HAR-CTGAN: A Mobile Sensor Data Generation Tool for Human Activity Recognition

Joshua DeOliveira[1], Walter Gerych[1], Aruzhan Koshkarova[1,2], Elke Rundensteiner[1], and Emmanuel Agu[2]

[1]Data Science Program, Worcester Polytechnic Institute, Worcester, MA
[2]Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020848

***Abstract*—Human activity recognition (HAR) is the process of using mobile sensor data to determine the physical activities performed by individuals. 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 architectures and big data in order to accurately classify activities. 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 by real numbers — recorded from gyroscopes, accelerometers, and other sensors that produce continuous data. This is limiting since mobile sensor data commonly has discrete features that provide additional context such as device location and the time-of-day, which have been shown to substantially improve HAR classification. 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, and when state-of-the-art models were modified with HAR-CTGAN characteristics, downstream performance also improves.**

***Index Terms*—GAN, CTGAN, mobile healthcare, human activity recognition, sensor data, synthetic data generation, discrete features.**

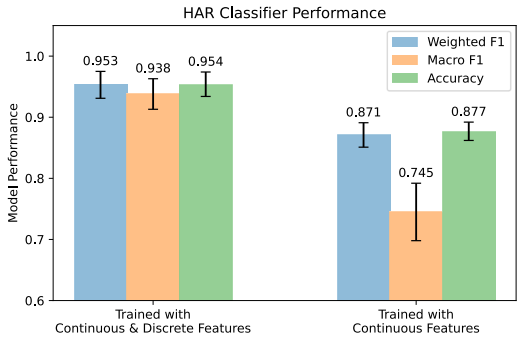

Figure 1: HAR classifier model performances (averaged over 10 HAR classifiers of varying complexity), each trained to classify 7 unique activities found in *ExtraSensory* [20]. Classifiers that used both discrete and continuous features as input (on left) yielded consistently much better performances than classifiers that ignored discrete features and only accepted continuous features (on right). This indicates that the ability to accept discrete features can enrich model quality.

I. Introduction

Background. Human Activity Recognition (HAR) is the task of using sensor data to classify the physical activities performed by individuals [1]. These physical activities are typically day-to-day low-level tasks, such as walking, standing, and sitting [2], [3]. Data is captured by a variety of sensors such as gyroscopes, accelerometers, and GPS trackers among others [4]. Advancements in high-precision sensors along with the heightened ubiquity of wearable technology that embed these sensors [5], such as smartphones and smartwatches, have led to rapidly greater access to high volumes of HAR 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 variety in people's behavior and the manner in which they perform activities [3]. To create HAR models that perform well on such diverse real-world data, state-of-the-art approaches train models on *in-the-wild* datasets [20]. In-the-wild HAR datasets utilize passively collected data from mobile devices 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]. 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. For example biking, swimming, walking, and sleeping are common HAR activities researchers record mobile sensor data. In an average day, most people may spend 6-8 hours a day sleeping, spend only an hour or two exercising in some way, and walking occasionally throughout the day. In this case, all

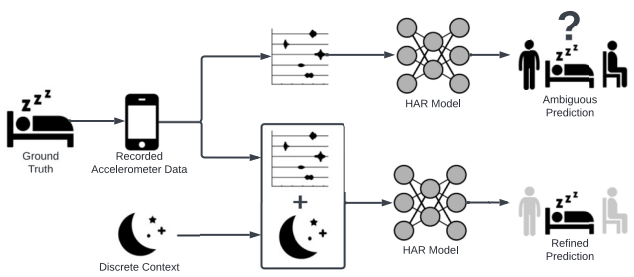

Figure 2: When HAR models only use continuous sensor features due to the upsampling limitations of GANs, it can be difficult to distinguish similar activities. But, when contextual discrete features are leveraged, refined classifications can be made from previously ambiguous data. For example, sensor data when sleeping can look similar to data when sitting or standing, leading to potential misclassifications. But, when sleeping data is coupled with a discrete feature indicating the time of day (night time), then a HAR model can correctly classify the data with high confidence.

exercising activities are rarely ever seen, making up about 4%-8% of all recorded data in a 24-hour cycle.

Furthermore, in the healthcare field specifically, HAR has been used to develop early warning systems to 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 depend on 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, there lie several critical hurdles in achieving data that can reasonable work for HAR systems. In the age of big data in which mobile sensor datasets are large in volume, diverse in their variety, and semi-trustworthy in their veracity, existing simple solutions to address these data issues can be expensive, laborious, or simply 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, accurately labeled, high-quality mobile sensor datasets without suffering the potential down-sides of financial limitations and time-constraints.

State-of-the-Art. State-of-the-art solutions for the above data quality and imbalance issues 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 deep learning framework in which two 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 the judge or critic, is tasked with deciphering which data is real and fake when given mixed batches of samples taken from the original dataset and the generator.

While many GAN approaches have been proposed in the HAR setting in an attempt to mend various data quality 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 [27].

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 is an open and 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 [28]. Second, modeling discrete features is difficult as this involves making a discrete choice, for which backpropagating through is not straightforward [27]. Additionally, the large diversity in HAR datasets can lead to multiple individuals performing the same activity in different styles depending on body mechanics and the use of assistive equipment [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. We propose to develop a HAR GAN data synthesizer for both continuous and discrete data, which is the first solution to this open problem. To achieve this, we propose the use of a recently developed Conditional Tabular Generative Adversarial Network (CTGAN) [27], which is a GAN developed for tabular data. CTGAN to-date has not

been studied in the HAR domain. We thus apply CTGAN to the task of generating in-the-wild HAR data, and refer to the resulting model as HAR-CTGAN. We perform an extensive set of experiments to validate not only that CTGAN can accurately generate discrete features, a task existing HAR GANs fail at, but also improves the quality of generated data overall. Furthermore, we study each component of CTGAN, and develop compared models that meld unique components of CTGAN with existing GAN approaches. Through this, we demonstrate that every component of CTGAN is necessary to the generation of realistic HAR data; models that use only some components perform significantly worse than the whole.

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.
- Demonstrating synthetic data from HAR-CTGAN yields more realistic generation quality than that of the state-of-the-art GANs when trained 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.

II. Related Work

A. Non-GAN Solutions for HAR Class Imbalances

Before the proliferation of neural-network-based generative modeling, novel HAR data generation was achieved through a variety of other machine learning techniques, notably 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]. These approaches 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.

B. GAN-Based HAR Data Generation

Due to HAR becoming a growing area of research over the past decade [1], [4], [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], [27], [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.

III. Problem Definition

Suppose we are given a real dataset $\mathbb{X} = \{(x_c, x_d, a)_i\}_{i=1}^{N}$, where $x_c \in \mathbb{C}$ corresponds to a tuple representing an instance of the continuous features (e.g. accelerometer features), $x_d \in \mathbb{D}$ corresponds to a tuple representing an instance of the discrete features (prioception, device locked/unlocked, bluetooth on/off, etc.), $a \in \mathbb{A}$ 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 $\forall a \in \mathbb{A}$ $\mathcal{G}(a) \sim P_R((\mathbb{C}, \mathbb{D})|\mathbb{A} = a)$. Thus, we want 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. The notation used in this work is shown in Table I.

Symbol	Meaning
$\mathbb{A}$	Set of activities
$\mathbb{C}$	Set of continuous features
$\mathbb{D}$	Set of discrete features
$\mathbb{X}$	Real Dataset
$\mathcal{G}(\cdot)$	Generator
$\mathcal{D}(\cdot)$	Binary Discriminator
$f_w(\cdot)$	Wasserstein Discriminator
P_R	Probability Distribution of Real Data
$P_{(C,D)\|A=a}$	Probability distribution of continuous features X given that the activity is a.
$\mathcal{N}(\mu, \sigma)$	An multivariate Gaussian distribution with mean μ and covariance σ.

Table I: Table of notation.

IV. Background

Before describing our HAR-CTGAN approach, we briefly describe the process behind standard GANs [26].

A. Standard Generative Adversarial Networks

First proposed by Goodfellow et al. [26], generative adversarial networks consist of two neural networks, a generator $\mathcal{G}$ and discriminator $\mathcal{D}$, that compete in a zero-sum game according to the following optimization equation $\min_{\mathcal{G}} \max_{\mathcal{D}} f(\mathcal{G}, \mathcal{D})$,

$$\mathbb{E}_{x\sim\mathbb{P}(\mathbb{X})}[\log(\mathcal{D}(x))] + \mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma)}[\log(1-\mathcal{D}(\mathcal{G}(z))] \quad (1)$$

in which $\mathcal{G}$ 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 $\mathbb{X}$. By drawing noise from random samples of a latent space Z according to some easy-to-sample-from distribution, commonly Gaussian or uniform, $\mathcal{G}$ effectively learns to sample the unknown distribution P_R using a known distribution. The discriminator $\mathcal{D}$ on the other hand maximizes f by being tasked with learning to discriminate between real data x and the synthetically generated data $\mathcal{G}(z)$ when given

a batch mixed with both types of data. The networks learn 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, $\mathcal{G}$ finds a mapping that transforms random noise into synthetic data that can consistently and effectively fool $\mathcal{D}$. This results in $\mathcal{G}$ producing such realistic data that $\mathcal{D}$ can't effectively distinguish real from fake data, and its decision-making ability is analogous to randomly guessing.

B. Existing HAR GANs Can Not Generate Discrete Data.

Using the assumptions presumed in Tanielian et al. [37], the following proposition holds in which the generator architecture that is utilized in start-of-the-art HAR GAN models (i.e., any of the methods described in the Related Works section [2], [23]–[25]) is unable to learn to perfectly generate discrete features from latent noise.

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 a homomorphism that relates every multi-layer perceptron to a vector-valued function f. 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 $\mathbb{H} \subset \mathbb{R}^m$ 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 must exist some sub-region $M \subset Z$ where $\mathcal{G}(m) \notin \mathbb{H}\ \forall m \in M$.
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.

Thus, a discrete data generator can not be achieved by naively applying the above GAN methods to discrete HAR data. Rather, a more sophisticated approach is required.

V. The HAR-CTGAN Methodology

In this section we describe the HAR-CTGAN approach, which utilizes the *Conditional Tabular GAN* (CTGAN) method proposed by Xu et al. [27]. CTGAN succeeds in generating discrete as well as continuous features by utilizing a Gumbel-Softmax activation function [38], which allows for efficient backpropagation through the sampling of discrete variables. Additionally, CTGAN learns a faithful representation of the *joint* distribution between discrete and continuous variables such that the generated continuous variables are logically consistent with the discrete variables and vice versa by utilizing a novel *discrete conditioning* training process. Furthermore, CTGAN learns by attempting to minimize/maximize a divergence metric, Wasserstein Divergence, as its loss function rather than conventional cross-entropy. These steps are explained in more detail below.

A. Discrete Conditioning

When sampling noise for the generator's synthesis, the generator is given additional conditioning information that allows for the generator to learn realistic synthesis with complete coverage across the discrete space. For each batch of noise sampled for the generator's input, the noise is coupled with a mask of all the discrete features to be generated. Within the discrete feature mask, one of the discrete features is randomly chosen according to a uniform distribution for conditioning. Then, a randomly chosen one-hot representation is chosen to represent the previously chosen discrete feature. This chosen feature representation tailors the generator's synthesis to be compliant with the chosen condition. As a result, for a well-trained generator, all of the synthetic data in the generated batch has the same one-hot representation for the discrete feature used for conditioning. Discrete conditioning allows for practitioner-desired generation in order to yield synthetic examples of realistic instances with certain properties.

When the discriminator is learning during training, this conditional sampling mask doubles as a filtering condition for choosing what real data is chosen for mini-batches passed to the discriminator. If applied to a simple GAN, it would affect the loss function $\min_{\mathcal{G}} \max_{\mathcal{D}} f(\mathcal{G}, \mathcal{D})$ to then become:

$$\begin{aligned} &\mathbb{E}_{d\sim P_{D\cup A}} \mathbb{E}_{c\sim P_d} \mathbb{E}_{x\sim P(\mathbb{X}|c)} \log[\mathcal{D}(x)] \\ &+\mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma)} \log[1-\mathcal{D}(\mathcal{G}(z|c))] \end{aligned} \tag{2}$$

where $P_{D\cup A}$ is the distribution over discrete features and activities, c is a value of a discrete variable or activity, and P_d is the probability distribution of possible values for discrete variable d. By convention, P_d follows the log-probability of the frequency of c for d.

Having each batch the discriminator sees have at least one discrete feature with an identical representation with respect to the synthetic batch of data forces the job of the discriminator to be more challenging. Thus, the discriminator must learn to distinguish a batch of real data from synthetic data that is the most similar looking to the real batch that the generator can currently produce. This is beneficial to the generator-discriminator tandem since this limits the chances of the discriminator being able to simply memorize specific real data

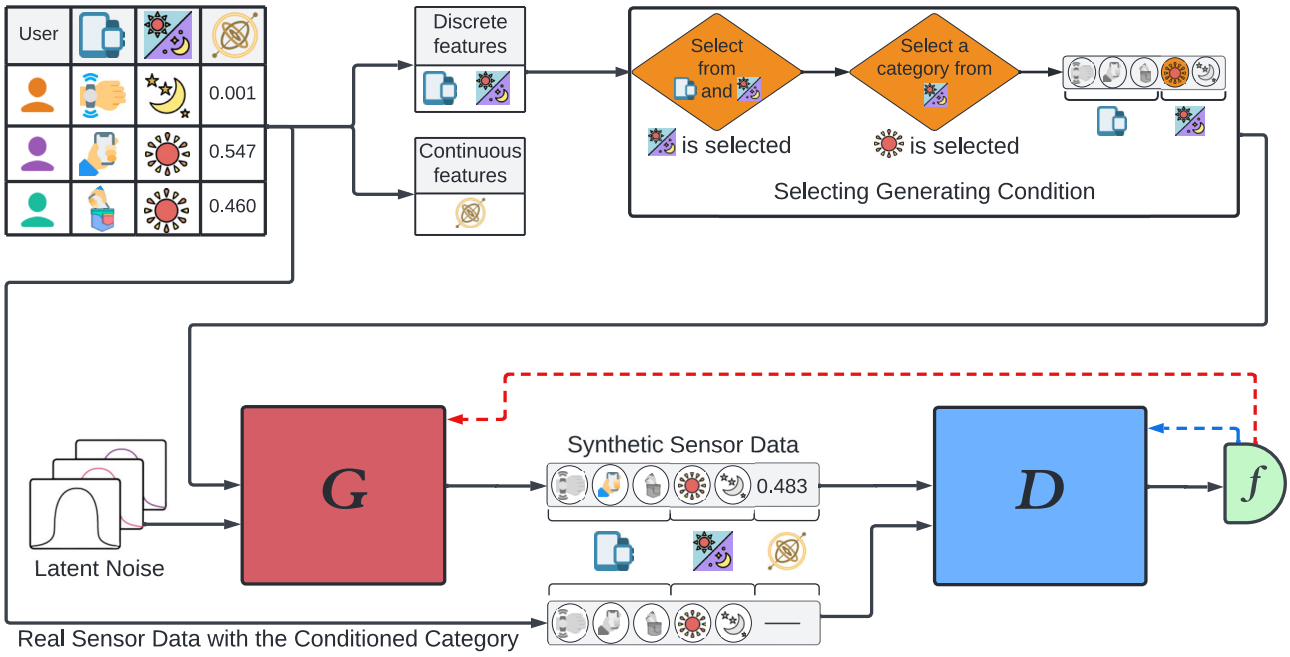

Figure 3: HAR-CTGAN

instances and consequently overfitting on the real data distribution. Discrete conditioning for driving minibatch creation also aids in combating mode collapse for the generator. If there are certain combinations of discrete feature representations that fool the discriminator very well (i.e. one mode for which the generator receives low cost), the generator's compliance with generating the chosen condition limits the generator from having complete freedom to generate synthetic data in only the low-cost mode.

B. Gumbel-SoftMax

Gumbel-SoftMax allows for the one-hot vectorization of categorical distributions. In the context of deep generative modeling, this is particularly helpful for generating exact one-hot vectors rather than normalized probability distributions. By annealing, or smoothing out, a discrete distribution into a continuous one, a discrete distribution can be sampled from using an approximate continuous distribution. Continuity is especially impactful in the context of deep models for the sake of backpropagation during training. Since techniques like ArgMax (forcing the greatest-valued entry in a tensor to one, and forcing all other values to zero) can only generate a one-hot vector in a way that can't be backpropagated, or SoftMax (tensor sigmoid normalization) which can be backpropagated but can't generate purely one-hot vectors, Gumbel-SoftMax has the unique capability of both attributes. For a given discrete feature x_d with k representations, the value of each entry x_{d_i} is computed by the following expression:

$$\hat{x}_{d_i} = \frac{e^{\frac{\log(x_{d_i})+g_i}{\tau}}}{\sum_{j=1}^{k} e^{\frac{\log(x_{d_j})+g_j}{\tau}}} \quad \text{for } i = 1, \ldots, k \tag{3}$$

where g is sampled from the following distribution

$$g = -\log[-\log(u)] \quad \text{where} \quad u \sim \mathcal{U}(0,1) \tag{4}$$

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 $[0, \inf)$. As τ approaches zero, the categorical distribution is not annealed and the probability distribution matches the expectation. Conversely, as τ approaches larger values tending towards infinity, the categorical distribution is annealed to such as degree that the distribution transforms into a uniform distribution. In the context of using Gumbel-SoftMax for one-hot generation, low temperatures are used.

C. Wasserstein Loss

The overall goal of the generator is to match the distribution of real data. Divergence metrics are a class of measures used to quantify the *dissimilarity* between two probability distributions. Therefore, one can train the generator and discriminator to directly minimize or maximize a divergence metric. While Equation 1 represents the standard GAN divergence, optimizing for this equation is known to be classically unstable [39].

Instead of the standard GAN divergence, we optimize for the Wasserstein loss [39]. Wasserstein loss is an objective function for GANs that directly aims at utilizing Wasserstein divergence

for training, which is often more stable than standard GAN training [39]. The discriminator is still tasked with discerning between real or fake data, but uniquely does not use any non-linear activation function in its final layer. Wasserstein loss for GANs is then optimizing $\min_{\mathcal{G}} \max_{\mathcal{D}} f(\mathcal{G}, \mathcal{D})$ as follows in Equation 5:

$$\mathbb{E}_{x \sim P(\mathbb{X})} f_w[x] - \mathbb{E}_{z \sim \mathcal{N}(\mu,\sigma)} f_w[\mathcal{G}(z)], \tag{5}$$

where f_w equal $\mathcal{D}$ with an additional Lipschitz constraint [39]. To train the two machines without knowing the distribution or real data directly, the Wasserstein distance is calculated using the distribution of the real data evaluated by the discriminator and the distribution of the discriminator's evaluation on synthetic data. When this loss is incorporated into the discrete conditional procedure of CTGAN, the resulting loss function $\min_{\mathcal{G}} \max_{\mathcal{D}} f(\mathcal{G}, \mathcal{D})$ is then:

$$\mathbb{E}_{d \sim P_{D \cup A}} \mathbb{E}_{c \sim P_d} \left[\mathbb{E}_{x \sim P(\mathbb{X}|c)} f_w[x] - \mathbb{E}_{z \sim \mathcal{N}(\mu,\sigma)} f_w[\mathcal{G}(z|c)] \right] \tag{6}$$

Thus, Equation 6 represents the optimization equation for our HAR-CTGAN approach.

VI. Results

A. Experimental Setup and Methodology

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 recordings 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. [1]

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

Feature Type	No. of Features
Activities	7
Discrete	48
Continuous	192

1) Activities: Each instance of mobile sensor data is coupled with a label corresponding to one of seven mutually exclusive activities — lying down, cleaning, sleeping, sitting, walking, running, and bicycling. We can also intuitively group certain activities together based on their similarity in motion. For example, we can split the labels into two main groups: idle activities and dynamic activities. Idle activities like lying down, sleeping, and sitting all are different human-interpretable representations of an individual performing little to no movement. Likewise, dynamic activities like walking, running, and bicycling are all representations of individuals constantly moving their bodies in different ways or under different circumstances. Lastly, the "cleaning" activity is a somewhat dynamic, somewhat static activity. While it may be simple to distinguish an idle activity from a dynamic one, the primary ways to distinguish idle activities from one another or dynamic activities from one another primarily lies in the context in which they are performed.

2) 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 likewise input into a downstream HAR model. The multitude of methods to engineer these features are described in Table III.

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 III: Statistical functions we compute on the accelerometer and gyroscope data

3) Discrete Features: A portion of discrete features come from device states recorded from the device's operating system such as battery status, screen status, ringer settings, wifi availability, and time-of-day. 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 was in (indoors/outdoors) provides additional context to the data that would be difficult or even impossible to extract directly from the continuous data. Additionally, consider when an individual is riding their bike outside on a path 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. Ergo, contextual features can also resolve issues in which the same continuous sensor data can be ambiguous to a variety of classes, such as in Figure 2.

Comparative Models. To illustrate the performance of HAR-CTGAN, we compare our approach to a variety of

[1]Code and supplementary material relating to the cleaning and feature engineering of the data used from *ExtraSensory* in this work is all open-sourced at `https://github.com/deoliveirajoshua/HAR-CTGAN`

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 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* implementation.
2) **SoftMax.** An additional SoftMax activation function is applied across each discrete one-hot vector individually when synthesized by the Generator before being passed to the Discriminator (and independent classifier for the controllable GAN).
3) **Gumbel-SoftMax.** An additional Gumbel-SoftMax [38] activation function with hyperparameter $\tau = 0.2$ applied to each discrete one-hot vector separately when synthesized by the Generator before being passed to the Discriminator (and independent classifier for the controllable GANs). The use of Gumbel-Softmax is native to CTGAN frameworks.

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

B. 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:

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

By doing so, the weighted average F1 score is then calculated by:

$$F1\_avg\_weighted = \frac{1}{|\mathbb{A}|} \sum_{a}^{|\mathbb{A}|} \frac{a'}{|\mathbb{X}|} F1(a),$$

where a' is the number of instances of class a, where a is a specific activity.

To quantify the generation quality of synthetic data through another perspective, we utilize our own metric named *ambiguity score*. When a balanced test set is passed into a shallow model such as a soft-margin SVM, logistic regression, or k-means, the ambiguity metric describes the percentage of the test set that is incorrectly classified. Whether it be synthetic data that the shallow model predicted as real or real data that the shallow model predicted as fake, the ambiguity score encapsulates both scenarios.

$$Ambiguity\_Score = \frac{FP + FN}{TP + TN + FP + FN}$$

This metric is also analogous to calculating $(1 - Accuracy)$ but depends on having balanced classes in order to have a trustworthy scoring of the generated data quality.

C. 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 both the types of activities performed by individuals and the mobile sensors used for data capture. A machine evaluation study is one avenue for examining 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 the performance of downstream models.

In order to quantify the realism of the synthetic data generated from each of the GANs, we use a common generative model evaluation technique [40] by utilizing additional classifiers independent of the generative model post-training. These classifiers are deep HAR classifiers that are designed to mimic some plausible downstream HAR task utilizing mobile sensor data as input. The goal of the HAR classifier is to classify the given mobile sensor instance into one of the seven mutually exclusive activities mentioned previously.

Firstly, we partition our real dataset into two subsets: a training set, and a hidden 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 accepts either only continuous, only discrete, or both sets of features as input. After training these models, they are each evaluated on their performance using the hidden test set and scored 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 that the baseline HAR models used*. Each GAN model trained for 1,000 epochs with a learning rate of 2e-5 and a batch size of 500 with the same generator and discriminator architectures. After GAN training, 40,000 synthetic instances were sampled from each GAN and individually used to train HAR models with the same architecture as those used for the baseline models. This time, exclusively training on the fake data sampled from each GAN's generator. From there, we evaluate the weighted-average-f1 scores from each of these models using the same fixed test set that both the GANs and the respective HAR models have never seen during training.

Models that achieve a fitness closest to the fitness of the benchmark mean the synthetic data is most realistic and thus the GAN it was synthesized from yields the best generation quality. Fitness scores closer to 0 correspond to poor, unrealistic generation quality by failing to mimic the patterns in the real data. This effect may be a result of the generator's synthesis oversimplifying the patterns in the real data or failing to generalize these patterns in any consistent way. In either case, upsampling datasets with data that is meaningless or misleading will harm model performance, especially in minority classes that are upsampled with synthetic data the most. The results in Table IV show that HAR-CTGAN's

Table IV: Machine evaluation study for comparing the synthetic data generated from each of the state-of-the-art GAN models and their modified variations. When using the performances of downstream HAR models trained using synthetic corpora as a proxy for evaluating the quality of each generated corpus, HAR-CTGAN's generation produced a synthetic corpus that was more realistic than any other synthetic corpora tested. When the generated data is broken down into the synthesized discrete and synthesized continuous features, the generation HAR-CTGAN is the most realistic with respect to the real mobile sensor dataset in both categories.

Training Data	Post-Processing	Weighted Average F1 By Feature Type		
		Continuous	Discrete	All
Original Corpus	N/A	0.794 (±0.003)	0.873 (±0.002)	0.958 (±0.002)
Vanilla GAN	None	0.620 (±0.027)	0.635 (±0.029)	0.714 (±0.004)
	SoftMax	0.496 (±0.040)	0.222 (±0.302)	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**	**0.754 (±0.010)**	**0.807 (±0.011)**	**0.766 (±0.006)**

generator, which has never directly 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.

D. Separability

One mode of evaluating generation quality is to directly test how indistinguishable the synthetic data from a generative model is in comparison to the original real dataset. If synthetic data is truly indistinguishable from real data, binary classifiers should struggle to classify all real data as real and all fake data as fake. Therefore, we trained a shallow that was independent of the GAN's training and tasked it with binary classification where one class was entirely real mobile sensor data and the other class is synthetic data generated from one of the trained GANs. Using a shallow model trained with balanced classes inhibits the chances of overfitting. For our shallow model, we chose to use a logistic regression model.

Table V: Separability study for determining how indistinguishable synthetic data is from real data when comparing the continuous features, discrete features, and the entire data instance from each other. State-of-the-art GANs produce data that is easily separable due to an inability to generate every feature realistically as well as unstable training.

Model	Ambiguity Score		
	Continuous	Discrete	All
Vanilla GAN	0.000 (± 0.000)	0.004 (± 0.003)	0.000 (± 0.000)
HAR-CTGAN	0.121 (± 0.008)	0.409 (± 0.011)	0.112 (± 0.009)

When generation quality is extremely poor, the patterns in the synthetic data would appear very different than those in real data. Consequently, the SVM would be able to perfectly distinguish real data from synthetic, and the synthetic data would receive an ambiguity score of 0. However, when the generation quality of the synthetic data from a GAN is superb, the synthetic is so indistinguishable from real data that the SVM fails to learn the differences between real and synthetic data, producing a perfect 0.5 ambiguity score. Table V illustrates that HAR-CTGAN produces a far better ambiguity score than even some of the best performing GANs from the machine evaluation study (Table IV). This can be explained by the non-HAR-CTGANs struggle to produce *every* continuous feature with the correct patterns of that continuous feature in the real data. When training a logistic classifier in which even a single continuous feature is easily distinguishable, a logistic classifier can find a hyperplane/decision boundary that can perfectly separate the data using that single unrealistic feature. Thus, while non-HAR-CTGANs may be able to generate some features fairly realistically, they lack the ability to generate data that is holistically realistic.

VII. Discussion

A. Effect of Synthetic Corpus Size

When upscaling real datasets with synthetic data, the ultimate goal is to fill in the "gaps" in real data with realistic-looking data in order to maintain a diverse, balanced dataset for downstream applications. As larger and larger volumes of data are needed for deep HAR models that have larger and more sophisticated architectures, there is even more of a push for truly big-data in the HAR domain. Thus, if we have only a limited quantity of real, clean data instances, yet can build generative models that can provide near limitless volumes of synthetic data that is near indistinguishable from real data, we can arrive at the question: *do we truly need to train our downstream models on real data at all?*

To investigate whether synthetic data can completely substitute real data in practical settings and whether real data can be used *indirectly* as a means to train good generative models for HAR, we test the limits of HAR-CTGAN. In order to explore this idea for the future of HAR, we train HAR-CTGAN for a variety of epochs, then train a deep HAR classifier using exclusively synthetic samples of different orders of magnitude. Ideally, there should exist a large enough sample size of synthetic data produced from a sufficiently good generator

that should yield a downstream performance that is at least equivalent, if not superior, to using real data.

Thus, if HAR-CTGAN can successfully generate data that completely spans the space that realistic mobile sensor data lives on, then as the size of the synthetic dataset approaches infinitely many instances, then the dataset should approach complete and more expansive coverage of the realistic space than the original real dataset

Table VI: The size of synthetic sampling greatly affects downstream classification performance.

Synthetic Volume	F1 Performance By Epochs Trained		
	100	500	1000
$1 \cdot 10^3$	0.706 ($\pm$ 0.022)	0.768 ($\pm$ 0.016)	0.795 ($\pm$ 0.005)
$5 \cdot 10^3$	0.691 ($\pm$ 0.014)	0.778 ($\pm$ 0.012)	0.797 ($\pm$ 0.005)
$1 \cdot 10^4$	0.639 ($\pm$ 0.031)	0.771 ($\pm$ 0.013)	0.791 ($\pm$ 0.008)
$5 \cdot 10^4$	0.705 ($\pm$ 0.008)	0.768 ($\pm$ 0.014)	0.783 ($\pm$ 0.006)
$1 \cdot 10^5$	0.747 ($\pm$ 0.008)	0.790 ($\pm$ 0.013)	0.810 ($\pm$ 0.004)
$5 \cdot 10^5$	0.761 ($\pm$ 0.016)	0.815 ($\pm$ 0.011)	0.833 ($\pm$ 0.007)
$1 \cdot 10^6$	0.765 ($\pm$ 0.013)	0.819 ($\pm$ 0.014)	0.837 ($\pm$ 0.007)

Table VI reflects the mean and standard deviation of performances of a deep HAR classifier when trained on completely synthetic data from a variety of HAR-CTGANs trained for 3 different training lengths. Generally, there is an increasing improvement in HAR classifier performance, nearly reaching and surpassing the same classifier when trained on exclusively real data or upsampled real data up to the largest class size. However, we see that within a single duration of the training, increasing the sample size of synthetic data only has a steady increase in improving model performance. Also, unsurprisingly, as HAR-CTGAN trained longer, the synthetic data from well-trained models produced the best model performance for each sample size for downstream HAR training. This implies that while generation may be realistic, fully substituting real datasets with synthetic data may be possible with enough computational resources.

It is important to recognize that even when ignoring the several-fold longer time requirement to train HAR-CTGAN than state-of-the-art GANs, training a deep HAR classifier on an enormously large synthetic sample in order to get negligibly different performance than training on real data is not a data-efficient task. If synthetic datasets that are orders of magnitude larger than real datasets can produce equivalent results, then there is a case to be made that real data is more meaningful per instance than each instance of synthetic data.

B. 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 [28], [39].

C. Future Work

Generative adversarial paradigms require large corpora in order to effectively have their generator machine synthesize realistic data. With more and more limited examples in the 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.

VIII. 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 demonstrate on a conceptual and empirical level that discrete features in HAR data add significant impact to HAR classification tasks. 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 mobile sensor data, a task never before achieved in the HAR domain. 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 on the continuous and discrete aspects of novel data generation.

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 down-stream classification models with the best features, whether they are continuous or discrete, for passive healthcare monitoring via mobile devices.

Acknowledgements. This work was funded by the DARPA WASH program HR001117S0032 and the Dean of Arts and Science Summer Research Grant. Results in this paper were obtained in part using a high-performance computing system acquired through NSF MRI grant DMS-1337943 to WPI.

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

[4] 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.

[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

[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* care," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1843–1850, 2011.
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. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

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

[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] 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.