HAR-CTGAN:

Synthesizing Continuous and Discrete Mobile Sensor Data Using GAN models for Human Activity Recognition

A Major Qualifying Project

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science in

COMPUTER SCIENCE

and

DATA SCIENCE

by:

Joshua DeOliveira

August, 2022

Elke Rundensteiner, Ph.D.

Founding Director in Data Science

Abstract

Human activity recognition (HAR) is the process of determining physical activities performed by individuals using mobile sensor data. HAR is the backbone of many mobile healthcare applications, such as passive health monitoring systems, early diagnosing systems, and fall detection systems. Effective HAR models rely on deep learning technologies in order to accurately classify what activity was being performed in a data instance. In turn, HAR models require large collections of labeled real-world human activity data. Unfortunately, HAR datasets are expensive to collect, are often mislabeled, and have large class imbalances. State-of-the-art approaches to address these challenges utilize Generative Adversarial Networks (GANs) for generating additional synthetic data along with their labels. Problematically, these HAR GANs only synthesize continuous features — features that are represented with real numbers — recorded from gyroscopes, accelerometers, GPS systems, and other sensors that produce continuous data. This is limiting since mobile sensor data commonly has discrete features that provide additional context, such as Bluetooth state, sensor location (prioception), and time-of-day. It has been shown that the availability of these discrete features can substantially improve HAR classification. Within the healthcare domain, misclassifications can have damaging or even fatal impacts on the individuals that rely on these models. Hence, we studied Conditional Tabular Generative Adversarial Networks (CTGANs) for data generation to synthesize mobile sensor data containing both continuous and discrete features, a task never been done by state-of-the-art approaches. We show HAR-CTGANs generate data with greater realism resulting in allowing better downstream performance in HAR models. Synthesized data from HAR-CTGAN when used in HAR model training resulted in a 63% greater improvement in F1 performance than using synthesized data from state-of-the-art. When state-of-the-art models were modified with HAR-CTGAN characteristics downstream F1 performed increased by 18%.

Acknowledgements. I would like to thank my advisor Elke Rundensteiner, PhD and graduate mentor Walter Gerych for their help and guidance throughout this project. I would also like to thank Kyle Costello, Alek Lewis, and Ryan Astor for their work as collaborators on the MQP project *Building a Customizable GAN Package Tool in Python* in order to better prepare the work done in this MQP. Finally, I thank Nichole Ramirez for their continual support and encouragement throughout the project.

The results in this paper were obtained in part using a high-performance computing system acquired through NSF MRI grant DMS-1337943 to WPI.

1 Introduction

Background. Human Activity Recognition (HAR) is the process of classifying the physical activities performed by individuals using sensor data. These physical activities are typically day-to-day low-level tasks, such as walking, standing, and sitting [1–4]. The sensors used for capturing data come from a variety of sensor components in mobile devices, namely gyroscopes, accelerometers, GPS trackers, altimeters, and microphones. Advancements in high-precision sensors along with the heightened ubiquity of wearable technology [5], such as smartphones and smartwatches, have led to rapidly greater access to high volumes of accurate sensor data. HAR classifiers have been applied to a wide range of domains, such as security [6] and urban development [7]. In particular, HAR classifiers are the backbone of many mobile healthcare applications [8]. HAR methods are important for mobile healthcare as they are able to detect a variety of illnesses, such as depression [9], Parkinson's disease [10], autism [11], and Covid-19 [12, 13]. Additionally, HAR classifiers have been used for assisted living systems [14–17] as they can detect adverse events, such as falling [17, 18] and early stroke diagnosis [19].

In the real world, there is a huge diversity in people's behavior and the manner in which they perform activities [4]. To create HAR models that perform well on real-world data, state-of-the-art approaches train models on *in-the-wild* datasets [20]. In-the-wild datasets are collected passively as study participants go about their daily lives [5]. However, HAR models often require labels for each activity that is being performed at any given instance [21]. For this reason, study participants are asked to provide annotations for the activities they perform throughout their day [20]. Unfortunately, due to the imperfect nature of human annotators and the huge time cost required to annotate every activity at every minute of the day, HAR data sets are often riddled with incorrect or missing labels [21]. Additionally, in-the-wild datasets often have egregious class imbalances with some activities rarely performed by specific individuals [22]. This phenomenon can be largely attributed to subjects simply choosing not to perform certain activities, imprecisely labeling their activities retroactively,

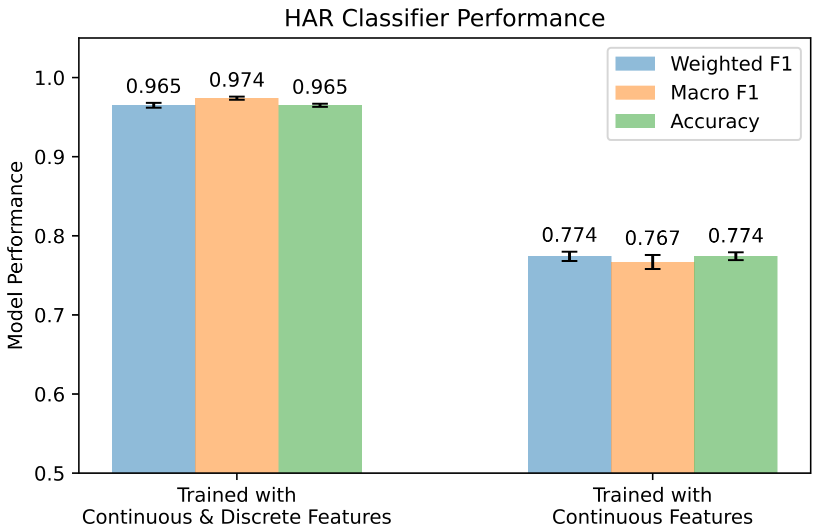


Figure 1: Plot of the model performances of 20 HAR classifiers trained to distinguish between 7 unique activities found in *ExtraSensory* [20]. If the discrete features were meaningless or added ambiguous context to the sensor data, the models trained with these features would add no effect or confuse the models, causing equivalent or worse performance than models trained only with continuous features. However, this figure demonstrates that classifiers that used discrete features yielded higher performance than classifiers that only accepted continuous features. 10 classifiers were trained using exclusively continuous features, whereas the other 10 classifiers were trained with the same continuous features along with also discrete features.

or completely forgetting to label. For example biking, swimming, walking, and sleeping are common HAR activities researchers record mobile sensor data. In an average week, most people will 6-8 hours a day sleeping and do occasional walking, and exercise only a few days a week for only an hour or two on those days. In these cases, exercising activities like swimming and biking are rarely ever seen.

Furthermore, in the healthcare field specifically, HAR has been used to develop early warning systems or detect falling or tripping, which are useful for the elderly and individuals with disabilities [14–19] However, the data to train these systems are typically collected from young, able-bodied people due to the high risks of having physically vulnerable individuals repeatedly fall – just so that sample data could be collected. This means the data collected doesn't accurately represent the demographic using these downstream models. Misclassifications due to user-specific movement patterns can have serious life-or-death implications for individuals that rely on these passive monitoring systems to automatically contact first responders.

Approaches to remedy these problems can range from dedicated staff that validate these labels with additional monitoring systems, all the way to collecting additional data from individuals that will hopefully perform a diverse set of activities [20]. Intuitive solutions such as having subjects perform specifically desired activities are not fruitful, as recent studies show that HAR datasets collected in controlled lab environments do not effectively mimic activities from in-the-wild studies [21]. Thus, these existing simple solutions to address these data issues can be expensive, laborious, or infeasible.

With the multiple issues laid out above plaguing downstream HAR models, there is a high demand for reliable techniques to consistently have clean, realistic, high quality mobile sensor datasets without suffering the potential down-sides of disregarding financial limitations, or time-constraints.

State-of-the-Art.

State-of-the-art solutions employ Generative Adversarial Networks (GANs) to upsample real datasets with synthesized realistic-looking data that mimics users performing activities that are commonly recorded in HAR studies [2, 23–25]. GANs are a machine learning framework in which two deep networks, a generator and a discriminator, train against each other in an adversarial environment in order to synthesize data that is indistinguishable from a given dataset. The generator is tasked with learning to map random noise into fake data that matches patterns in real data, while a discriminator, also known as a judge or critic, tasked with deciphering which data is real and fake when blindly given mixed batches of sample from an original dataset and from the generator.

Under ideal training conditions, the generator learns to effectively synthesize data that the discriminator incorrectly recognizes as real data. It is important to note that the generator doesn't learn to synthesize data that are blatant copies of the real data, as it would

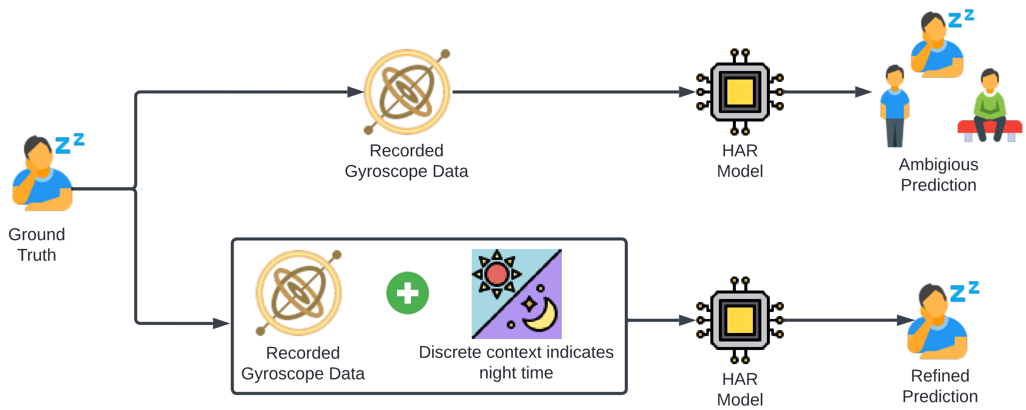


Figure 2: When HAR models only use continuous sensor features due to the upsampling limitations of GANs, it can be difficult to distinguish similar activities. But, when contextual discrete features are also leveraged, refined classifications can be made from previously ambiguous data. In this example, sensor data when sleeping can look similar to sensor data when sitting or standing, leading to easily classifying the ground truth incorrectly. But, when sensor data when sleeping is coupled with a discrete feature indicating its night time, then a HAR model can classify the data as sleeping with high confidence.

not yield a meaningful impact for downstream models. GANs are particularly powerful in their capability of near limitlessly sampling of new data for constructing arbitrarily large, enriched datasets.

While many GAN approaches proposed for HAR have mended data specific issues [2, 23–25], these HAR-GANs are only tasked with generating the *continuous* features of mobile sensor data. This is a major limitation, as HAR data sets commonly have a significant volume of discrete features that provide contextual details to the continuous features and improve HAR classification, as seen in figure 1. State-of-the-art GANs for HAR don't incorporate discrete features in their generation as GANs originally were designed to synthesize images and other continuous forms of data [26], and historically fail at synthesizing discrete data.

Problem Definition. In this work, we address the problem of generating realistic HAR sensor data. In particular, we aim to generate both continuous features as well as realistic *discrete* features. This to-date is an open unstudied problem in the HAR domain.

A successful generative model will produce discrete sensor data that is not only realistic in isolation, but is also realistic when paired with the continuous features that are being simultaneously generated. For instance, if a discrete feature indicates that the mobile sensor is at rest on a table, then the corresponding generated continuous accelerometer data should not indicate significant movement.

Challenges. First, the application of GANs on multimodal datasets is a challenging task, as GANs are notorious for their unstable training process and sensitivity to mode collapses [27]. Second, modeling discrete features is difficult as this involves making a discrete choice, for which backpropagating through is not straightforward [28]. Additionally, there is a large diversity in HAR datasets, as no two individuals perform the same activity in the same way [2]. The context in which an activity is performed also affects the corresponding sensor data. For instance, walking over a hardwood floor will yield a different sensor stream than walking over uneven terrain while hiking.

Proposed Solution. To the best of our knowledge, there is no current technique in the literature that has been proposed for upscaling mobile sensor datasets with discrete or nominal feature data. We believe this is the first approach to be applied for HAR applications. Thus, we propose the explore if the new state-of-the-art conditional tabular generative adversarial network (CTGAN) could be adopted and then adapted to be utilized for unsupervised data generation in the HAR domain. We refer to the resulting model henceforth as HAR-CTGAN. We hypothesize that the model when applied to HAR data will succeed to synthesize realistic mobile sensor data containing both continuous and discrete features.

Contributions. Our contributions include:

- Applying Conditional Tabular GANs to the HAR domain to generate mobile sensor data with both continuous and discrete data features.
- Demonstrating a need for generating discrete features for use in downstream HAR models.

- Synthesizing data from HAR-CTGAN model outperforms the generation quality of the state-of-the-art GANs on a real HAR dataset.
- Demonstrating when state-of-the-art GANs are modified to have attributes from HAR-CTGAN the state-of-the-art model's generation qualities improve.

2 Related Work

2.1 Non-GAN Solutions for HAR Class Imbalances

Before the proliferation of neural-network-based generative modeling, novel data generation was achieved through a variety of other machine learning techniques. k-Nearest Neighbor interpolation techniques such as SMOTE: Synthetic Minority Oversampling Technique [29]. SMOTE derivatives such as BLL-SMOTE [22] have shown to generate more realistic synthesized data across non-convex feature spaces, and SMOTE-SVM [30] uses a SMOTE-like approach directly when training the HAR classifier. Other proposed ways to deal with class imbalanced data directly have used Weighted SVMs [31], Cost-Sensitive SVMs [32], Random Forest classifiers [33], and dual-ensembles [34].

The authors of these papers commonly ignore and drop any of the nominal and ordinal features in the datasets their methods are applied to, as their methods typically cannot handle these types of data well.

2.2 GAN-Based Data Generation

Due to HAR becoming a growing area of research over the past decade [1, 3, 5, 14], there is a greater demand for better techniques for handling HAR class imbalances. For multiple years now, GAN frameworks have been implemented as a method of generating realistic data suitable for upsampling real image datasets [26, 28, 35, 36]. More recently, they have been seamlessly and successfully applied to the HAR domain. Due to the unique highly-dimensional and tabular nature of mobile sensor data, developments in generating activity-specific data to ensure the generated data can follow patterns in the minority HAR classes [23–25]. GANs can be further tailored to handle modeling the different styles in which multiple users can perform the same activity [2]. However, all of the techniques mentioned fail to recognize the utility of generating the discrete features in sensor data.

3 Problem Definition

Suppose we are given a dataset $\mathbb{X} = \{(x_c, x_d, y)_i\}_{i=1}^{N}$, where $x_c \in C$ corresponds to the continuous features (e.g. accelerometer features), $x_d \in D$ the discrete features (prioception, device locked/unlocked, bluetooth on/off, etc.), $y \in \mathbb{Y}$ is the activity being performed for that instance, and N is the number of instances in the dataset.

Our task is to obtain a generative model $\mathcal{G}$, such that $\mathcal{G}(y) \sim P_R((C, D)|Y = y)$; in order for our generative model to follow the distribution of continuous and discrete features according to the specific human activity the mobile sensor data is meant to be representative of. As indicated in Tanielian et al. [37], the following proposition holds, the generator architecture that is utilized in start-of-the-art HAR GAN models is unable to learn to perfectly generate discrete features from latent noise and requires additional complexity to approximate a discrete image. The table of notation used in this work is given in Table 1.

Proposition 1. Let $\mathcal{G}$ be a multi-layer perceptron whose domain Z is a latent space in $\mathbb{R}^n$ that is sampled to get random noise z according to a Gaussian distribution, such that $\mathcal{G}$ yields an image $\mathbb{H}$ lying in $\mathbb{R}^m$. If so, then $\mathcal{G}(z)$ cannot yield a discrete image.

Proof.

1. Let ϕ be an isomorphism that relates every multi-layer perceptron to a vector-valued function f and vice-versa. Since a multi-layer perceptron consists of a finite sequence of layers in which there is a linear transformation W and then a non-linear transformation V that is lipschitz continuous, then f can be constructed as $f(\vec{v})=$

$\vec{v}W_1V_1W_2V_2 \dots W_kV_k$ where k is the number of layers in the multi-layer perceptron. Since f is a product of transformations that all of which are continuous, then we can assert f is continuous and that every multi-layer perceptron can be thought of as a continuous vector-valued function.

2. Since Z lies in $\mathbb{R}^n$ it is a connected space. Therefore, since f is a continuous function acting on a connected space, the image of f must be connected.

3. Let $\mathbb{H}$ be the finite set of points desired for $\mathcal{G}$ to yield as its image. As $\mathcal{G}$ can only yield a connected space when randomly sampling from a connected domain, then there always exists some sub-region of this domain that maps outside the desired image. More formally: $H \subset Z, |H| = n, |Z| = \aleph_0, n \in \mathbb{N} \implies \exists M \subset Z : \mathcal{G}(m) \notin \mathbb{H}\ \forall m \in M,$.

4. In conclusion, we show by contradiction that a multi-layer perceptron cannot produce a perfectly finite or solely discrete image when its domain is a connected space.

Symbol	Meaning
$\mathbb{A}$	Set of activities
$\mathbb{X}$	Set of all features
$\mathbb{C}$	Set of continuous features
$\mathbb{D}$	Set of discrete features
$\mathbf{D}_{tr}$	Training set
$\mathbf{D}_{te}$	Test set
$\mathcal{G}$	Generator
$\mathcal{D}$	Discriminator
P_R	Probability distribution of real data
P_G	Probability distribution of generated data.
$P_{(C,D)\|A=a}$	Probability distribution of continuous features X given that the activity is a.
$\mathcal{N}^m(\mu, \sigma)$	An m-dimensional Gaussian distribution with mean μ and covariance σ.

Table 1: Table of notation.

4 Background

To understand the generative models we will directly compare to HAR-CTGAN in our experiments, we have prepared a background that explains the architecture of them and their key qualities. We also analyze the key novelties of CTGANs and the attributes we will be using from CTGANs to modify state-of-the-art models.

4.1 Vanilla Generative Adversarial Networks

First proposed by Goodfellow et al. [26], generative adversarial networks consist of two neural networks, a generator G and discriminator D, that compete in a zero-sum game according to the following loss function $\min_G \max_D f(G, D)$,

$$\mathbb{E}_{x \sim p_{data}(x)}[log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z))] \tag{1}$$

in which G minimizes the loss of f by learning to map noise z to the continuous feature space $\mathbb{R}^m$ that the real data x lives in from a corpus $\mathbb{D}$. By drawing noise from random samples of a latent space Z according to some distribution, commonly Gaussian or uniform, G effectively learns to sample the unknown distribution $\mathbb{D}$ using a known distribution: Z. The discriminator D on the other hand maximizes f by being tasked with learning to discriminate between real data x and the synthetically generated data $G(z)$ when blindly given a batch mixed with both types of data. This tandem learns in an iterative and alternating fashion where only one of the two machines train for several epochs, and then its adversary trains for several epochs, and so on. This process continues until the end of training where, ideally, G finds a mapping that transforms random noise into synthetic that can consistently and effectively fool D. This results in G producing such realistic data that D can't effectively distinguish real from fake data, and its decision-making ability is analogous to randomly guessing.

4.2 Conditional Generative Adversarial Networks

One of the first improvements to the GAN architecture was the conditional GAN [35]. This new type of GAN modifies the structure of G and D to include additional class-specific information as input into each model, commonly coming in the form of class labels $y \in Y$. For the generator, this modification means G accepts both noise and a class label. This additional label conditions the generation to be compliant with the patterns of that specific class. For the discriminator, the real and synthetic data are passed in along with their associated class labels (the real label for real data, and the conditioned label for synthetic data). These architecture adjustments result in a new loss function $\min_G \max_D f(G, D)$ which can be written as

$$\begin{aligned} &\mathbb{E}_{x \sim p_{data}(x), y \sim p_y(y)}[log(D(x|y))]+ \\ &\mathbb{E}_{z \sim p_z(z), y \sim p_y(y)}[log(1 - D(G(z|y)|y))] \end{aligned} \tag{2}$$

The task of D has consequently changed, causing D not just to distinguish real from fake data, but additionally to learn whether a given datum is realistic and of the correct class according to its label y. This additional complexity allows for a class-tunable architecture in which data generation can be tuned to the desired class by the model practitioner when used post-training.

4.3 Controllable Generative Adversarial Networks

Controllable GANs are another class-tunable architecture [36] that builds upon conditional GANs with the inclusion of an additional pre-trained neural-net classifier, C. This classifier is tasked with identifying whether the synthesized, conditional data $G(z|y)$ best matches the class it was conditioned on during generation. Before training the generator or discriminator, C is fully trained exclusively on real data. For multi-class problems with k independent classes, C learns to minimize the categorical cross-entropy function $f(C)$

$$\min_C f(C) = -\sum_{i}^{K} k_i log(C(x)_i) \tag{3}$$

C effectively offloads work from the discriminator with the sole task of identifying intra-class patterns and inter-class differences on realistic, correctly labeled data. Similar to the landmark simple GAN, the discriminator D only needs to distinguish between real and fake blindly regardless of its associated class label. The generator G learns to synthesize conditional data $G(x|y)$ that still effectively fools the discriminator, but also agrees with the classifier as to what class the data was conditioned on. The loss $\min_G \max_D f(G, D, C)$ can be written as

$$\begin{aligned}
&\mathbb{E}_{x \sim p_{data},(y) \sim p_(y)}[log(D(x)] + \\
&\mathbb{E}_{z \sim p_z,(y) \sim p_(y)}[log(1 - D(G(z|y)))] - \\
&\mathbb{E}_{z \sim p_z,(y) \sim p_(y)}[log(C(G(z|y)|y))]
\end{aligned} \tag{4}$$

It is important to note that the classifier at no point sees real data *during* the GAN's training nor does it ever communicate with D since its job is to maintain the discriminator's simplicity while leveraging the class-tuning capabilities of conditional GAN architectures.

4.4 Conditional Tabular GAN

Proposed by Xu et al. [28] conditional GANs can be modified with several key attributes in order to effectively synthesize discrete data for tabular data sets to form Conditional Tabular GANs (CTGAN). The generator takes in random noise along with a random one-hot representation from a randomly chosen discrete feature. In this context, class labels and nominal features are both considered discrete features. When the generator synthesizes data, it generates synthetic that *includes* the condition the generation was conditioned on, as well as applies a Gumbel-Softmax activation function [38] to the discrete features in the synthetic data. When the discriminator trains, the discriminator receives a mixed batch of synthetic data and carefully selected real data that also follows the condition that the synthetic data was conditioned with.

The two following subsections lay out the attributes of CTGANs that will be incorporated into the state-of-the-art models to modify their behavior. By utilizing these attributes we will demonstrate how CTGAN components can be used to improve the generation quality of other GANs for HAR data synthesis as well.

4.4.1 Gumbel-SoftMax

Gumbel-SoftMax allows the one-hot vectorization of categorical distributions. In the context of deep generative modeling, this is helpful for generating exact one-hot vectors rather than normalized probability distributions. This is done by annealing (smoothing out) a discrete distribution into a continuous one. By doing so, a discrete distribution can be sampled from using a continuous distribution, which is especially impactful for machine learning models for the sake of backpropagation during training. Since techniques like ArgMax can only generate a one-hot vector in a way that can't be backpropagated, or SoftMax with can be backpropagated but only normalize a vector, Gumbel-SoftMax has the unique capability of both attributes.

A major driver of Gumbel-SoftMax depends on the tuning of its only hyperparameter τ, which can be set to any positive real-valued number $[0, \inf]$. When $\tau \to 0$, there isn't any annealing of the categorical distribution, and the probability distribution matches the expectation. When τ approaches larger values tending towards infinity, the categorical distribution is annealed so much so that the distribution transforms into a uniform distribution. In the context of using Gumbel-SoftMax for one-hot generation, low temperatures are used.

4.4.2 Wasserstein Loss

In this architecture, the discriminator is still tasked with discerning between real or fake data, but uniquely does not use a conventional log-loss activation function in its final layer. Instead, the discriminator leverages Wasserstein Divergence as the loss function [39] such as

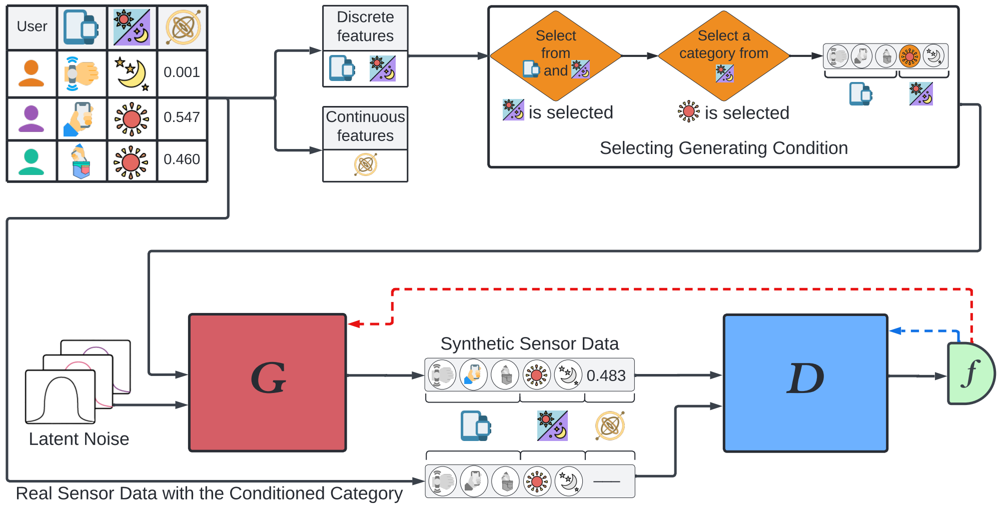


Figure 3: HAR-CTGAN schema with a dramatized mobile sensor dataset.

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim p_z}[f_w(g_\theta(z))] \tag{5}$$

in order to train the two machines according to the distance probability density function of the real data evaluated by the discriminator is away from the probability density function of the discriminator's evaluation of synthetic data.

5 Methodology

5.1 Dataset

To validate our framework, we employ the ExtraSensory dataset [20], a HAR dataset of over 300,000 instances of featured sensor data from in-the-wild recording on mobile devices from 60 users. The dataset covers a wide diversity in user ethnicities, user heights, user ages, types of activities performed, and mobile sensors employed.

Table 2: Breakdown of features in the *ExtraSensory* dataset.

Feature Type	No. of Features
Activities	7
Discrete	78
Continuous	192

5.1.1 Continuous Features

The data captured for the continuous features comes from raw data captured by accelerometer, gyroscope, magnetometer, compass, GPS, and microphone sensors which are then broken up into 1-minute intervals (also known as chunking). To distill these 1-minute chunks into a single data instance, the raw chunks are feature engineered and aggregated by using a series of transformations. This in turn yields a variety of features to consider for synthetic generation and input into downstream HAR models. The multitude of methods to engineer these features are described in Table 3.

mean()	Mean value
std()	Standard deviation
mad()	Median absolute deviation
max()	Largest value in array
min()	Smallest value in array
sma()	Signal magnitude area
energy()	Energy measure. Sum of the squares divided by the number of values.
iqr()	Interquartile range
entropy()	Signal entropy
arCoeff()	Autorregresion coefficients with Burg order equal to 4
correlation()	Correlation coefficient between two signals
maxInds()	Index of the frequency component with largest magnitude
meanFreq()	Weighted average of the frequency components to obtain a mean frequency
skewness()	Skewness of the frequency domain signal
kurtosis()	Kurtosis of the frequency domain signal
bandsEnergy()	Energy of a frequency interval within the 64 bins of the FFT of each window.
angle()	Angle between to vectors.

Table 3: Statistical functions we compute on the accelerometer and gyroscope data

5.1.2 Discrete Features

A portion of discrete features come from device states recorded from the device's operating system such as battery status, wifi availability, time-of-day, BlueTooth status, and screen status. The other portion of discrete features comes from contextual aspects that pertain to the conditions the sensor data was collected. For example, the location of the device on the user's body (ie. prioception) or the type of environment the user is in (indoors or outdoors) provides additional context to the data that allows for the continuous data to have different representations of the same activity is being performed. Consider when an individual is riding their bike outside on a path or side of the street versus when that same user rides a stationary bike at the gym. While the sensor data may look different in these two different environments, additional context allows HAR models to classify these tasks as identical. Additionally, these contextual features resolve a different issue in which the same continuous sensor data can be ambiguous to a variety of classes, such as in Figure 2.

5.2 Feature Engineering

Since the *ExtraSensory* mobile sensor data set consists of such a wide and diverse set of continuous and discrete features, it is important to properly extract the most meaningful features for a downstream HAR classification. Do so will guide our experiments as to what features are the most valuable for up-sampling with more instances of. In order to find these important features, we use random forest (RF) feature importance [40]. When utilized in the context of RF, the feature importance algorithm first quantifies how much each feature contributes to the final prediction of a tree, and then determines the average values of these contributions across the forest. More specifically, the RF feature importance [41] is computed by measuring the degree to which each feature reduces the Gini Index, defined as:

$$\text{Gini}(w) = \sum_{k=1}^{K} w_k \left(1 - w_k\right) = 1 - \sum_{k=1}^{K} w_k^2 \tag{6}$$

where K refers to the total number of features considered, and w_k represents sample weights.

Moreover, within a single tree's internal node m, the feature importance γ of x is

$$\gamma_{jm}^{(Gini)} = GI_m - GI_l - GI_r \tag{7}$$

where GI_l and GI_r are the Gini Index of the two child nodes after a split respectively. Given that a feature x appears in a decision tree i in nodes M, γ of x in the i-th tree is defined

$$\gamma_{ij}^{(Gini)} = \sum_{m \in M} \gamma_{jm}^{(Gini)} \tag{8}$$

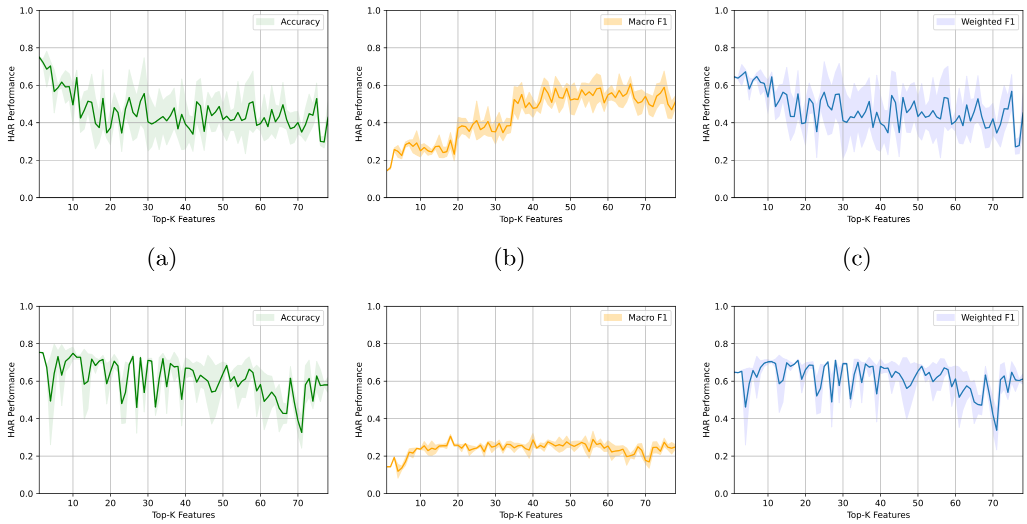
Furthermore, given that there are n trees in the forest

$$\gamma_{j}^{(Gini)} = \sum_{i=1}^{n} \gamma_{ij}^{(Gini)} \tag{9}$$

Finally, we normalize the values of γ by dividing a feature's importance by the total sum of all feature importance values.

$$\gamma_j = \frac{\gamma_j}{\sum_{i=1}^{c} \gamma_i} \tag{10}$$

To find the most important continuous and discrete features separately, we ran a set of stratified random forest feature importances where, for each of the 7 HAR activities, the feature importances for the continuous features and discrete features were evaluated separately, and given a score based on the positive modular additive inverse of its rank. These scores for the top discrete and continuous features were then aggregated respectively across the 7 activities and given their final rankings for their feature importance. From there, we perform stepwise feature selection to determine how many of the top-k most important discrete features and the top-j most important continuous features to consider for our final baseline feature input for a hypothetical downstream HAR model. In doing so, as seen in Figure 4, we choose to consider 35 discrete features and 18 continuous features for synthetic generation as well as input to our HAR classifier.



(a) (b) (c)

(d) (e) (f)

Figure 4:
Plots a-c. From left to right: accuracy, macro-f1, and weighted-f1. HAR model performance when employing stepwise selection of discrete features. When using 35 discrete features, macro-f1 begins to converge to just under 0.6, while accuracy and weighted oscillate just above 0.4 when incorporating additional features.
Plots d-f. From left to right: accuracy, macro-f1, and weighted-f1. HAR model performance when employing stepwise selection of continuous features. When using 18 continuous features, macro-f1 peaks and converges above 0.2, while accuracy and weighted oscillate with little change.

6 Experiments

6.1 Experimental Setup and Methodology

Comparative Models. To illustrate the performance of HAR-CTGAN, we compare our approach to a variety of models that fall into three primary classes of GANs: vanilla (simple), conditional, and controllable GANs. Within each of these architectures, we train a batch of 5 models with modifications spanning 3 varying degrees:

1. **None.** The generator has no additional post-processing to its output and functions in its standard *run-of-the-mill* fashion.

2. **SoftMax.** An additional SoftMax activation function is applied across each discrete one-hot vector separately when synthesized by the Generator before being passed to the Discriminator (and independent classifier for the controllable GANs).

3. **Gumbel-SoftMax.** An additional Gumbel-SoftMax [38] activation function ($\tau = 0.2$) is applied across each discrete one-hot vector separately when synthesized by the Generator before being passed to the Discriminator (and independent classifier for the controllable GANs). This technique is utilized in CTGAN frameworks.

We modify the generator's in these 3 degrees to progressively shift these models closer to a CTGAN architecture than their original architecture. This is done to show the positive effect CTGAN attributes have even when applied to architectures that weren't originally proposed for supporting them.

6.2 Metrics Used for Evaluation.

We evaluate GAN performance via the weighted average F1 score of a classifier trained on real data and evaluated on the GANs synthetic data. This metric require is found by computing the F1 score for each class,

$$F1(c) = \frac{precision(c) \cdot recall(c)}{precision(c) + recall(c)}.$$

Table 4: Machine evaluation study for comparing the synthetic data generated from each of the models. When using the model performances using each synthetic corpus as a proxy for evaluating the quality of each corpus, HAR-CTGAN's generation produced the most realistic corpus with respect to the original dataset.

Training Data	Post-Processing	Weighted Average F1		
		Continuous	Discrete Features	All Features
Original Corpus	N/A	0.794 (±0.003)	0.873 (±0.002)	0.958 (±0.002)
Vanilla GAN	None	0.386 (±0.121)	0.405 (±0.093)	0.454 (±0.116)
	SoftMax	0.496 (±0.040)	0.456 (±0.035)	0.536 (±0.020)
	Gumbel-Softmax	0.438 (±0.031)	0.421 (±0.068)	0.490 (±0.055)
Conditional GAN	None	0.266 (±0.100)	0.284 (±0.135)	0.311 (±0.141)
	SoftMax	0.249 (±0.067)	0.335 (±0.117)	0.338 (±0.131)
	Gumbel-Softmax	0.135 (±0.070)	0.244 (±0.148)	0.176 (±0.090)
Controllable GAN	None	0.222 (±0.032)	0.176 (±0.096)	0.238 (±0.068)
	SoftMax	0.294 (±0.033)	0.368 (±0.088)	0.405 (±0.052)
	Gumbel-Softmax	0.346 (±0.062)	0.325 (±0.092)	0.423 (±0.075)
HAR-CTGAN	**Gumbel-Softmax**	**0.629 (±0.005)**	**0.725 (±0.013)**	**0.742 (±0.004)**

By doing so, the Weighted Average F1 score is then calculated by

$$F1\_avg\_weighted = \frac{1}{|\mathcal{A}|}\sum_{c=1}^{|\mathcal{A}|}\frac{c'}{|\mathbb{D}_{tr}|}F1(c),$$

where c' is the number of instances of class c, where c is a specific user-activity pair. The delta metric is thus $F1\_avg\_weighted(real) - F1\_avg\_weighted(\mathcal{G})$, where $F1\_avg\_weighted(real)$ is the Weighted Average F1 of the real data and $F1\_avg\_weighted(\mathcal{G})$ is the Weighted Average F1 of the generated data from generative model $\mathcal{G}$.

6.3 Machine Evaluation

To validate our framework, we employ the HAR dataset from UCSD, *ExtraSensory* [20], which has an extensive amount of data that is diverse in the types of activities performed by individuals and the mobile sensors used. Diversity looks into how closely the generated data matches the patterns of real data. If the generated data is very realistic, it will match the pattern. However, we do not want to match the pattern exactly, as the generated data will be not meaningful and will not improve model performance.

In order to quantify how realistic the synthetic data from each of the GANs we use a common generative model evaluation technique [42] by utilizing additional classifiers independent of the generative model post-training. Firstly, partition our dataset into two subsets: a training set, and a test set. We then train 3 different hypothetical downstream HAR models with exact same architecture each time from scratch exclusively on real data from the training set, where each HAR model either only continuous features, only discrete, or both sets of features as input. After training these models, they are each evaluated on their performance using the test set, using weighted-average-f1 as the fitness metric. These three weighted-average-f1 scores serve as a baseline for comparing all of our GANS. Then, we train all 12 GAN variations including HAR-CTGAN using exclusively the same training set the baseline HAR models used. Each model is trained for 1,000 epochs with a learning rate of 2e-5 and a batch size of 500. Post GAN training, for every GAN trained we take a sample of 40,000 synthetic instances and train a HAR model with the same architecture as the baseline models this time using exclusively the fake data sample from its own generator. From there, we evaluate the weighted-average-f1 scores from each of these models using the same fixed test set that both the GAN and the respective HAR models have never seen during training.

Models that achieved fitnesses closest to the fitness of the benchmark mean the synthetic data is most realistic and ergo the machine it was synthesized from has the best generation quality. Fitness scores closer to 0 or that surpass the benchmark towards 1 correspond to poor, unrealistic generation quality. While its unintuitive that having a high fitness score can be an undesired trait, this means that the generator's synthesis is oversimplifying the patterns in the real data and does not exhibit a synthesis that will lie in the same distribution as real data. Ultimately, this leads to meaningless data when used to upsample the dataset. The results in Table 4 that HAR-CTGAN's generator, which has never seen real data, was able to build the classifier closest to training on real data than any other GANs. This shows that the generated data from HAR-CTGAN is more realistic than the generated data from other models.

7 Discussion

7.1 Limitations

Despite the success of HAR-CTGAN in the continuous and discrete synthesis of mobile sensor data, there are several points of failure in which this approach can become sub-optimal or, in extreme scenarios, fail completely. Since HAR-CTGAN is rooted in a generative adversarial framework, it is no less susceptible to unstable training than other GAN architectures mentioned. GANs are notoriously susceptible to oscillating converge or failure to learn in training due to mode collapse or overfitted discriminators.

7.2 Future Work

Generative adversarial paradigms require large corpora in order to effectively have its generator machine synthesize realistic data. With more and more limited examples in tje corpora, generative models commonly begin to fail as their task of generating new meaningful data that interpolates potentially non-linear patterns becomes harder and harder. In the case of the *ExtraSensory* dataset, there are adequate volumes of mobile sensor instances despite the stark class imbalances present. Burgeoning techniques in generative modeling have explored avenues to learn disentangled representations of real data to extract greater meaning from each instance to distinguish underlying global patterns from extraneous ones. One avenue of future work would be to explore ways to apply HAR-CTGAN concepts for realistic discrete feature generation on sparse mobile sensor datasets.

8 Conclusion

In this paper, we have identified and shown the efficacy of a new tool to remedy an open problem that heavily impacts HAR applications across the domain. Up-sampling tools that counteract class imbalances in mobile sensor datasets lead to HAR models that can fully utilize their cleaned data without having to jeopardize poor multi-class performance nor discard expensive-to-collect data in order to down-sample for uniform class distributions. We identify state-of-the-art approaches that perform synthetic up-samplings of class-tailored HAR data still lack the ability to generate discrete contextual HAR features that are realistic. We propose our approach, HAR-CTGAN, which is a Conditional Tabular GAN that learns to generate synthetic data of the continuous and discrete features in HAR data. We evaluated our performance against multiple state-of-the-art architectures on a publicly available benchmark HAR dataset. Our results show that HAR-CTGAN consistently outperforms the state-of-the-art models.

When state-of-the-art models are modified to have properties of HAR-CTGAN, the modified model's generation qualities improve. This further emphasizes how impactful the characteristics of HAR-CTGAN are for improving GAN generation quality and GAN training stability. In short, HAR-CTGAN provides greater flexibility for training high-quality downstream classification models with the best features, whether they are continuous or discrete, for passive healthcare monitoring via mobile devices.

References

[1] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 48–53, 2010.

[2] W. Gerych, H. Kim, J. DeOliveira, M. Martin, L. Buquicchio, K. Chandrasekaran, A. Alajaji, H. Mansoor, E. Rundensteiner, and E. Agu, "Gan for generating user-specific human activity data from an incomplete training corpus," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 4705–4714, 2021.

[3] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019. The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International

Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology.

[4] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1321–1330, 2015.

[5] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, "Trends in human activity recognition using smartphones," *Journal of Reliable Intelligent Environments*, pp. 1–25, 2021.

[6] W. Niu, J. Long, D. Han, and Y.-F. Wang, "Human activity detection and recognition for video surveillance," in *2004 IEEE international conference on multimedia and expo (ICME)(IEEE Cat. No. 04TH8763)*, vol. 1, pp. 719–722, IEEE, 2004.

[7] S. Liu, L. Zhang, Y. Long, Y. Long, and M. Xu, "A new urban vitality analysis and evaluation framework based on human activity modeling using multi-source big data," *ISPRS International Journal of Geo-Information*, vol. 9, no. 11, p. 617, 2020.

[8] P. Castillejo, J.-F. Martinez, J. Rodriguez-Molina, and A. Cuerva, "Integration of wearable devices in a wireless sensor network for an e-health application," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 38–49, 2013.

[9] A. Barua, A. K. M. Masum, E. H. Bahadur, M. R. Alam, M. A. U. Z. Chowdhury, and M. S. Alam, "Human activity recognition in prognosis of depression using long short-term memory approach," *International Journal of Advanced Science and Technology*, vol. 29, pp. 4998–5017, 2020.

[10] W.-Y. Cheng, A. Scotland, F. Lipsmeier, T. Kilchenmann, L. Jin, J. Schjodt-Eriksen, D. Wolf, Y.-P. Zhang-Schaerer, I. F. Garcia, J. Siebourg-Polster, *et al.*, "Human activity recognition from sensor-based large-scale continuous monitoring of parkinson's disease patients," in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pp. 249–250, IEEE, 2017.

[11] S. Rahman, S. F. Ahmed, O. Shahid, M. A. Arrafi, and M. Ahad, "Automated detection approaches to autism spectrum disorder based on human activity analysis: A review," *Cognitive Computation*, pp. 1–28, 2021.

[12] M. L. Hoang, M. Carratù, V. Paciello, and A. Pietrosanto, "Body temperature—indoor condition monitor and activity recognition by mems accelerometer based on iot-alert system for people in quarantine due to covid-19," *Sensors*, vol. 21, no. 7, p. 2313, 2021.

[13] L. Zhang, Y. Zhu, M. Jiang, Y. Wu, K. Deng, and Q. Ni, "Body temperature monitoring for regular covid-19 prevention based on human daily activity recognition," *Sensors*, vol. 21, no. 22, p. 7540, 2021.

[14] C. Zhu and W. Sheng, "Multi-sensor fusion for human daily activity recognition in robot-assisted living," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 303–304, 2009.

[15] S. Zolfaghari and M. R. Keyvanpour, "Sarf: Smart activity recognition framework in ambient assisted living," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 1435–1443, 2016.

[16] G. Sebestyen, I. Stoica, and A. Hangan, "Human activity recognition and monitoring for elderly people," in *2016 IEEE 12th international conference on intelligent computer communication and processing (ICCP)*, pp. 341–347, IEEE, 2016.

[17] Z. A. Khan and W. Sohn, "Abnormal human activity recognition system based on r-transform and kernel discriminant technique for elderly home care," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1843–1850, 2011.

[18] M. B. Rasheed, N. Javaid, T. A. Alghamdi, S. Mukhtar, U. Qasim, Z. A. Khan, and M. H. B. Raja, "Evaluation of human activity recognition and fall detection using android phone," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, pp. 163–170, IEEE, 2015.

[19] J. R. Villar, S. González, J. Sedano, C. Chira, and J. M. Trejo-Gabriel-Galan, "Improving human activity recognition and its application in early stroke diagnosis," *International journal of neural systems*, vol. 25, no. 04, p. 1450036, 2015.

[20] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE pervasive computing*, vol. 16, no. 4, pp. 62–74, 2017.

[21] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2021.

[22] K. T. Nguyen, F. Portet, and C. Garbay, "Dealing with imbalanced data sets for human activity recognition using mobile phone sensors," in *3rd International Workshop on Smart Sensing Systems*, 2018.

[23] J. Wang, Y. Chen, Y. Gu, Y. Xiao, and H. Pan, "Sensorygans: an effective generative adversarial framework for sensor-based human activity recognition," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[24] F. Alharbi, L. Ouarbya, and J. A. Ward, "Synthetic sensor data for human activity recognition," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, IEEE, 2020.

[25] J. Shi, D. Zuo, and Z. Zhang, "A gan-based data augmentation method for human activity recognition via the caching ability," *Internet technology letters*, vol. 4, no. 5, p. e257, 2021.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[27] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of gans," *Advances in neural information processing systems*, vol. 30, 2017.

[28] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[30] B. Fergani *et al.*, "Comparing hmm, lda, svm and smote-svm algorithms in classifying human activities," in *Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015*, pp. 639–644, Springer, 2016.

[31] B. A. M'hamed and B. Fergani, "A new multi-class wsvm classification to imbalanced human activity dataset.," *J. Comput.*, vol. 9, no. 7, pp. 1560–1565, 2014.

[32] B. M. Abidine, B. Fergani, M. Oussalah, and L. Fergani, "A new classification strategy for human activity recognition using cost sensitive support vector machines for imbalanced data," *Kybernetes*, 2014.

[33] C. Chen, A. Liaw, L. Breiman, *et al.*, "Using random forest to learn imbalanced data," *University of California, Berkeley*, vol. 110, no. 1-12, p. 24, 2004.

[34] Y. Guo, Y. Chu, B. Jiao, J. Cheng, Z. Yu, N. Cui, and L. Ma, "Evolutionary dual-ensemble class imbalance learning for human activity recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[35] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[36] M. Lee and J. Seok, "Controllable generative adversarial network," *Ieee Access*, vol. 7, pp. 28158–28169, 2019.

[37] U. Tanielian, T. Issenhuth, E. Dohmatob, and J. Mary, "Learning disconnected manifolds: a no gan's land," in *International Conference on Machine Learning*, pp. 9418–9427, PMLR, 2020.

[38] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[39] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

[40] J. Yu, C. Xia, J. Xie, and H. Zhang, "Research on feature importance of gait mechanomyography signal based on random forest," in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, IEEE, 2020.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, 2011.

[42] F. Alharbi, L. Ouarbya, and J. A. Ward, "Synthetic sensor data for human activity recognition," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2020.