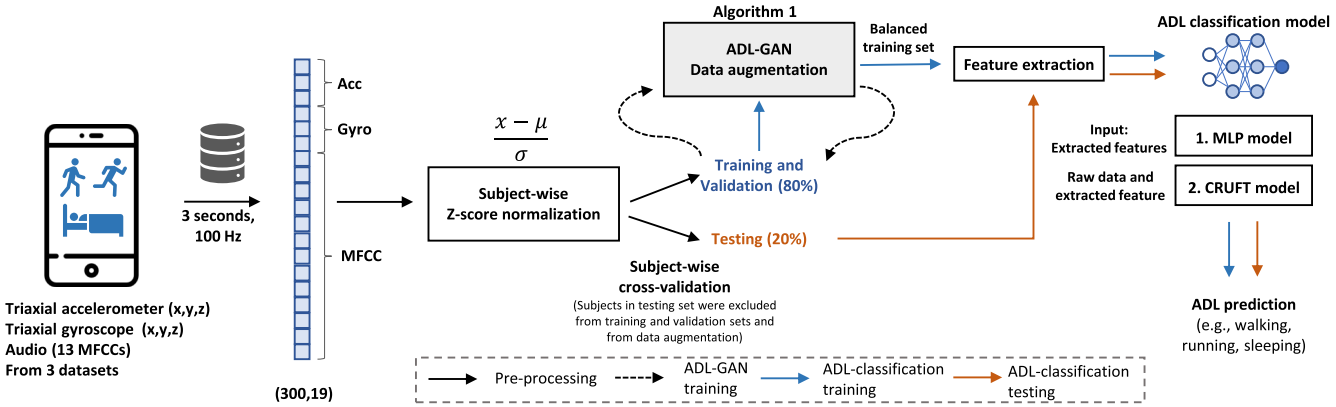


FIGURE 5. Pipeline of data pre-processing and training steps in ADL-GAN and ADL classification.

TABLE 3. Extrasensory's handcrafted features extracted from low-level features.

Features	Description
Accelerometer and Gyroscope Magnitude(m_i)	$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$
1.Mean (μ_{m_i})	$\sum_{i=1}^{N} m_i/N$
2.Standard deviation	$\sqrt{\sum_{i=1}^{N}(m_i - \mu_{m_i})^2/N}$
3.3rd, 4th Moment	$\sqrt[k]{\sum_{i=1}^{N}(m_i - \mu_{m_i})^k/N}, k = 3, 4$
4.Percentile	25, 50, and 75 percentile of m
5.Value-entropy	Histogram of m quantization (20 bins)
6.Time entropy	Normalized m as probability distribution
7.Log energy	Spectral energy in 0–0.5, 0.5–1, 1–3, 3–5 and >5Hz
8.Period	Duration between two peeks of autocorrelations
9.Autocorrelation	Normalized highest autocorrelation of m
Axis feature	Computed for each axis
1.Mean	$\sum_{i=1}^{N} x_i/N$
2.Standard deviation	$\sqrt{\sum_{i=1}^{N}(x_i - \mu_{x_i})^2/N}$
3.Inter-axis correlation	$Ro_{xy} = Cov_{xy}/\sqrt{Cov_{xx} \times Cov_{yy}}$
MFCC	13 coefficients $(c_1, c_2, c_3, ..., c_{13})$
1.Mean	$\sum_{i=1}^{N} c_{k,i}/N$ for $k = 1 - 13$
2.Standard deviation	$\sqrt{\sum_{i=1}^{N}(c_{k,i} - \mu_{c_{k,i}})^2/N}$ for $k = 1 - 13$

3) CONTEXT RECOGNITION UNDER UNCERTAINTY USING FUSION AND TEMPORAL LEARNING (CRUFT)

CRUFT [6] is a state-of-the-art method proposed originally for HCR. It uses a joint learning model with two arms: One arm has an MLP to analyze high-level features and a second arm with deep CNNs that analyzes raw accelerometer and gyroscope data. The model was trained using multi-task learning with Mean Squared Error (MSE) loss for predicting tasks and negative log-likelihood for predicting uncertainty, which was derived from the mean and variance of 5 consecutive ADL labels. The MLP part of CRUFT has one layer of 64 units, which is connected to the last layer of the deep CNN network. The deep CNN network is comprised of a 4-layer CNN with max-pooling after the first and third layers. Following the CNNs, an attention layer with max-pooling and mean pooling is included prior to a bi-directional Long Short Term Memory (bi-LSTM) layer. The output of LSTM is then fused with MLP output to predict the context class and class uncertainty. In rigorous evaluation, CRUFT achieved an HCR Balanced Accuracy of 94.3% on the *scripted* WASH dataset, outperforming the ExtraSensory model that only uses MLP by 2.72%.

D. BASELINE CLASS IMBALANCE MITIGATION METHODS

1) INSTANCE-WEIGHTING

was used to mitigate imbalance in data input to both the ExtraSensory MLP [3] and CRUFT [6] models. Vaizman et al. showed that instance-weighting improved

HCR accuracy from 55.2% to 77.3% [3]. The loss function of each instance was multiplied with a sample weight $w_{p,i}$, computed as 1/Number of samples in class i of subject p.

2) SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE (SMOTE)

generates minority-class samples from real samples within its k-nearest neighborhood [8]. Previous work [14] applied SMOTE on the ExtraSensory feature set and reported an improvement from 65% to 76% in classification F1-score using MLP. SMOTE was applied in this study in the same fashion as [14]. It is worth noting that SMOTE may not exhibit such an improvement on raw sensor data as observed on high-level features as is the case in the CRUFT model.

3) SIGNAL TRANSFORMATION

Smartphone sensor data can be augmented using simple signal transformation. For MFCC, we applied SpecAugment [29], which performs time warping, frequency masking and time masking, on the time–frequency (Mel-spectrogram) domain. For data augmentations of triaxial accelerometer and gyroscope features, we simply applied three transformations previously proposed for wearable sensor data [30]: rotation, permutation, and time-warping. We picked these three transformations (out of seven) due to a better performance found in our experiments and in [30]. Specifically, each sensor feature was rotated using a rotation matrix with a uniformly random angle. Permutation was used to manipulate the sample by segmenting data into five equal segments and permuting them. Finally, the time-warping (f_t) factor was applied using a random warping scale.

4) ActivityGAN [11]

ActivityGAN utilized vanilla GAN to augment sensor data from a random noise vector. The generator of ActivityGAN consists of 5 De-CNN layers and 5 CNN layers with LReLu activation except for the last layer that uses Sigmoid. ActivityGAN was originally developed to generate the amplitude of the accelerometer over 100 timestamps. In this evaluation, we modified them to augment 19 features over 300 temporal steps by increasing the stride and padding factor of De-CNN layers. For the discriminator, we adopted the use of 3 2D-CNN layers with a modification on the input layers from 1 feature to 19 features. Each ADL class was trained separately as a vanilla GAN learns data distribution solely from the training data without conditional variables.

5) AUXILIARY CLASSIFIER GENERATIVE ADVERSARIAL NETWORK (ACGAN) [25]

ACGAN is a conditional GAN that takes a random vector as input with auxiliary conditional arguments, i.e., ADL class and subject embedding vectors. ACGAN was implemented for ADL sensor data augmentation with model architecture similar to ADL-GANs as in Table 1. Specifically, the generator is composed of backbone and upsampling parts where "128+c" of input size in Table 1 combines 64-dimensional random vector z, 64-dimensional subject embedding vector v, and ADL vector of c.

6) CycleGAN

In order for CycleGAN to learn transformations between all possible pairs of ADL classes in an n-class dataset, $n(n-1)/2$ CycleGANs would be required. For the WASH dataset that has 13 ADL classes, 78 separate CycleGAN models need to be trained. The models were implemented using the same generator and discriminator as the context-transfer ADL-GAN but one-hot encoding and ADL classifier network were not used.

7) StarGANv2 [39]

is an improvement on StarGAN, which was proposed to capture the data distribution of multiple domains whereas StarGAN depends on a discrete label using a diversity regularizer that encourages the generator to produce more diverse images. We included StarGANv2 as one of our baseline models during evaluation as it has performed well in translating images between multiple domains. However, we found its mapping function inferior to one-hot-encoding, which we used instead in ADL-GAN.

E. IMPLEMENTATION

The implementation of ADL-GAN[2] was done on PyTorch 1.4 [40] with evaluation performed on NVIDIA Tesla V100 and A100 GPUs. Five-fold cross-validation with subject-wise splitting was utilized with 60% for training, 20% for validation, and 20% for testing. The validation set was used to tune network configurations and hyperparameters such as learning rate and decay rate via grid-search. Only training and validation sets were used in the ADL-GAN data augmentation method. The testing set was held out and only used to evaluate the ADL classification without applying any data augmentation. The Adam optimizer [41] was used to train the models as follows.

ADL-GAN was trained for 1,000,000 iterations (30-40 hours) with a batch size of 8 samples, randomly drawn from different subjects. Generator parameters were updated every 10 iterations while the real/fake discriminator and ADL classifier were updated every iteration. A learning rate of 0.0002 was used with a decay rate of $1e^{-5}$.

ExtraSensory MLP was trained for 50 epochs with a batch size of 300. A learning rate of 0.002 was applied with a decay rate of $5e^{-4}$. Cross-entropy loss was used to train the model.

CRUFT was trained using a batch size of 128 for 100 epochs. A learning rate of 0.001 was used with $1e^{-5}$ decay rate.

Subject embedding vector was implemented based on d-vector as in [38]. We considered 200 ms with 50% overlap of MFCC, gyroscope and accelerometer as an instance. The D-vector model was trained for 1,000,000 iterations using a

[2]Python code: github.com/ADL-GAN/ADL-GAN

batch size of 640, comprising 10 samples from each of the 64 subjects from the training set. The split testing set was not used in this step. A learning rate of 0.002 was applied with a decay rate of $1e^{-5}$.

F. METRICS

1) ADL CLASSIFICATION

The proposed methods were evaluated using the Balanced Accuracy (BA $= \sum_i^c (Sensitivity_i + Specificity_i)/2c$) and F1-score metrics (F1=$2(Precision \times Recall)/(Precision + Recall)$), computed over c ADL classes. The standard error for each measurement is in parentheses.

2) GAN METRICS

The quality of the generated image was measured using the Inception Score (IS) [42] and Fréchet Inception Distance (FID) [43]. IS and FID were proposed as alternatives to a subjective human evaluation of GAN-generated images and indicate correlations with human annotation. Both metrics are based on the recognition performance of the pre-trained model, where the CRUFT model was trained on real data as a reference model. IS was computed on the predicted class probability using Kullback–Leibler divergence (D_{KL}) following $IS = exp(\mathbb{E}_{X\sim p_f} D_{KL}(p(y \mid X) \mid\mid p(y)))$, where $X \sim p_f$ are fake data. FID was computed from the mean and variance of values in the LSTM layer of the CRUFT model from fake (μ_f, Σ_f) and real data (μ_r, Σ_r), as $FID = \|\mu_r - \mu_f\|_2^2 + tr(\Sigma_r + \Sigma_f - 2\sqrt{\Sigma_r \Sigma_f})$.

3) SHANNON ENTROPY

Class imbalance can be considered as an impurity in class distribution that can be measured by the entropy metric. Shannon entropy [23] was utilized as an uncertainty measurement in class distribution. A lower Shannon entropy corresponds to certain information, which means class distribution is skewed to the majority class while a higher Shannon entropy indicates uncertain information, which means class distribution is more evenly distributed [44]. Shannon entropy can be defined as $\frac{1}{n}\sum_{i=1}^{n} -p(c_i)\log p(c_i)$ where $p(c_i)$ is probability distribution of class i.

Subsets of the WASH and MobiAct datasets were created with Shannon entropy values ranging from 2 to the log(1/number of class). Samples of the minority class were randomly subsampled to create a subset with lower entropy (more imbalanced) whereas samples in the majority class were randomly subsampled to create a subset with higher entropy (more balanced). The MobiAct dataset was selected to be compared with the WASH dataset because it has a similar number of classes (13 classes in the WASH dataset and 9 classes in the MobiAct dataset). The subsampling was repeated 30 times and applied in the same ADL recognition pipeline, but filter sizes in the CNN and LSTM were reduced by half to prevent overfitting that occurs due to sub-sampling methods.

VI. RESULTS

A. DATA AUGMENTATION USING ADL-GAN

Based on the BA in Table 4 and the F1-score in Table 5, we found that by augmenting ADL features using ADL-GAN and combining them with real data in the training set, ADL classification performance was improved in the WASH, ExtraSensory and MobiAct datasets using the CRUFT and MLP models. Compared to ACGAN and ActivityGAN, augmenting sensor data using translation GANs (ADL-GAN and CycleGAN) significantly improves ADL classification accuracy. Subject-transfer ADL-GAN increases the BA in the WASH dataset by 27.9 and statistically outperforms SMOTE, a popular data-level augmentation method, as well as instance-weighting, a commonly used algorithm-level technique for mitigating imbalance in datasets. In the MobiAct dataset, context-transfer ADL-GAN outperforms subject-transfer ADL-GAN and all baselines. This breakthrough performance may be because the MobiAct dataset is a scripted dataset – the training of context-transfer ADL-GAN utilizes class embedding where accurate label benefits the algorithms – and is more balanced than the other two datasets. In comparison to CycleGAN, context-transfer ADL-GAN is competitive with the CycleGAN that utilizes 78 GANs to achieve transfer between 13 ADL classes of the WASH dataset whereas only one ADL-GAN is used in the context-transfer ADL-GAN, which requires computing resources up to 17 times less than CycleGAN. This result indicates that one-hot encoding of the class is sufficient to map the distribution in the WASH dataset. In MLP, a better BA is achieved by the subject-transfer ADL-GAN but the differences between the methods with respect to baselines are not statistically significant. The classification improvement on the ExtraSensory dataset is significantly less than that in the WASH dataset, possibly because the ExtraSensory dataset contains up to three times the number of ADL classes as the WASH dataset. However, the ExtraSensory dataset exhibits the same trend of improvement as the WASH dataset. We did not perform ActivityGAN and CycleGAN on the Extrasensory dataset due to the large number of classes in the Extrasensory dataset.

Based on the quantitative measures of GAN performance in Table 6, subject-transfer ADL-GAN has the highest IS, indicating that the generated sample is more distinct, especially for the ExtraSensory dataset. According to the FID metric, CycleGAN had a lower distance between fake and real samples in the WASH dataset, however, with a significantly longer training time. The translation GANs (ADL-GAN and CycleGAN) outperformed ACGAN and ActivityGAN for both metrics.

The improvement from augmenting each ADL class in the WASH dataset is summarized in Table 7. Subject-transfer ADL-GAN improves classification accuracy in most classes, followed by Context-transfer ADL-GAN. Without data augmentation, the sensitivities of most minority classes are lower than those of the majority classes. Instance-weighting considerably improves the sensitivities of minority classes, but

TABLE 4. ADL classification results: balanced accuracy.

Method	WASH dataset CRUFT	WASH dataset ES-MLP	WASH dataset OFS-MLP	ExtraSensory dataset CRUFT	ExtraSensory dataset ES-MLP	ExtraSensory dataset OFS-MLP	MobiAct CRUFT	MobiAct ES-MLP	MobiAct OFS-MLP
ADL-GAN									
Context-transfer	79.46 (3.50)	60.23 (3.44)	**63.70 (3.86)**	62.60 (3.21)	58.07 (3.13)	60.45 (3.22)	**82.33 (3.27)**	**76.12 (2.94)**†	**78.90 (2.53)**
Subject-transfer	**83.21 (3.51)**†	**62.11 (3.92)**†	63.42 (3.45)	**68.04 (2.83)**†	**58.78 (3.87)**	**61.40 (3.29)**	81.34 (3.03)	74.31 (3.06)	78.56 (3.19)
Baseline									
No augmentation	55.36 (3.64)	49.62 (4.10)	53.13 (2.77)	51.14 (2.44)	55.36 (3.64)	55.28 (2.88)	72.75 (2.49)	69.55 (3.24)	72.33 (3.04)
SMOTE	56.45 (2.90)	59.10 (3.31)	58.93 (3.04)	53.69 (2.66)	56.88 (3.34)	57.53 (3.14)	70.18 (3.35)	70.11 (2.97)	73.80 (3.17)
Instance-weighting	63.88 (3.11)	59.28 (3.42)	59.66 (2.77)	55.37 (2.90)	54.56 (2.32)	55.25 (2.27)	78.09 (3.07)	75.44 (3.30)	75.50 (3.22)
Signal transformation	66.15 (3.97)	56.58 (4.33)	56.32 (2.69)	55.52 (3.77)	54.10 (3.16)	56.49 (2.16)	73.96 (2.93)	70.05 (3.06)	73.11 (3.16)
ACGAN	64.13 (3.12)	56.10 (4.10)	58.33 (3.77)	52.09 (3.28)	51.04 (3.18)	52.62 (3.20)	72.77 (2.92)	72.80 (3.08)	74.55 (3.10)
ActivityGAN	63.48 (3.25)	57.30 (4.06)	57.66 (3.99)	-	-	-	72.93 (2.86)	72.55 (3.36)	74.62 (3.18)
CycleGAN	80.84 (3.90)	60.08 (3.37)	58.70 (2.48)	-	-	-	-	-	-
StarGANv2	75.34 (3.60)	58.77 (3.82)	58.30 (3.14)	-	-	-	-	-	-

† indicates a significance level of 0.05 using Wilcoxon's signed-rank test.

TABLE 5. ADL classification results: F1-score.

Method	WASH dataset CRUFT	WASH dataset ES-MLP	WASH dataset OFS-MLP	ExtraSensory dataset CRUFT	ExtraSensory dataset ES-MLP	ExtraSensory dataset OFS-MLP	MobiAct CRUFT	MobiAct ES-MLP	MobiAct OFS-MLP
ADL-GAN									
Context-transfer	71.52 (3.84)	41.39 (3.80)	**44.52 (3.67)**	55.29 (3.78)	47.35 (3.20)	51.53 (3.12)	**74.87 (3.22)**†	**70.53 (3.16)**	**70.90 (3.16)**
Subject-transfer	**75.67 (4.12)**†	42.85 (3.62)	43.18 (3.55)	**64.33 (3.86)**†	**49.73 (3.31)**	**53.62 (3.18)**	72.35 (3.24)	67.99 (3.09)	68.60 (3.13)
Baseline									
No augmentation	40.08 (3.81)	38.62 (3.57)	41.53 (3.35)	38.54 (2.85)	40.08 (3.81)	40.44 (3.25)	66.78 (3.30)	63.35 (3.15)	66.30 (3.04)
SMOTE	44.83 (3.12)	42.58 (3.44)	42.77 (3.39)	40.70 (2.85)	47.97 (3.15)	49.03 (2.97)	65.46 (3.06)	65.63 (2.95)	70.30 (2.99)
Instance-weighting	52.47 (3.32)	41.30 (3.16)	42.56 (3.37)	42.88 (2.77)	48.13 (2.10)	48.68 (2.64)	72.53 (3.14)	71.68 (2.88)	71.80 (2.98)
Signal transformation	56.71 (3.42)	41.56 (3.14)	41.97 (3.30)	44.63 (3.16)	46.45 (3.09)	49.01 (2.95)	68.35 (3.06)	64.35 (2.89)	68.35 (3.04)
ACGAN	54.24 (3.34)	42.65 (4.26)	42.77 (3.88)	50.33 (3.04)	46.12 (3.22)	40.03 (3.19)	68.10 (2.95)	64.80 (2.95)	67.90 (3.07)
ActivityGAN	58.66 (3.58)	42.34 (3.82)	41.68 (3.41)	-	-	-	67.99 (3.07)	64.92 (2.91)	68.26 (3.15)
CycleGAN	73.70 (3.11)	**44.10 (3.57)**	42.89 (3.24)	-	-	-	-	-	-
StarGANv2	70.22 (3.31)	41.67 (3.22)	41.55 (3.14)	-	-	-	-	-	-

† indicates a significance level of 0.05 using Wilcoxon's signed-rank test.

TABLE 6. Quantitative measurement of generated sensor data.

Method	WASH IS	WASH FID	ExtraSensory IS	ExtraSensory FID	MobiAct IS	MobiAct FID	Training Time(Hr)
ADL-GAN							
Context-transfer	3.67 (1.2)	14.76 (3.1)	2.81 (1.0)	28.56 (4.1)	**3.91 (1.1)**	**9.67 (3.2)**	**27**
Subject-transfer	**3.97 (1.4)**	11.10 (3.6)	**3.37 (1.2)**	**12.90 (4.9)**	3.86 (1.2)	9.96 (3.0)	36
ACGAN	3.06 (1.2)	20.76 (4.0)	1.95 (0.8)	45.70 (5.2)	2.24 (1.3)	12.57 (3.1)	32
ActivityGAN	2.56 (0.9)	36.15 (3.8)	-	-	2.26 (1.4)	13.53 (3.6)	284
CycleGAN	3.86 (1.1)	**9.85 (3.8)**	-	-	-	-	468

Higher IS indicates a better diversity of augmented data and Lower FID indicates a better similarity between real and generated data. Training time is reported for WASH dataset.

at the expense of majority class sensitivities — sensitivities of *Sleeping* and *Walking* classes are reduced. Additionally, the results demonstrate that using instance-weighting and SMOTE results in a large gap between sensitivity and specificity for the minority class; sensitivity is greater than specificity in the minority class, indicating that the model is more sensitive to minority classes than other classes. This drawback does not exist in context-transfer and subject-transfer ADL-GANs. Moreover, when SMOTE is employed to augment sensor data, the BAs of the *Talking on phone*, *Stairs down*, *Stairs up*, *Jogging*, and *Exercising* ADL classes are decreased. This decline in performance may be due to the fact that traditional data augmentation techniques, such as linear interpolation, generate samples in a way that does not represent the true data distribution. Furthermore, ADL-GAN and CycleGAN show improvements in the accuracy of intense ADL classes, such as *Exercising*, *Jogging*, *Stairs down* and *Stairs up*, which is significantly better than the baselines, as visualized in Fig. 6. ACGAN and ActivityGAN alleviate the class imbalance problem better than instance-weighting and SMOTE, based on the small gaps between the sensitivity and specificity of the minority class, but the BAs are outperformed by ADL-GANs and CycleGAN.

To investigate the effect of subject embedding on ADL classification accuracy, we analyzed the relation between subject cluster, plotted in embedding space using t-distributed Stochastic Neighbor Embedding (t-SNE) in Fig. 7, and the ADL classification accuracy. An average cosine distance ($1 - \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|}$) was calculated for each subject to determine how well the subject clustered on the embedding space. The result indicates a weak correlation of -0.682 between average cosine distance and ADL classification accuracy. In other words, subject-transfer ADL-GAN performs more effectively on subjects that are well clustered.

TABLE 7. ADL classification performance in each ADL class of WASH dataset.

		No augmentation			Context-transfer			Subject-transfer			ACGAN			CycleGAN		
		BA	Sen	Spec	BA	Sen	Spec	BA	Sen	Spec	BA	Sen	Spec	BA	Sen	Spec
Low-level activity	Exercising†	72.86	60.76	84.96	93.25	91.42	95.08	**93.53**	**92.51**	94.55	84.26	81.62	86.90	93.44	90.53	96.35
	Running†	61.88	58.53	65.23	**91.53**	**93.44**	89.62	90.37	92.31	88.43	81.55	78.95	84.15	90.44	92.74	88.14
	Jogging†	65.75	63.40	68.11	88.72	82.51	94.93	86.44	82.47	90.41	78.89	75.12	82.66	**90.92**	**83.23**	98.61
	Walking	54.96	57.93	51.99	85.10	**87.86**	82.34	**86.51**	85.14	87.88	72.6	82.95	62.25	85.32	86.55	84.09
	Stairs down†	61.04	45.62	76.46	73.62	71.52	75.72	**80.59**	**82.57**	78.61	51.88	51.02	52.74	79.29	72.11	86.47
	Stairs up†	57.98	41.22	74.74	75.53	77.09	73.97	**81.33**	**86.70**	75.96	45.55	48.51	42.59	63.38	75.46	51.30
	Standing	55.74	61.78	49.70	80.15	78.26	82.04	**88.47**	**87.13**	89.81	61.08	69.44	52.72	81.55	77.35	85.75
	Sitting	43.64	47.51	39.77	78.52	76.22	80.82	**85.35**	**83.29**	87.41	56.11	52.15	60.07	81.11	80.06	82.16
	Lying down	41.61	49.33	33.89	66.64	63.13	70.14	**78.73**	**82.85**	74.61	47.09	45.8	48.38	74.78	78.95	70.61
ADL	Sleeping	52.86	71.42	34.30	67.48	70.22	64.75	**73.52**	**70.14**	76.90	63.72	66.71	60.73	68.35	66.20	70.50
	Talking on phone †	63.35	69.03	57.67	81.42	80.54	82.30	**85.62**	**88.11**	83.13	68.04	67.23	68.85	83.47	84.47	82.47
	Typing	46.56	44.02	49.10	76.62	**80.11**	73.13	**81.48**	77.43	85.53	58.52	55.19	61.85	79.12	77.18	81.06
	Bathroom †	41.41	40.77	42.05	74.45	77.33	71.57	69.97	63.41	76.53	64.39	66.53	62.25	**79.77**	**81.59**	77.95

		Instance weighting			SMOTE			Signal transformation			ActivityGAN		
		BA	Sen	Spec	BA	Sen	Spec	BA	Sen	Spec	BA	Sen	Spec
Low-level activity	Exercising†	74.60	90.49	58.72	64.31	52.57	76.05	76.08	66.16	86.00	71.29	66.12	76.46
	Running†	72.32	89.75	54.90	61.85	56.15	67.55	77.90	78.83	76.97	72.80	69.44	76.16
	Jogging†	71.75	78.93	64.57	61.18	54.42	67.94	78.61	74.12	83.10	78.32	82.93	73.71
	Walking	68.70	55.00	82.40	68.20	81.58	54.82	67.22	75.22	59.22	73.15	75.16	71.14
	Stairs down†	49.96	63.47	36.45	44.65	47.13	42.17	57.39	60.14	54.64	57.61	55.37	59.85
	Stairs up†	48.64	79.31	17.94	42.48	45.14	39.82	52.46	52.60	52.32	52.53	54.20	50.86
	Standing	66.97	58.21	75.73	71.04	55.64	86.44	68.24	71.25	65.23	67.84	65.88	69.80
	Sitting	61.11	50.91	71.30	58.69	77.42	39.96	67.18	70.56	63.80	58.10	62.97	53.23
	Lying down	54.09	55.57	52.64	53.3	60.25	46.35	57.85	61.44	54.26	62.46	64.51	60.41
ADL	Sleeping	64.50	62.76	66.24	65.11	55.24	74.98	62.05	55.61	68.49	68.77	70.92	66.62
	Talking on phone †	69.43	93.43	45.44	44.56	20.41	68.71	62.38	60.79	63.97	51.02	53.32	48.72
	Typing	61.56	54.96	68.17	53.02	31.55	74.49	62.91	65.15	60.67	62.98	64.27	61.69
	Bathroom †	66.79	71.42	62.16	45.44	42.51	48.37	69.74	66.08	73.40	48.41	50.56	46.26

The best performance of data augmentation method in each ADL class is in bold. We highlight sensitivity (Sen) and specificity (Spec) that have its difference higher than 25% of BA. † indicates minority class in WASH dataset.

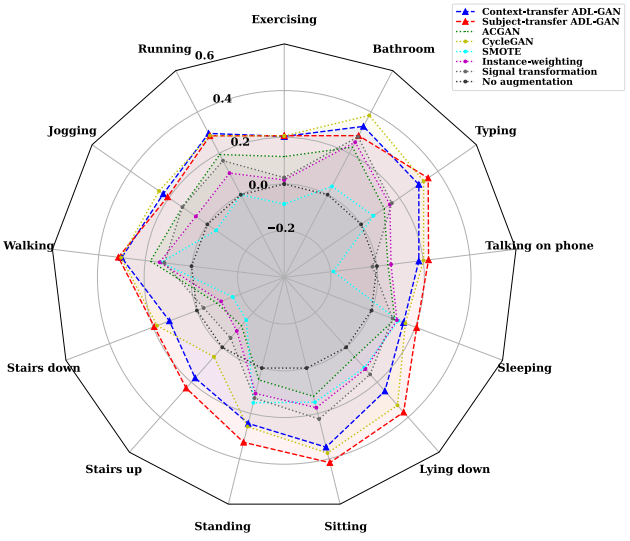


FIGURE 6. Improvement of ADL classification performance for each class in WASH dataset.

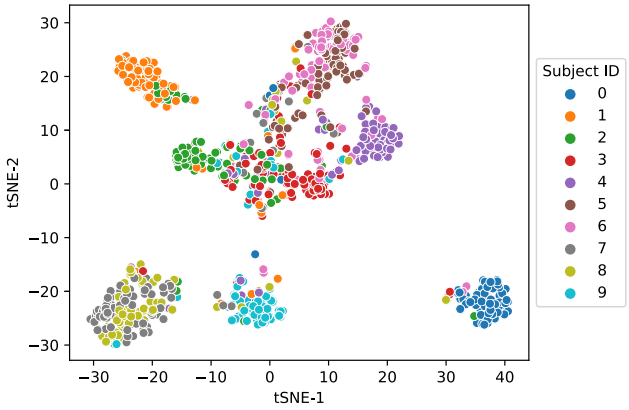


FIGURE 7. t-SNE visualizations of subject embedding.

B. CLASS ENTROPY

In order to analyze the degradation of ADL classification performance caused by imbalanced classes, we varied the class entropy of WASH and MobiAct datasets, as shown in Table 8. The proposed ADL-GAN was compared against SMOTE, ACGAN and signal transformation baselines. Augmenting data with subject-transfer ADL-GAN improves the BA the most in very skewed data distributions (entropy$<2^{1.4}$) whereas context-transfer ADL-GAN provides the best performance when the class distribution is more balanced (entropy$>2^{1.5}$). These findings imply that augmenting ADL with imbalanced class distributions from the same ADL class

generates a better sample than augmenting with a different class and from a random vector (ACGAN). However, the main limitation of ADL-GAN and other deep learning-based data augmentation methods is that it requires a vast amount of training samples. For the MobiAct dataset, signal transformation and SMOTE all do not manifest a trend of decreasing BA while other methods do.

C. SENSOR-WISE AUGMENTATION

To measure the importance of each sensor, we retrained subject-transfer ADL-GAN to augment all combinations of these three sensors. The results are shown in Table 8. When comparing each sensor individually, the accelerometer is the most important sensor, attaining a BA of 48.25 using instance-weighting, which can be enhanced by 34% using subject-transfer ADL-GAN. The CRUFT model performs optimally on combinations of Acc+MFCC and Acc+Gyro+MFCC, with slightly superior performance on Acc+Gyro+MFCC. ADL-GAN also outperforms signal transformation and ActivityGAN on all sensor combinations. In the signal transformation method, an improvement achieved from introducing MFCC features is limited compared to other data augmentation methods, indicating that SpecAugment may not be suitable for augmenting MFCCs for ADL recognition tasks. For ActivityGAN, the data augmentation method does not surpass the instance-weighting baseline in all sensor combinations. The best performance of ActivityGAN is observed in Acc+MFCC and Acc+Gyro+MFCC with the lowest BA observed in Gyro.

VII. DISCUSSION

A. TRANSFER GANs CREATE MORE REALISTIC AND DIVERSE DATA THAN OTHER GANs

Our results show that synthesizing a sample by transforming from real data, which is visualized in Fig. 9, yields more realistic samples than from a random vector with the condition. This aligns with results in StarGAN and CycleGAN, which both generated more realistic images than the conditional GAN. While the conditional GAN is able to generate a sample without using a real sample, the samples generated are often less realistic than those generated using transfer GANs such as ADL-GAN and CycleGAN. Table 7 and 8 also demonstrate that transfer GANs are able to improve ADL recognitions in an imbalanced dataset in most ADL class and all sensors, which outperforms conditional GAN (ACGAN) and vanilla GAN (ActivityGAN).

B. DATA AUGMENTATION OUTPERFORMS COST-SENSITIVE LEARNING IN MITIGATING CLASS IMBALANCE

Oversampling minority classes with data augmentation improves overall classification BA in the CRUFT model. This is seen in both ADL-GAN and baseline signal transformation methods. ADL-GAN outperforms the instance-weighting baseline by 31%. While cost-sensitive learning incorporates the minority class into the weight gradient calculation, the minority class may not be well-sampled, which leads to overfitting of some ADL classes as has previously been reported in [12].

C. DEEP LEARNING-BASED ADL CLASSIFICATION PERFORMED BEST USING DATA AUGMENTATIONS GENERATED USING ADL-GAN

It was demonstrated that a deep learning model that classifies low-level features outperforms one that classifies handcrafted features. Although the handcrafted features were extracted from ADL-GAN-augmented features, the performance of the MLP model does improve significantly, as compared to the CRUFT model. This may due to the limitation of handcrafted features or the fact that CRUFT model contains more parameters.

D. MFCC IS THE MOST IMPORTANT FEATURE IN FOR THE ES-MLP AND OFS-MLP ADL RECOGNITION MODELS

Feature importance was analyzed using GradientSHAP [45], which estimates SHAP values by adding Gaussian noise into each training sample as baseline and randomly selecting points between baseline and test samples to compute the gradient. The contribution of each input feature to the output of MLP model is approximated by gradients * (inputs - baselines).

The mean value of the first MFCC component is indicated as features that have high impact on the MLP model based on SHAP values plotted in Figure 10. Although MFCC was initially developed for the speech task, e.g. the first MFCC captures most information in human speech (e.g., formants and spectral envelope). MFCC is not robust to noise, which benefits ADLs that have trivial speech content. These findings align with those of Vaizman et al. [3] that reported that audio was a significant sensor in analyses and modeling performed as part of Extrasensory HCR project when location and phone state features were not considered. The ADL class predictions that are affected by MFCC audio features the most are sitting, lying down, running, sleeping, bathroom and talking on the phone.

Although MFCCs are demonstrated to be important for ADL recognition, there are concerns that collected audio features or MFCCs may violate smartphone owner's privacy. In traditional speech processing, MFCCs are frequently adopted as features in the speaker recognition and ASR tasks. In a data leak incident, MFCCs may reveal the identity of the speaker and speech contents to a certain degree. However, ASR requires more MFCCs to perform well. This study considers only MFCCs up to 13 components, which is not sufficient for ASR that needs 21-42 MFCCs for speech to be intelligible [46].

E. POSSIBLE APPLICATIONS OF ADL-GAN

This study proposes ADL-GAN to address the class imbalance problem that is common in in-the-wild datasets.

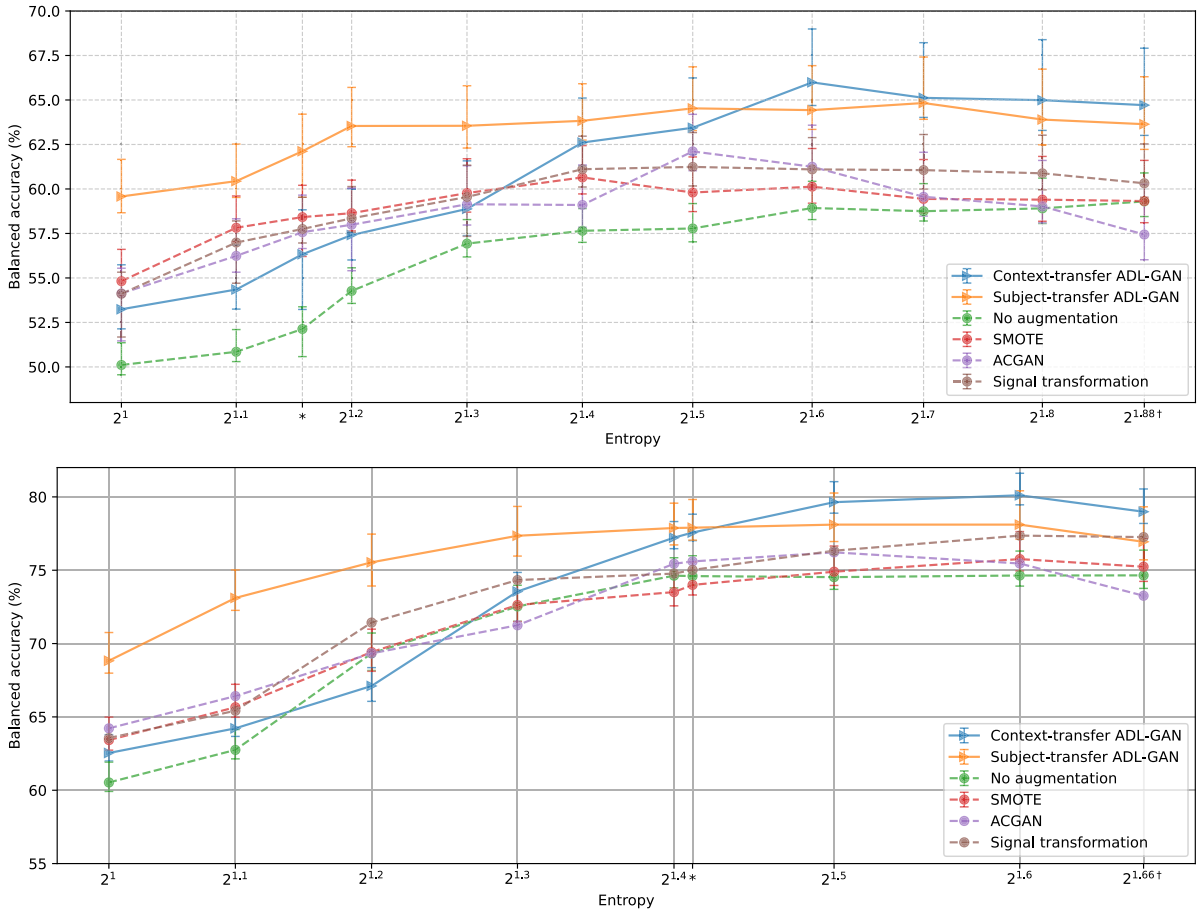


FIGURE 8. ADL-GAN data augmentation method in various class imbalanced level, Top: WASH dataset, Bottom: MobiAct dataset, Annotation on entropy value: * indicates original entropy of the dataset and † indicates upper bound of the entropy.

TABLE 8. ADL classification result using the CRUFT model on the WASH dataset.

	Subject-transfer ADL-GAN		Instance-weighting		Signal transformation		ActivityGAN	
Sensors	BA	Weighted F1	BA	Weighted F1	BA	Weighted F1	BA	Weighted F1
Acc	64.76 (3.14)	56.67 (3.62)	48.25 (3.05)	48.90 (2.68)	46.94 (2.87)	47.61 (2.93)	49.11 (3.04)	49.22 (3.15)
Gyro	55.18 (3.07)	51.12 (2.99)	44.77 (3.12)	47.02 (3.05)	45.11 (3.14)	46.98 (2.87)	40.03 (2.66)	42.74 (2.81)
MFCC	48.52 (2.81)	51.26 (3.01)	43.10 (2.85)	46.86 (3.12)	41.64 (3.43)	46.19 (3.00)	40.72 (2.84)	41.59 (2.40)
Acc+Gyro	72.69 (3.28)	68.50 (3.08)	53.22 (3.45)	49.61 (3.17)	**64.62 (3.07)**	**59.46 (2.95)**	50.27 (3.31)	49.64 (2.97)
Acc+MFCC	**81.96 (3.18)**	**74.35 (3.90)**	**63.41 (3.22)**	51.17 (3.47)	48.50 (3.26)	46.43 (3.05)	**64.92 (3.16)**	**63.90 (2.98)**
Gyro+MFCC	76.09 (3.37)	68.40 (3.88)	55.10 (2.89)	50.78 (3.00)	49.52 (2.92)	52.20 (2.64)	52.50 (3.04)	49.66 (3.00)
All three	**83.21 (3.51)**	**75.67 (4.12)**	**63.88 (3.11)**	52.47 (3.32)	**66.15 (3.97)**	56.71 (3.42)	**63.48 (3.25)**	58.66 (3.58)

ADL-GAN has the potential to be applied to other applications in ADL and HAR, such as subject/device adaptation, data scarcity, and soft-labeling of unlabeled samples. For instance, subject-transfer ADL-GAN can used for subject adaptation and soft-labeling by augmenting testing subjects data from the embedding vector extracted from a short (label or unlabeled) snippet of data. For data scarcity, Figure 8 visualizes the performance of ADL-GANs in two extreme scenarios. At the lowest entropy value of 2, the amount of training data in minority classes are significantly reduced, demonstrating how the models perform on a dataset that contains ADL classes that rarely occur. At the highest entropy value, total training samples are greatly reduced to obtain a training set with equal numbers of samples in all ADL classes, which demonstates how the models perform in a limited data scenarios without class imbalance. Our ADL-GANs

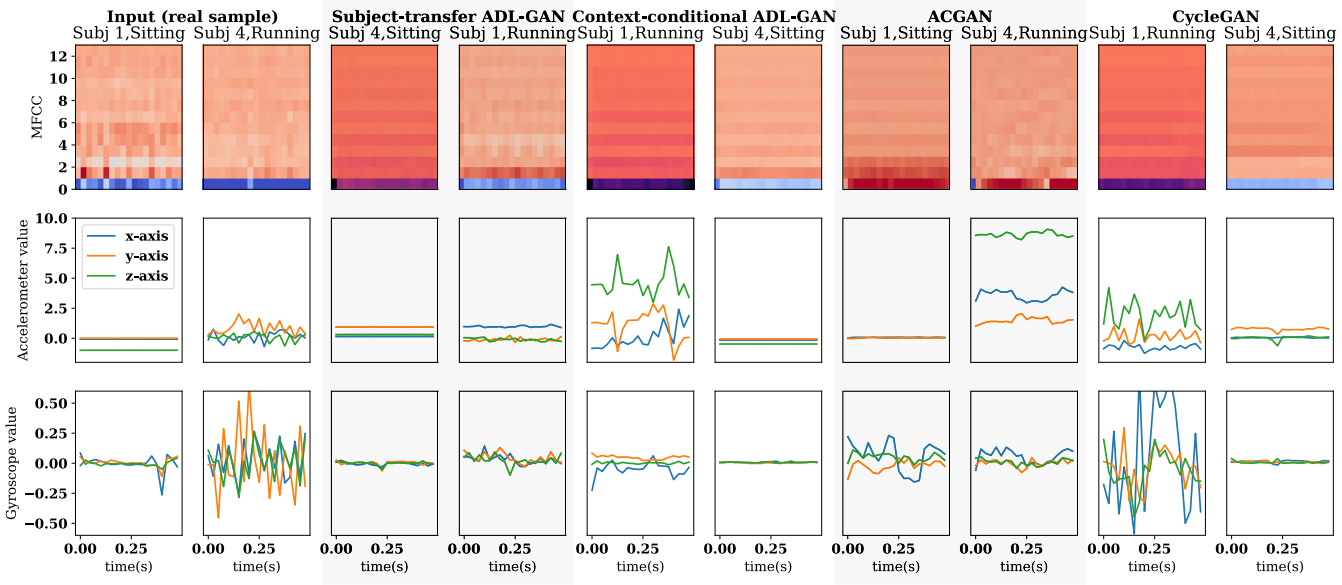


FIGURE 9. Augmented MFCC, accelerometer and gyroscope features using context-transfer ADL-GAN, subject-transfer ADL-GAN, ACGAN and CycleGAN in WASH dataset.

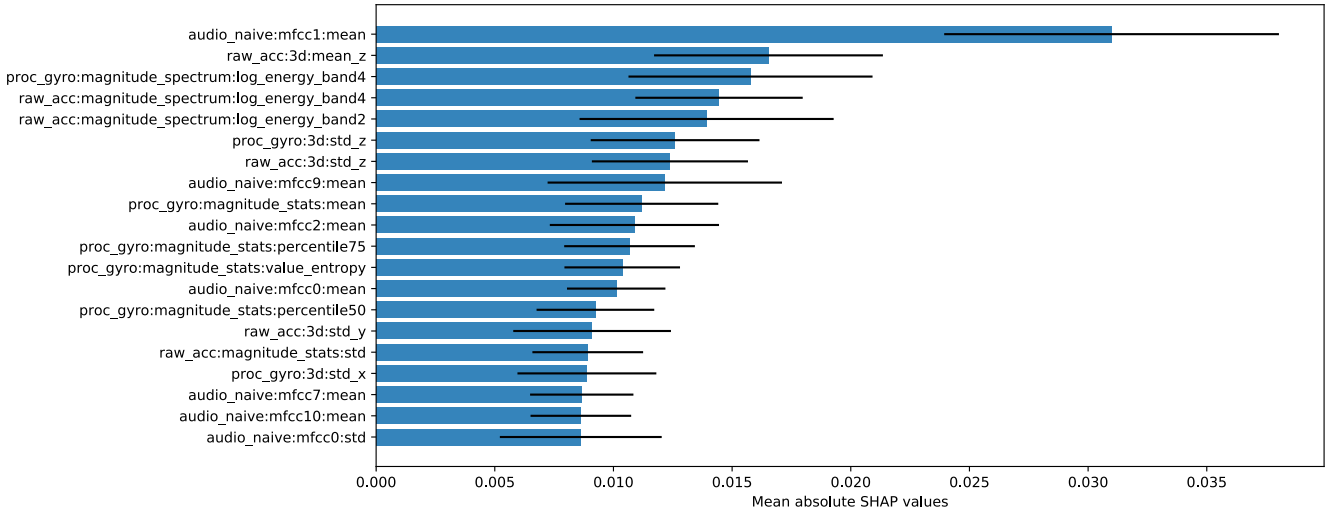


FIGURE 10. Feature importance: Top 10 features of ES-MLP in WASH dataset.

were demonstrated to outperform the baselines in both scenarios.

F. LIMITATIONS OF ADL-GAN

ADL-GAN was designed for low-level sensor features, a subset of features presented in the datasets. Some soft features or high-level ones that accumulate through time, such as "time of day" and "walking step", were omitted from this investigation due to overfitting during GAN training. Another disadvantage of this method is due to the nature of GANs, which requires more training time and a significant amount of training samples.

G. FUTURE WORK

First of all, we would like to evaluate ADL-GAN on more ADL classes. Secondly, both ADL-GANs can also be used to augment data to solve the limited-data problem. Subject embedding can work effectively on a small set of samples, and samples can be generated using the proposed subject-transfer ADL-GAN. Another possibility is to employ context-transfer or subject-transfer ADL-GAN to adapt subjects in a meta-learning and meta-testing fashion. In the experiments involving sub-sampling, some models overfitted to the training set where we tackled the problem by reducing the model size. Alternatively, transfer learning or pre-training

model may be considered as they can reduce the overfitting issue.

VIII. CONCLUSION

Class imbalance is a common classification problem in in-the-wild datasets, including those collected for ADL. This paper proposed ADL-GAN, which augments data in two different ways: 1) context-transfer synthesizes features in a minority ADL class for a given subject using features from the majority class, and 2) Subject-transfer synthesizes a minority ADL class of a subject using data collected for that ADL from other subjects. We show that subject-transfer ADL-GAN, which utilizes a subject embedding with contrastive loss, generates more distinct training samples and outperforms conventional data augmentation methods, including SMOTE, instance-weighting and a conditional GAN in the ADL datasets with skewed class distributions. It improved in-the-wild ADL classification the most, up to 27.9 on balanced accuracy and 35.6 on F1-score. Context-transfer ADL-GAN improved scripted ADL classification the most, up to 9.58 on balanced accuracy and 8.09 on F1-score. Our findings demonstrate that ADL-GANs are a viable method for augmenting data of minority classes to address the problem of imbalanced classes.

ACKNOWLEDGMENT

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not with standing any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

REFERENCES

[1] B. Guidet, D. W. De Lange, A. Boumendil, S. Leaver, X. Watson, C. Boulanger, W. Szczeklik, A. Artigas, A. Morandi, and F. Andersen, "The contribution of frailty, cognition, activity of daily life and comorbidities on outcome in acutely admitted patients over 80 years in European ICUs: The VIP2 study," *Intensive Care Med.*, vol. 46, no. 1, pp. 57–69, Jan. 2020.

[2] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The MobiAct dataset: Recognition of activities of daily living using smartphones," in *Proc. Int. Conf. Inf. Commun. Technol. Ageing Well E-Health*, 2016, pp. 143–151.

[3] Y. Vaizman, N. Weibel, and G. Lanckriet, "Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–22, Jan. 2018.

[4] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–28, 2017.

[5] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Comput.*, vol. 16, no. 4, pp. 62–74, Oct./Dec. 2017.

[6] W. Ge and E. Agu, "CRUFT: Context recognition under uncertainty using fusion and temporal learning," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2020, pp. 747–752.

[7] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2008.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 28, pp. 321–357, Jun. 2006.

[9] J. Wang, Y. Chen, Y. Gu, Y. Xiao, and H. Pan, "SensoryGANs: An effective generative adversarial framework for sensor-based human activity recognition," in *Proc. IJCNN*, 2018, pp. 1–8.

[10] W.-H. Chen and P.-C. Cho, "A GAN-based data augmentation approach for sensor-based human activity recognition," *Int. J. Comput. Commun. Eng.*, vol. 10, no. 4, pp. 75–84, 2021.

[11] X. Li, J. Luo, and R. Younes, "ActivityGAN: Generative adversarial networks for data augmentation in sensor-based human activity recognition," in *Proc. ACM Ubicomp WearSys*, 2020, pp. 249–254.

[12] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, Jul. 2020.

[13] V. Sampath, I. Maurtua, J. J. A. Martín, and A. Gutierrez, "A survey on generative adversarial networks for imbalance problems in computer vision tasks," *J. Big Data*, vol. 8, no. 1, pp. 1–59, Dec. 2021.

[14] K. T. Nguyen, F. Portet, and C. Garbay, "Dealing with imbalanced data sets for human activity recognition using mobile phone sensors," in *Proc. Int. Workshop Smart Sens. Syst.*, 2018, pp. 1–11.

[15] M. Esmaeilpour, P. Cardinal, and A. L. Koerich, "Unsupervised feature learning for environmental sound classification using weighted cycle-consistent generative adversarial network," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105912.

[16] S. Shahnawazuddin, N. Adiga, K. Kumar, A. Poddar, and W. Ahmad, "Voice conversion based data augmentation to improve children's speech recognition in limited data scenario," in *Proc. Interspeech*, Oct. 2020, pp. 4382–4386.

[17] Q. Liu, G. Ma, and C. Cheng, "Data fusion generative adversarial network for multi-class imbalanced fault diagnosis of rotating machinery," *IEEE Access*, vol. 8, pp. 70111–70124, 2020.

[18] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE CVPR*, Jun. 2018, pp. 8789–8797.

[19] B. D. Setiawan, U. I. Serdult, and V. Kryssanov, "Smartphone sensor data augmentation for automatic road surface assessment using a small training dataset," in *Proc. IEEE Int. Conf. Big Data Smart Comput.*, Jun. 2021, pp. 239–245.

[20] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," 2016, *arXiv:1611.02163*.

[21] C. E. Roberts, L. H. Phillips, C. L. Cooper, S. Gray, and J. L. Allan, "Effect of different types of physical activity on activities of daily living in older adults: Systematic review and meta-analysis," *J. Aging Phys. Activity*, vol. 25, no. 4, pp. 653–670, Oct. 2017.

[22] C. G. Blankevoort, M. J. G. van Heuvelen, F. Boersma, H. Luning, J. de Jong, and E. J. A. Scherder, "Review of effects of physical activity on strength, balance, mobility and ADL performance in elderly subjects with dementia," *Dementia Geriatric Cognit. Disorders*, vol. 30, no. 5, pp. 392–402, 2010.

[23] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. NIPS*, vol. 27, 2014, pp. 2672–2680.

[25] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE ICCV*, Jun. 2017, pp. 2223–2232.

[27] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[28] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

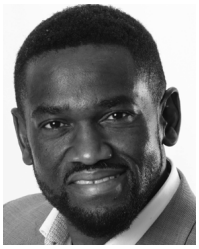
[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," 2019, *arXiv:1904.08779*.

[30] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulic, "Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks," in *Proc. 19th ACM Int. Conf. Multimodal Interact.*, Nov. 2017, pp. 216–220.

[31] T. Kaneko and H. Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," 2017, *arXiv:1711.11293*.
[32] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, *arXiv:1603.07285*.
[33] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and D-vectors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5274–5278.
[34] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2016, pp. 165–170.
[35] N. Kanda, S. Horiguchi, Y. Fujita, Y. Xue, K. Nagamatsu, and S. Watanabe, "Simultaneous speech recognition and speaker diarization for monaural dialogue recordings with target-speaker acoustic models," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 31–38.
[36] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4052–4056.
[37] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 266–273.
[38] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4879–4883.
[39] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN v2: Diverse image synthesis for multiple domains," in *Proc. IEEE CVPR*, Mar. 2020, pp. 8188–8197.
[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NIPS*, 2019, pp. 8024–8035.
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[42] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 2234–2242.
[43] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. NIPS*, vol. 30, 2017, pp. 1–12.
[44] M. A. U. H. Tahir, S. Asghar, A. Manzoor, and M. A. Noor, "A classification model for class imbalance dataset using genetic programming," *IEEE Access*, vol. 7, pp. 71013–71037, 2019.
[45] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 4765–4774.
[46] J. Psutka, L. Muller, and J. V. Psutka, "Comparison of MFCC and PLP parameterizations in the speaker independent continuous speech recognition task," in *Proc. 7th Eur. Conf. Speech Commun. Technol.*, 2001, pp. 1–4.

APIWAT DITTHAPRON received the B.S. degree in computer science from the Worcester Polytechnic Institute (WPI), Worcester, USA, in 2020, where he is currently pursuing the Ph.D. degree in computer science. His research interests include machine learning and deep learning for mobile health.

ADAM C. LAMMERT received the Ph.D. degree in computer science from the University of Southern California, Los Angeles, USA, in 2014. He is an Assistant Professor of biomedical engineering and a Core Faculty Member with the Neuroscience Program, Worcester Polytechnic Institute, Worcester, MA, USA. His research interests include the applications of signal processing and computational modeling to assessment of neurological health, with a special interest in the area of speech, language, and hearing.



EMMANUEL O. AGU received the Ph.D. degree in electrical and computer engineering from the University of Massachusetts Amherst, Amherst, MA, USA, in 2001. He is the Harold L. Jurist '61 and the Heather E. Jurist Dean's Professor with the Computer Science Department, Worcester Polytechnic Institute, Worcester, MA. He has been involved in research in the mobile and ubiquitous computing. He is currently working on mobile health projects to assist patients with diabetes, obesity, depression, wounds, TBI, and infectious diseases.