Synthetic Generation of Data for HCR using Image-to-Image Translation GANs

by

Ronak Sankaranarayanan

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Master of Science
in
Data Science
by

____

May 2022

APPROVED:

____

Professor Emmanuel O. Agu, Thesis Advisor

____

Professor Mohammad Y. Eltabakh, Thesis Reader

____

Professor Elke A. Rundensteiner, Program Head

ABSTRACT

Human Context Recognition (HCR) is an important task in Context-Aware (CA), ubiquitous computing systems that often utilize Machine Learning. HCR involves recognizing the user's context to adapt a context-aware application's behavior. HCR on smartphones faces an main challenge when data is collected in-the-wild: Imbalanced data as subjects do not visit all contexts equally. This problem cause significant reductions in the performance of HCR machine learning classifiers. To solve this problem, various methods of data augmentation and synthetic data generation have been proposed in prior work. Image-to-image translation Generative Adversarial Networks (GANs) convert images from one domain to another, and prior work has found them effective for augmenting smartphone human activity data, but they have not been explored for HCR data. In this thesis, we systematically studied, rigorously evaluated and compared three state-of-the-art Image-to-Image translation GANs for the task of generating synthetic smartphone HCR data to augment real HCR data to improve HCR performance. Specific state-of-the-art image-to-image translation GANs we explored include StarGAN V2, Gaussian Mixture Model Unsupervised Image-to-Image Translation (GMM-UNIT) and Domain Specific GAN (DOS-GAN). Various quantitative evaluation metrics were employed, including FID score and KL Divergence to evaluate the quality of the data generated. Along with the quantitative measures, the performance of a deep learning classifier trained on synthetic data generated by different GAN models were compared. All three models have been able to create quality and diverse data. The HCR dataset generated using GMM-Unit has achieved the lowest KL Divergence and FID Score of 4.11 and 21.91 respectively. An HCR classifier trained on the synthetic data generated by the GMM-Unit slightly outperformed one trained on real HCR data with minor improvements of 0.72% and 0.3 in accuracy and F1 score respectively.

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my thesis advisor, Prof. Emmanuel O. Agu, for his guidance and support of my research. His advice and feedback helped me shape my research and writing of this thesis in the right direction. I would like to mention the significant contribution from my fellow colleagues from Dr Agu's, especially Apiwat Ditthapron and Wen Ge, who helped me in various parts of my research, leveraging their prior experience and knowledge in this research domain. I would like to thank Prof. Mohamed Y. Eltabakh for being the reader of my thesis reader and attending my thesis presentation.

I would like to thank my family and friends for their support and encouragement throughout my years of studying.

NOMENCLATURE

HCR	Human Context Recognition
GAN	Generative Adversarial Network
HAR	Human Activity Recognition
CA	Context Aware
ADL	Activities of Daily Living
SMOTE	Synthetic Minority Over Sampling Technique
ADASYN	Adaptive Synthetic
VAE	Variational Auto-Encoder
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
DOSGAN	Domain Specific GAN
GMM-UNIT	Gaussian Mixture Model - Unsupervised Image-to-Image Translation
KL	Kullback-Leibler
FID	Fréchet inception distance

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Context Aware Applications in Ubiquitous Computing

Context Aware (CA) applications have been proposed for addressing important problems in various diverse fields, including for understanding the user's health condition and daily fitness levels in the medical field. By monitoring patients' Activities of Daily Living (ADLs) and context visit patterns, doctors can also monitor the recovery of patients from various ailments.

1.2 Human Context Recognition

Human Context Recognition (HCR), which involves recognizing the current context or situation of a user, is an important task for Context Aware (CA) applications. The recognition of the user's context (or current situation), which frequently integrates additional information such as phone placement, user's location and social situation, provides richer information but is also more challenging than Human Activity Recognition (HAR). Specifically, the smartphone sensor signature corresponding to running with the phone placed in the coat pocket differs significantly from the sensor signature for running with the phone held in the user's hand. Modern HCR approaches often utilize machine or deep learning models for optimal performance. To gather realistic datasets, emerging data gathering studies often gather HCR datasets in the real world with the user providing labels periodically to annotate the contexts they visit.

1.3 Importance of HCR

Due to the near pervasive ownership and usage of smartphones and smartwatches, a wide variety of diverse Human Context Recognition applications have been explored in recent years. Such mobile devices come equipped with a broad range of sensors from which data can be collected, including an accelerometer, and gyroscope that can all be utilized to enhance HCR. Applications have been explored in broad domains including personal fitness and medical monitoring. Different machine learning models that have been trained on pre-recorded datasets are executed in the watch and smart phone to detect the activity. For personal fitness, users' body stats will be calculated by identifying the activity the user performs and helps the user maintain their fitness with little external resources. The advancement of smart watches has created multiple new monitoring systems for patients for doctors to monitor the user's activities in patients' day to day life and suggest a treatment that is effective for the patient. The models must predict the highest performance, since any mislabeling can lead to inadequate or harmful treatment for the patients, which can be dangerous.

1.4 Challenges in HCR

HCR data is very personal information about a user and thus collected only through volunteer work such as extrasensory or mhealth. There are multiple privacy concerns about collecting the data from the users directly and thus, it creates a scarcity of data for HCR. The dataset that has been collected consists of volunteer data collection from a group of people who label different activities every day, ranging from sleeping, walking etc. Among the collected data from volunteers, HCR on smartphones faces two main challenges when data is collected in-the-wild, which ultimately reduces model performance if not addressed: 1) missing or wrong data labels and 2) Imbalanced data as subjects do not visit all contexts equally. Data imbalance is a widely studied problem in supervised machine learning, which reduces the performance of supervised machine learning models and is encountered in almost all types of data, from text-based data to tabular format. One common example in which data imbalance can be dangerous is the medical field in which the occurrence of the positive class (ill people) is often low. Specifically for HCR, ill people may perform certain activities or visit certain contexts at a lower frequency, leading to severe data imbalance. For instance, a sick person may have lots of occurrences in the "lying down" context with very few examples of the "running" context. Models trained on such imbalanced data would tend to predict that the subject is lying down even when they do run. This in turn would lead to incorrect diagnoses and recommended treatments that could be dangerous to the patient. In most HCR datasets, while activities such as walking, standing, sleeping occur frequently, only very few users contribute examples of more intense activities such as rowing.

1.5 Prior work on mitigating data imbalance in HCR datasets

Initial techniques employed to overcome the data imbalance issue was performing minor transformations in the existing data to create new data. A few common techniques used are jittering, time wrapping etc. Classical machine learning models use oversampling techniques such as SMOTE [1] but the model assumes that there is no overlapping of data between classes and performs poorly in high dimensional data. SenseGen [2], SensoryGAN [3] use separate GAN models to generate data for each activity. But the resource requirement to implement multiple models for a single task is expensive and leads to a common problem of mode collapse. Different models have been implemented [4] by replacing the BCE loss function with Wasserstien loss function, which improves the stability of the model and correlates with the quality of generated data. But the time complexity of calculating Wasserstien loss function is high and leads to more consumption of resources.

1.6 Proposed methodology

Generative Adversarial Network [5], an advanced deep learning architecture, has proved effective for the task of Data Augmentation in Images. Primarily implemented for Images, different GAN models have been modified to accept sensor-based data and have proven effective in improving the performance of deep learning models for HAR and HCE. SensoryGAN [3], SenseGen [2], Activity GAN [6] are examples of GAN models implemented for sensor based data and have improved performance of deep learning classifier by at least 10%. Image-to-Image translation architecture aims at converting images of one domain to another by using the joint distribution of images in different domains based on the shared latent space distribution assumption and overcomes the issue of Mode collapse. Image-to-Image architecture has created quality and diverse images for translating synthetic images from SYNTHIA [7] to real images from Cityscape dataset [8] and vice-versa. But Image-to-Image architecture has hardly been used for Data Augmentation of Sensor data or time-series data. In the proposed architecture, the Image-to-Image architecture has been implemented and tested for the Data augmentation of the Sensor-based data.

Challenges:

1. Image-to-Image GAN architectures have primarily been developed for Image translation. Hence, there is limited prior knowledge of approaches for repurposing them for sensor-to-sensor translation that was necessary for HCR data augmentation.

2. A typical HCR dataset has many contexts visited, which presents a challenge to the image-to-image translation GAN. For instance, the ExtraSensory HCR dataset had 51 context labels present.

1.7 Thesis overview

In this master's thesis, a systematic evaluation and comparative study of utilizing various state-of-the-art Image-to-Image translation GANs for HCR data augmentation was done. Specifically, the GANs were used to generate synthetic HCR data corresponding to the minority class by translating real HCR data instances represented as an image from the majority class. This study explored various sub-types of Image-to-Image translation techniques including Directional Translation, Bi-Directional Translation, Autoencoder- based models, as well as Disentangler Representations. A representative set of image-to-image techniques were implemented and their effectiveness in balancing the dataset and hence improving HCR performance were evaluated. In our experimental pipeline, specific state-of-the-art image-to-image translation GANs that we explored included StarGAN V2, Gaussian Mixture Model Unsupervised Image-to-Image Translation (GMM-UNIT) and Domain Specific GAN (DOS-GAN). Evaluation metrics included Kullback–Leibler (KL) divergence and the Fréchet Inception Distance (FID) for evaluating the quality and diversity of the data. An external deep learning HCR classifier was trained on real and multiple synthetic data created by the Image-to-Image networks. The performance metrics including precision, recall, accuracy and F1 score of the classifier on unseen data were recorded and compared for each model as a evaluation metrics.

1.8 Thesis research questions

To define the scope of our research, we formulate primary research questions as follows.

1. *What is the performance of Image-to-Image models for Sensor based time series data on the task of data augmentation??*
2. *Which of the labels, the models can create quality data for and the reason behind it??*
3. *How well the performance of the classifier model has improved when trained on synthetic data??*

1.9 Novelty of this thesis in relation to prior work

The techniques that have been primarily used for augmentation of HCR data are somewhat dated and inefficient compared to state-of-the-art deep learning approaches. The research area in the field has come till usage of GANs and Conditional GAN for augmentation on HCR. Usage of highly advanced deep learning architecture such as Image-to-Image architecture for the process of data augmentation on HCR data or similar time series data is a research area that has been less explored. The thesis work focuses on understanding the impact of three different highly advanced architecture and the difficulty that arise due to it

1.10 Thesis contributions

The contribution of this work is as follows.

1. Generation of quality and diverse data for improving the performance of HCR models
2. Understanding the impact of Image-to-Image on HCR data also the difficulty
3. Opening of various research opportunities for exploration beyond this comparison study.

2. BACKGROUND AND RELATED WORK

2.1 Data Issues

Multiple recorded datasets of HCR have a basic issue of data imbalance due to the imbalance of performance or participation in activity and other context labels by the users. The Extrasensory dataset [9] is a common dataset that has been used for multiple HCR models. The Extrasensory dataset faces a major issue of data imbalance with, few labels such as sitting representing 40% of the entire dataset, while context labels such as running only represent 0.3% of the dataset. The difference between the highest occurring label and the lowest occurring label is 180,000 data points. The LIG Smart Phone Human Activity Dataset (LIG SPHAD) [10] is another common HCR dataset that faces the same issue of data imbalance, with the most common activity label Sitting represents 26.4% of the corpus, whereas the least common activity is Jumping represents merely 1.9% of the corpus.

2.2 Data Augmentation

Data Augmentation is the process of synthesizing multiple data points from the existing data available to overcome data issues mentioned above. Earlier models used methods of making minor changes to the original data in creating new data. For a spatial dataset such as images, minor transformation like shift, flip, rotation brightness and zoom can be performed to create more data for the same domain. Data Augmentation improves the machine learning model process by providing enough data and the diversity required for the model.

2.2.1 Sampling methodology for Data Augmentation

Synthetic Minority Over Sampling Technique (SMOTE) [1] is a machine learning technique that focuses on oversampling of tabular data. It selects k nearest data points to each minority class data point and oversamples data between them and assigns them as the minority data label. The basic idea behind is that closer objects to a data point should belong to the same class and data between them should also be of the same class. First put by Nitest et el and later boarderline

SMOTE [11]has been proposed by Hui which focus on the data points that are present on the boarders of the minority class distribution rather than picking each data from the minority class. Adaptive Synthetic (ADASYN) [12] implements a methodology that detects those samples of the minority class found in spaces dominated by the majority class to generate more data with less density. This is to create more samples of the data that are difficult to classify. The oversampling techniques assume that there is no overlapping between data of different classes, which is highly impossible in real world applications and performs relatively poorly on high dimensional data. Nyugen et el has proposed an extension of SMOTE for oversampling of data called Border Limited Link SMOTE (BLL SMOTE) [13] which is specifically designed for oversampling Human Activity Recognition and tackling the issue of non-convex space. The BLL SMOTE samples data closest to higher density of data points rather than random sampling, this eliminates the probability of the data being generated near the border of clusters or overlap of data distribution. BLL SMOTE has improved the MLP performance metric F1 score from 68% to 80%.

2.2.2 Deep Learning methodology for Data Augmentation

Recent developments in Deep Learning have opened new methodologies for synthesizing data of different data types with better quality and diversity. The Variational Auto-Encoders (VAE) [14] s a generative model that uses an encoder-decoder duo, in which the encoder works on converting an image to a feature space and the decoder works on converting any random vector on that feature space into an image. Generative Adversarial Networks (GAN) [5] is another deep learning architecture that uses generator and discriminator. The Generator converts a random vector into an image and the discriminator acts as a classifier identifying the real image from fake.

2.3 Generative Adversarial Networks

This thesis explores Generative Adversarial Networks (GANs) as a method for synthetic data augmentation. Generative Adversarial Network [5] uses a Generator-Discriminator architecture which focuses on Min-Max loss value optimization for training. The generator creates images from a random vector of a feature space and then feeds them to the discriminator which tries to identify the real and fake images. The discriminator tries to minimize the loss value by identifying all the fake images, whereas the generator updates the weights to maximize the loss by creating more realistic images. Although GAN models create quality images, a major limitation of GAN is the lack of diversity among different domains and GAN models cannot generate the images of a specific domain needed. Thus, for multi domain image augmentation, multiple models must be trained to generate data of different image domain. Conditional GAN [15] is an extension of GAN that solves the problem of the need for multi model training by taking the domain information along with the random vector as the input and generating images belonging to a specific domain. After the generator creates synthetic data based on the domain information provided, the discriminator uses the images and the domain information as input and performs two classification tasks. First, a binary classification of whether the given image is real or fake and second, a multi-class classification to identify the domain the image belongs to. The loss value calculated is used to train both the generator and the discriminator. This overcomes the need for multi-model training but leads to a common issue called mode-collapse in which the model generates similar data for two or more domains, which decreases the diversity of the data among different domain generators. Wassertein GAN [16] implements CGAN but replacing BCE loss function of the discriminator with Wassertein loss function to ensure better stability of the model. But Wassertein loss function has a huge resource requirement and is not commonly used.

2.4 GANs for Generating Sensor Time Series Data

The HCR datasets that the GANs we explored used as input contained smartphone sensor data (e.g., accelerometer values), which were essentially time-series data. Wang, Jiwei et al proposes SensoryGAN, a GAN that consists of an LSTM and convolutional layers [3].SensoryGAN creates different GAN models to generate sensor time series data for 3 different activities using LSTM 1D convolutional layers, Bidirectional LSTM, and fully connected layers. The resource requirement to create and generate different GANs for each different context is computationally expensive. The model implemented used a single GAN architecture but multiple instances of the model which is more expensive. Their implemented model achieved improved performance (accuracy) with the hybrid data model that combines real and synthetic data outperforms the model that utilized only real or synthetic data by almost 10% in classification experiments using 6 different classification algorithms. A similar GAN based model has been proposed by Alzantoto named SenseGen (Alzantot et al. 2017 [2]) which uses LSTM layers and Mixed Density Network in the generator and LSTM layers in the discriminator. One significant difference in SenseGen is the method of training Generator and Discriminator separately. All models utilized in our study differ from these models by training a single model to transform a single data instance into another.

Norgaard et al proposed a supervised GAN model for generating HAR synthetic sensor data (Norgaard et al. 2018 [4]). Synthetic sensor data implements a WGAN [16] which is a normal GAN but using a Wasserstien loss function rather than the usual BCE which could potentially lead to mode collapse. The research implements two different types of GAN for static and dynamic activities implemented using LSTM, CNN, and Dense layers. The synthetic data has been evaluated using two different classifiers consisting of 1D CNN and LSTM and how they are performed when trained on real, hybrid and synthetic data. The F1 score on the test data of the classifier models of 1D CNN and LSTM model is used as an evaluation metrics which achieves a significant improvement of 7%. While the model has better performance, the Wasserstien loss function utilized is expensive in terms of computational resources. Consequently, the proposed models explore several loss functions such as self-reconstruction, content reconstruction, attribute reconstruction etc. for training purposes.

2.5 Image to Image Translation GANs

This thesis explored repurposing image-to-image translation GANs, which were initially proposed for translating images, for the task of HCR sensor-to-sensor data translation. The GAN models [5][15][16] have previously seen different issues, such as multi-model training, mode-collapse, high resource requirement. These issues have reduced the performance of the model and data augmentation process by creating poor quality or less diverse images. Image-to-Image translation architecture using GANs which was first introduced by Isola, Philip et al [17] overcomes these issues by using a CGAN which takes the image of one domain as an input along with a sample image from the target domain as another input. The initial experimentation was performed on different paired datasets such as edges to shoes [18], edges to handbag [19]. Later advancements in architecture have led to data augmentation on unpaired datasets as well such as person to person translation [20], translation between facial features [21], season-to-season translation [22]. Aziz et al[23] did a comparison study of different Image-to-Image Translation techniques and categorizes them according to their training methodology. The paper divides proposed models into supervised translation and unsupervised translation along with the subdivisions under both. The survey outlines the techniques of each model with Input, Output, Training method, loss functions used. The study helps in understanding the difference between different sub sections and similarity of the models within the same sub section. The survey also gives examples of different applications of image-to-image translation with real world examples. Yongxin [24] also published a survey presenting different Image-to-Image translation models and their training methodologies.

3. METHODOLOGY

3.1 HCR Dataset

In our experiments, we utilized the ExtraSensory HCR dataset [9], which is a state-of-the-art smartphone HCR dataset gathered at the University of California at San Diego (UCSD). Data was gathered in the wild as subjects lived their lives. Subjects were prompted periodically to provide labels of the contexts they visited. The Dataset contains 51 distinct HCR labels representing different activities and phone placements collected from 60 different users. Data was collected from various smartphones and smartwatch sensors. The data was collected from a diverse set of individuals. Each sensor collects data at a different sampling rate and each minute is recorded and labeled in the dataset. Our work will only utilize data from the accelorometer, gyroscrope and audio. Accelerometer data collects the Tri-axial magnitude and direction of acceleration of the device and Gyroscope records the rate of rotation around the device 3 axes for a time window of 20 secs. Audio data represents the 13 Mel-Frequency Cepstrum Coefficients values that collectively represent the Mel-Frequency Cepstrum of the audio data collected from the device during the 20 secs time window. Both Accelerometer and Gyroscope have recorded the data at the rate of 40Hz for 20 secs whereas Audio data has been collected at the rate of 22Hz for 20 secs.

	Range	Average (Standard Deviation)
Age (years)	18-42	24.7 (5.6)
Height (cm)	145-188	171 (9)
Weight (kg)	50-93	66 (11)
Body mass index (kg/m^2)	18-32	23 (3)
Days of participation	2.9-28.1	7.6 (3.2)

Table 3.1: Demographic of UCSD Extrasensory data collection participants

Figure 3.1 displays randomly sampled accelerometer data from all three axis X, Y and Z axes from the Extrasensory dataset along for 800 time steps.

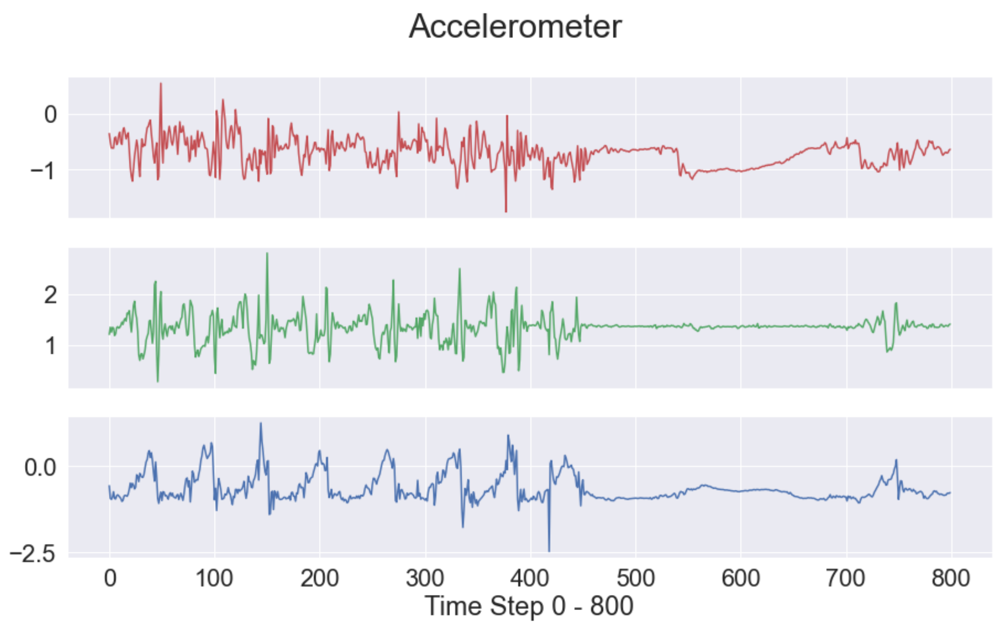

Figure 3.1: Accelerometer sample from ExtraSensory Dataset

Figure 3.2 displays the Gyroscrope data from a random sample data from the Extrasensory dataset along all three axis X, Y and Z for 800 time steps.

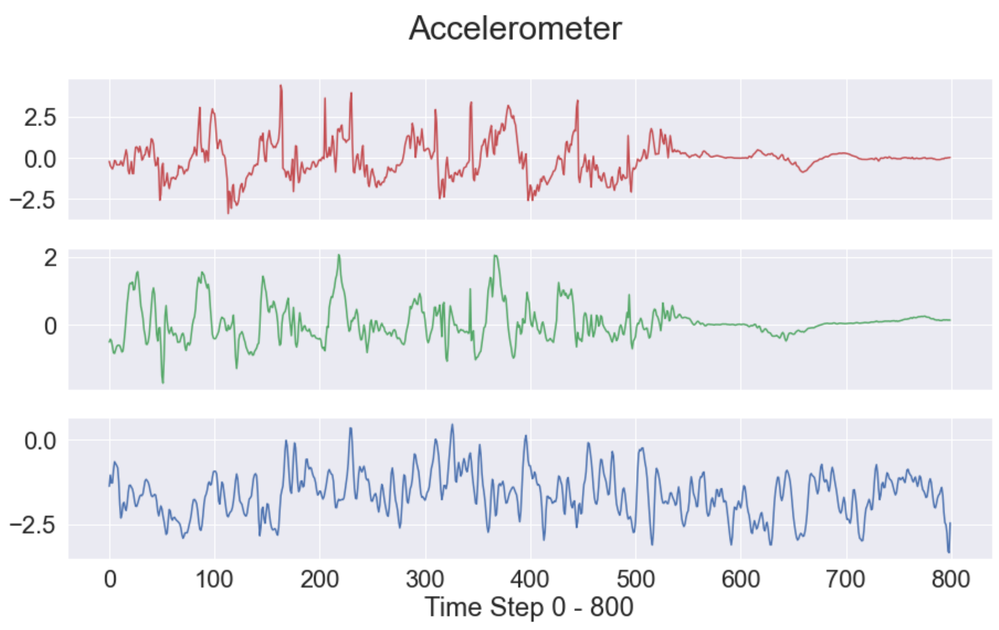

Figure 3.2: Gyroscope sample from ExtraSensory Dataset

3.2 Data Imbalance in HCR

Since the data is collected from multiple users voluntarily as they lived their lives, all participants did not visit all contexts equally. Consequently, there exists a huge difference between number of data points available for various labels and other. Figure 3.3 represents the data imbalance across 10 labels that had the highest number of data points, compared to 10 labels with the lowest number of data points. The Highest number of data points consists over 150,000 data points in each label whereas the lowest data points consist of less than 1000 data points.

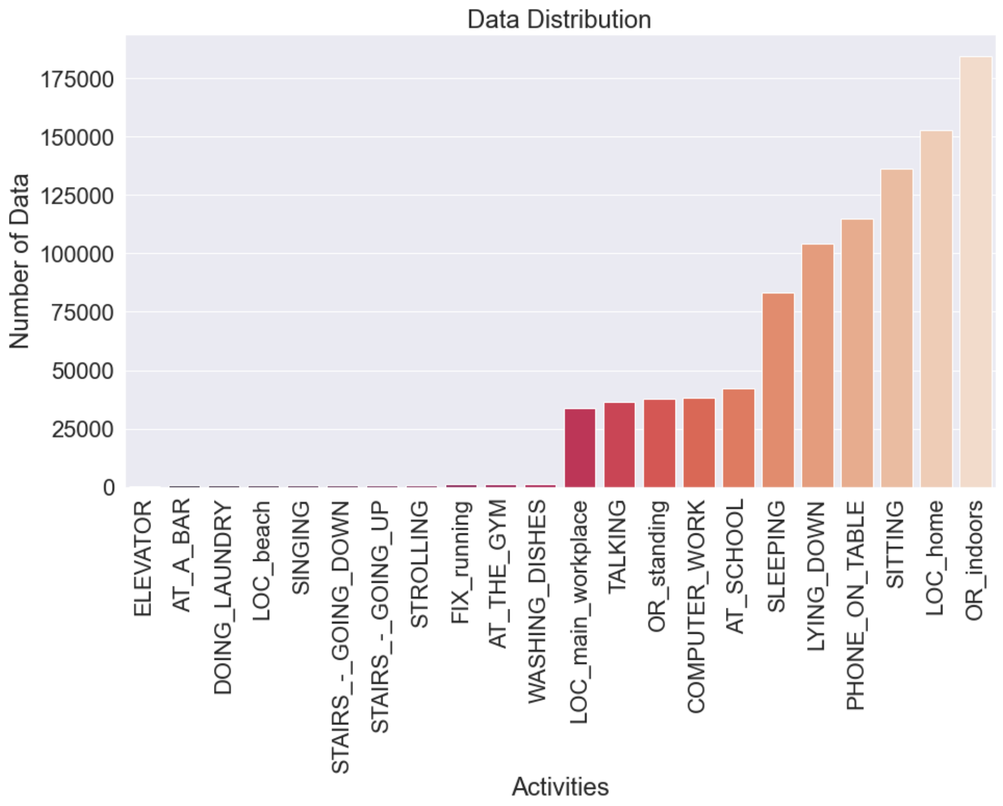

Figure 3.3: Imbalance in the number of data points gathered for various contexts in the Extrasensory HCR dataset

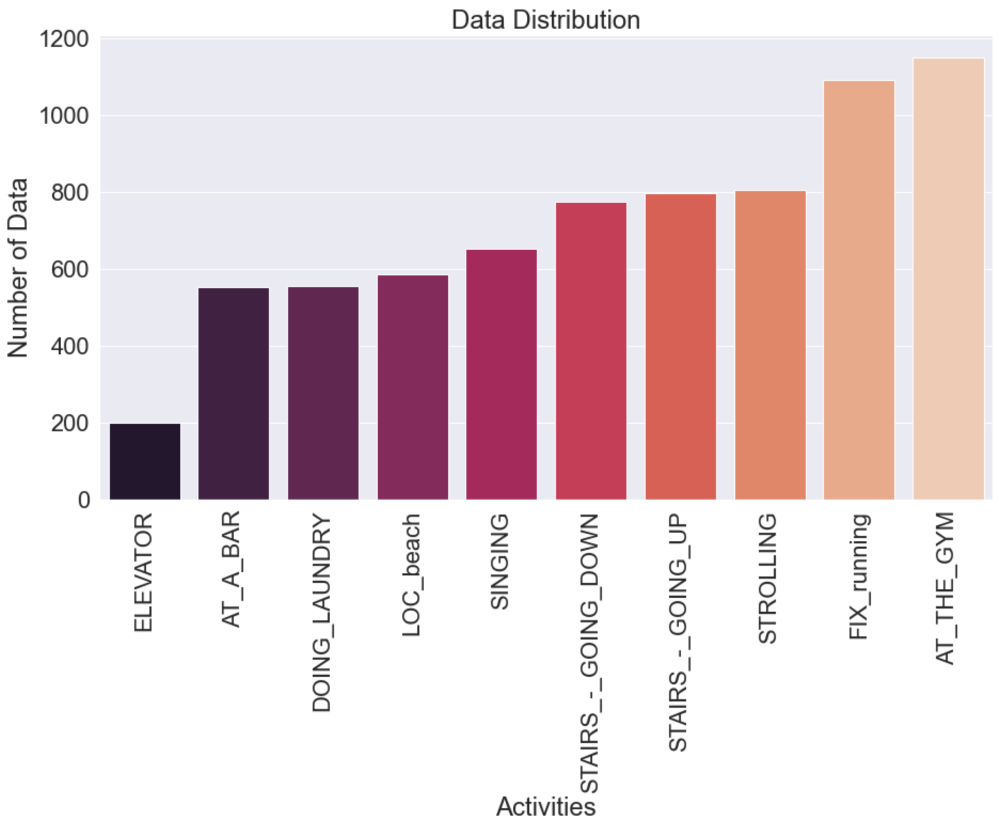

Figure 3.4: Data Imbalance

Figure 3.4 represents the number of data points collected for the least frequently visited contexts and Figure 3.5 illustrates imbalance in the number of users visiting various contexts. The most frequently visited contexts were visited by over 50 of the 60 participants, while the least frequently visited contexts were visited by less than 10 of the 60 participants.

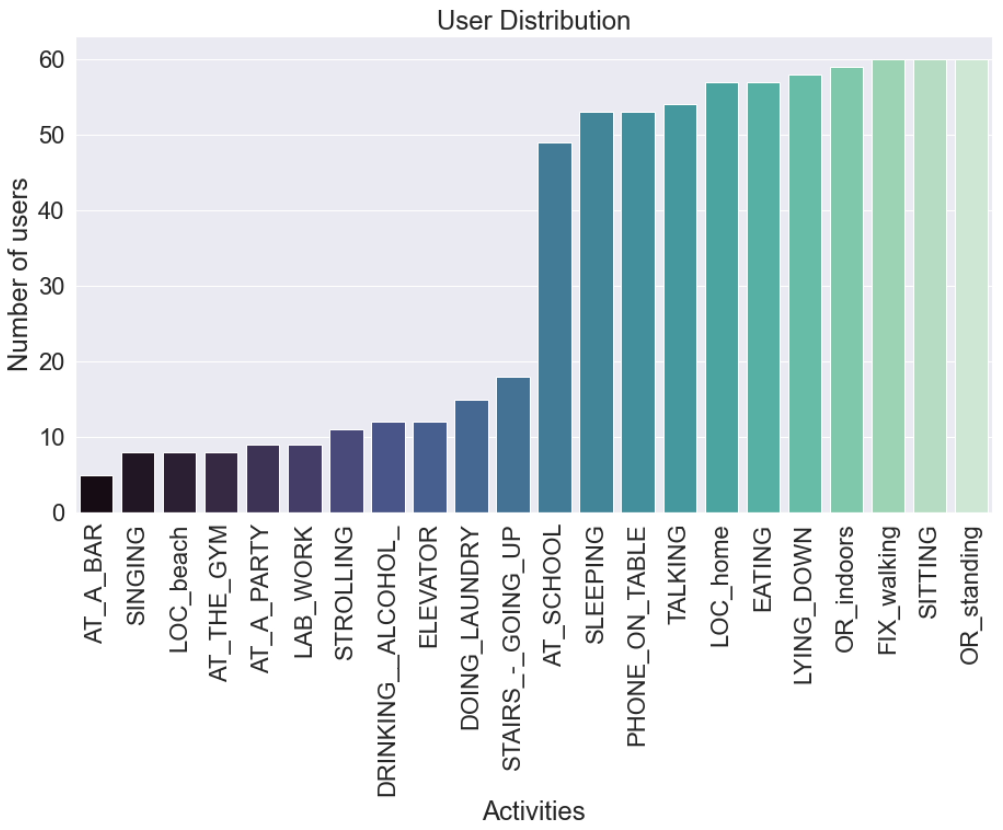

Figure 3.5: Showing imbalance in the activity labels provided by various users

3.3 Deep Learning Pipeline

Figure 3.6 represents the basic training pipeline steps performed for each generative model and its different hyper parameter sets. First the Accelerometer, Gyroscope and audio data are extracted, combined, and preprocessed into a single datapoint before being split into training and validation data. The training data is given as input for training the generative model from which each data will be used as source and target data for the generative model to translate between each other. After the model has been trained according to the training methodology of each model, the validation data is used as input from which each data point is used as source data and randomly chosen datapoint is used as target data from the same set. This validation data is used for synthetic data generation to create perfectly balanced data.

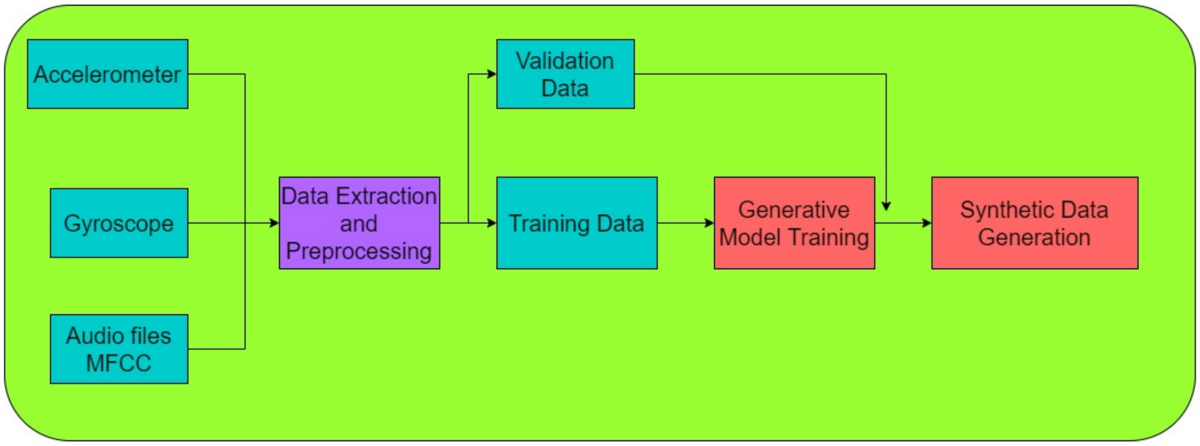
Accelerometer
Gyroscope
Audio files MFCC
Data Extraction and Preprocessing
Validation Data
Training Data
Generative Model Training
Synthetic Data Generation

Figure 3.6: GAN-based Data Augmentation Training Pipeline architecture

Figure 3.7 represents the evaluation pipeline for the Data Augmentation process. After the synthetic data generation, the validation data is used as Real data reference for the evaluation process. the first evaluation methodology is Quantitative evaluation in which random samples from Real and Synthetic data are taken for calculating evaluation metrics between them to evaluate how realistic the synthetic data is. After quantitative evaluation, both synthetic and real data is passed as training data for an HCR classifier model training. After training of the models, a subset of the real data is used as testing data to evaluate the performance metrics of the classification task for the models trained. The recorded metrics will evaluate the impact of the synthetic data in the performance of the model.

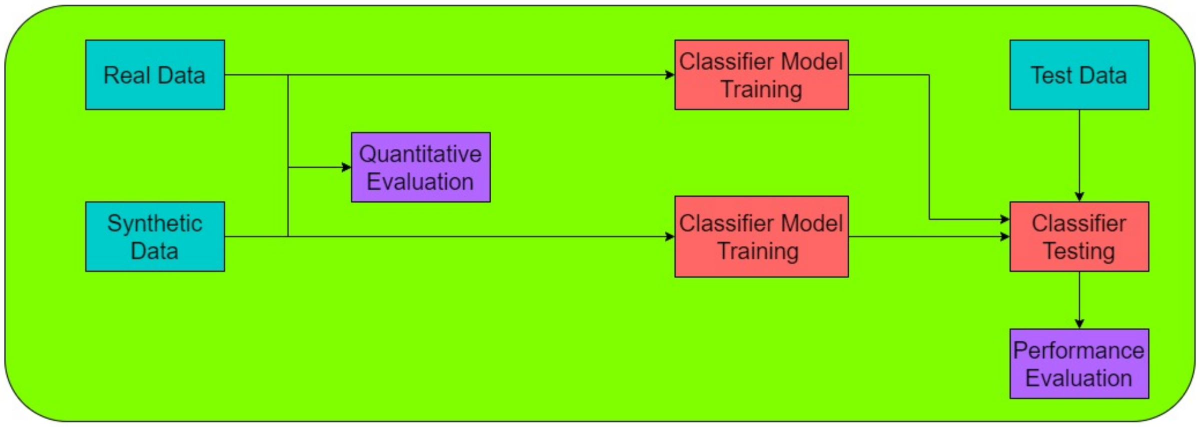

Figure 3.7: GAN-based Data Augmentation Testing Pipeline architecture

3.4 Data Extraction and Preprocessing

The data collected from Accelerometer, Gyroscope and Audio MFCC values which are stored in different files are extracted and combined to represent the same timestep as a single data point. The data is resampled to the same frequency, standardized, and sliding window methodology has been performed to create time series data of specified window size and step value. The file structure of the extracted dataset follows a hierarchical storage structure with each label represented as a sub folder under which all the data points for the label are stored. Each subfolder represents a context domain with multiple data under it.

3.5 Data Augmentation models

As training HCR were unpaired, unsupervised GAN models were utilized for image-to-image translation. From the taxonomy of GANs presented in Figure 1, recent GAN models in the Cyclic Constituency, Disentangle representation and auto encoder-based categories were select for evaluation. These selected GAN models have outperformed prior models by overcoming their other shortcomings in various ways.

3.5.1 DOS-GAN Sensor

The first model for the comparison study is Domain Specific GAN (DOS-GAN) [25], which uses a pre-train classifier neural network model as domain specific-feature extractor $\alpha(.)$, an encoder for domain-independent feature extractor $\beta(.)$, a generator *g* and a discriminator *d*. The domain specific feature extractor $\alpha(.)$ is achieved by training a neural network model on the complete training data in a supervised fashion. After the training is complete, the output of the second to last layer is able to extract all the domain specific information needed to distinguish the data between various domains. The pre-trained classifier training is stopped after the pre-training stage and no update takes place for the training of the GAN. A discriminator is used to classify the synthetic data as either real or fake and which domain the data belongs to. In training, the domain-specific features of the target domain are extracted using a pre-trained classifier $\alpha(x_{target})$ and the domain-independent features of the source domain are extracted using an encoder $\beta(x_{source})$. Both sets of features are appended ($\oplus$) and input to the generator which produces the data for translating from source to domain. The newly augmented data is input to the discriminator along with the original data to classify the domains and to identify the real and fake data. The complete architecture of the DOSGAN is outlined in Figure 3.8.

$$\begin{aligned} S_B &= \alpha(x_B) \\ S_A &= \beta(x_A) \\ x_{AB} &= S_A \oplus S_B \end{aligned} \tag{3.1}$$

Various loss values are calculated to update the weights of multiple versions of the DOS-GAN model as follows

Classification loss: The multi-class cross entropy loss was used for training the domain-specific extractor $\alpha(.)$. The loss function calculates the number of labels the classifier has predicted

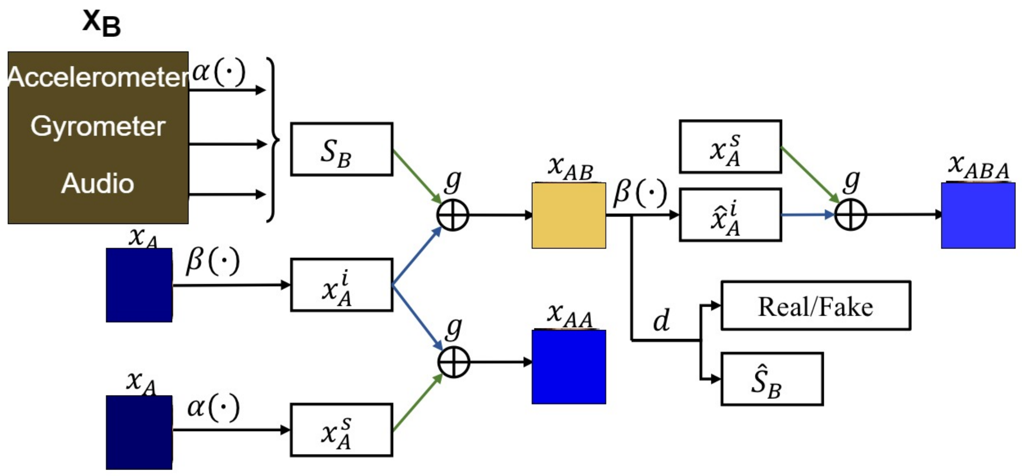

Figure 3.8: DOSGAN [25] Image-to-image translation architecture

correctly and updates the weights of the classifier.

$$\ell_{ce} = -\sum p(x).log(1 - p(x)) \tag{3.2}$$

Adversarial loss: The adversarial loss function is based on a GAN in which the discriminator takes the input of real and fake data and outputs a probability of how likely the data belongs to a real domain. A log loss of binary classification is used to represent the Adversarial loss, which the discriminator tries to maximize it and the generator tries to minimize.

$$\ell_{GAN} = \frac{1}{|D_A|}\sum log(d_{adv}(X_A)) + log(1 - d_{adv}(X_{AB})) \tag{3.3}$$

Domain-specific reconstruction loss: This loss uses the features extracted from the domain using $\alpha(.)$ and discriminator to compare the features. This loss value is used to ensure the consistency of the augmentation. $\ell_{\alpha,r}$ compares the features extracted from the source image by the discriminator and the domain-specific feature extractor to train the discriminator and $\ell_{\alpha,f}$ compares the features extracted from the translated image and target image to maintain the consistency of the translation and used to train the generator.

$$\begin{aligned}\ell_{\alpha,r} &= \frac{1}{|D_A|}\sum[||d_{feat}(x_A) - x_A^s||_1] \\ \ell_{\alpha,f} &= \frac{1}{|D_A|}\sum[||d_{feat}(x_{AB}) - S_B||_1]\end{aligned} \tag{3.4}$$

Image reconstruction loss:

(1) self-reconstruction loss evaluates the L1 norm between x_A and x_{AA} in which $x_AA = g(x_a^i, x_a^s)$ is a image translated to its own domain. (2) cross-domain loss is the L1 norm between x_A and x_{ABA}.

$$\ell_{im} = \frac{1}{|D_A|}\sum[||x_A - x_{AA}|| + ||x_A - x_{ABA}||_1] \tag{3.5}$$

Overall training loss:

The Generator and domain-independent feature extractor is trained by minimizing the loss function as follows:

$$\ell_{net}^{total} = \ell_{GAN} + \lambda_f \ell_{\alpha,f} + \lambda_{im}\ell_{im} \tag{3.6}$$

The Discriminator is trained by minimizing the loss function as follows :

$$\ell_{d}^{total} = -\ell_{GAN} + \lambda_f \ell_{\alpha,r} \tag{3.7}$$

3.5.2 GMM-UNIT

The next model that will be evaluated for image-to-image translation of HCR data is the Gaussian Mixture Model - Unsupervised Image-to-Image Translation (GMM-UNIT) [26] . GMM UNIT is based on content attribute disentangled representation where the attribute space is fitted with a GMM. The model disentangles the data into domain invariant content space and domain specific attribute space, assumes the domain specific attributes follow a Gaussian data distribution in feature space. The probability density of the latent space z is defined as

$$p(z) = \sum_{k=1}^{K} \phi_K N(z; \mu^k; \Sigma^k) \tag{3.8}$$

The model also does not use labels encoded as values but uses two encoders, which learn to convert the data into content and attribute features that are fed into a generator model. This assumption helps the GMM-UNIT to overcome the issue of mode collapse. The assumptions allow three key properties of mode diversity, multi-domain translation and few/zero-shot generations. By using GMM on the sensor data we will be able to evaluate how much data disentanglement is able to improve the performance of the Image-to-Image Translation on sensor data. The model uses two encoders to learn the content and attribute of the source data, these attributes are combined and provided as input to the Generator model to synthesize a target domain data which is given to a discriminator along with real data to identify real or fake as well as which domain the data belongs to.

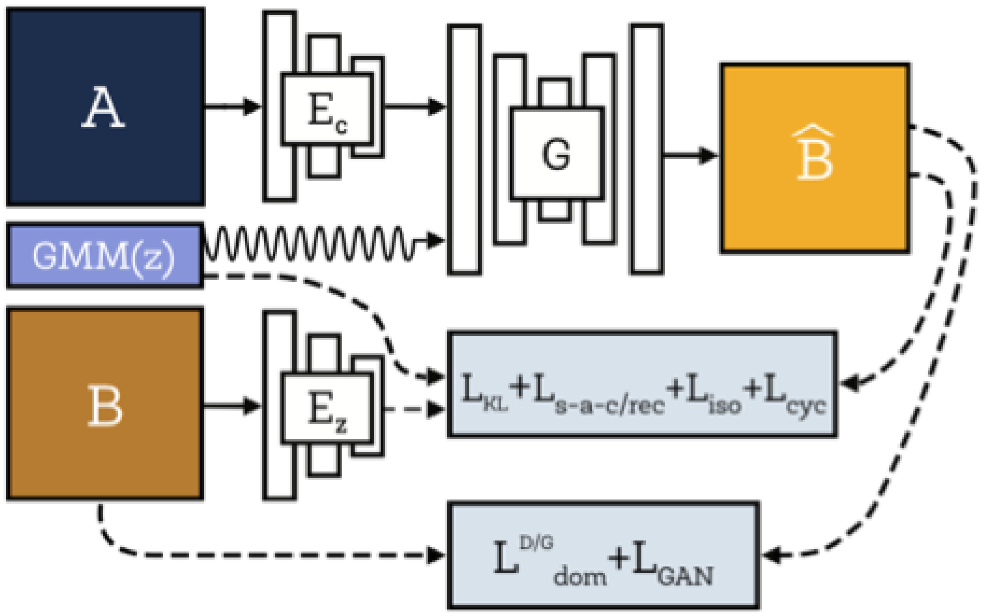

Figure 3.9: Training process based on Image-based implementation of GMM-UNIT [26]

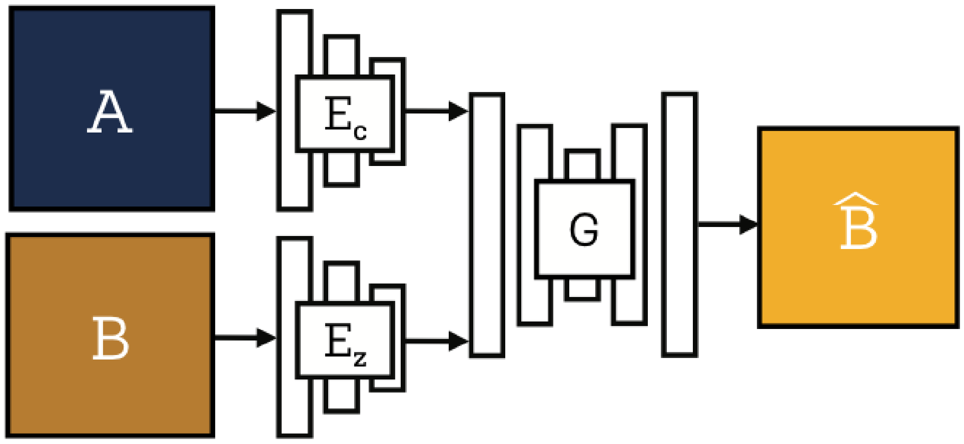

Figure 3.10: Training process based on Image-based implementation of GMM-UNIT [26]

the Losses calculated for the training of all the models are as follows

Self-reconstruction loss: The L1 norm between the original data and the reconstructed data with its own content and attributes

$$\ell_{s/rec} = \sum_{n=1}^{K} \mathbb{E}_x[||G(E_c(x), E_z(x)) - x||_1] \tag{3.9}$$

Content reconstruction loss: The L1 norm between the content features of original data and translated data. The loss is calculated to ensure all the contents of source domain has been maintained after the translation

$$\ell_{c/rec} = \sum_{n,m=1}^{K} \mathbb{E}_x[||E_c(G(E_c(x_A), E_z(x_B))) - E_c(x)||_1] \tag{3.10}$$

Attribute reconstruction loss: The L1 norm between the attributes of the target domain and the translated data. The loss is calculated to ensure the attributes are translated and maintained during the translation

$$\ell_{a/rec} = \sum_{n=1}^{K} \mathbb{E}_x[||E_z(G(E_c(x_A), E_z(x_B))) - E_z(x_B)||_1] \tag{3.11}$$

Cycle consistency loss: The L1 norm loss function to check whether the translated domain can be translated back to the original domain.

$$\ell_{cyc} = \sum_{n=1}^{K} \mathbb{E}_x[||G(E_c(x_{AB}), E_z(x_A)) - x_A||_1] \tag{3.12}$$

Domain Classification loss: The entropy loss function using the discriminator domain classification on the original data and the translated data. d_X^A and d_X^B represents the labels of domain A and B. ℓ_{dom}^D calculates the domain classification loss between the original domain data and the label and ℓ_{dom}^G calculates the domain classification loss between the generated data and the target label. The generator is trained only by ℓ_{dom}^G but the discriminator uses both for training.

$$\begin{aligned} \ell_{dom}^D &= \sum_{n=1}^{K} \mathbb{E}[-logD_{dom}(d_X^A|x_A)] \\ \ell_{dom}^G &= \sum_{n=1}^{K} \mathbb{E}[-logD_{dom}(d_X^B|G(E_c(x_A), E_z(x_B))] \end{aligned} \tag{3.13}$$

Adversarial loss: The entropy loss function using the discriminator real/fake classification on the real and translated data. The loss function is used by both Generator and Discriminator.

$$\ell_{GAN} = \sum_{n=1}^{K} \mathbb{E}[-logD_{r/f}(x)] + \mathbb{E}[-log(1 - D_{r/f}(G(E_c(x_A), E_z(x_B)))]$$

(3.14)

Total loss: ℓ_D is used to train the discriminator $D(.)$ architecture alone. ℓ_G is used to train the generator $G(.)$, content encoder $E_c(.)$ and attributed encoder $E_z(.)$

$$\begin{aligned}\ell_D &= \ell_{GAN} + \ell_{dom}^{D} + \ell_{dom}^{G} \\ \ell_G &= \ell_{GAN} + \lambda_{s/rec}.\ell_{s/rec} + \ell_{c/rec} + \ell_{a/rec} + \lambda_{cyc}.\ell_{cyc} + \lambda_{KL}.\ell_{KL} + \lambda_{iso}.\ell_{iso} + \ell_{dom}^{G}\end{aligned} \tag{3.15}$$

3.5.3 STARGAN-V2

StarGAN V2 [27] is an Image-to-Image generative model that consist of Generator G, Mapping Network $F_y(.)$, Style Encoder $E_x(.)$ and Discriminator D. The Mapping network$F_y(.)$ takes a latent code z and a domain y as input to generate style code for the domain y. The $F_y(.)$ has multiple output branches to generate diverse style code for all the domains present. The Style Encoder $E_x(.)$ takes an image x and its respective domain y as input to generate the style code for domain and the image. Like F_y, E_x also has multiple output branches to create diverse style code for all the domains present. Generator G takes an input image and a style code generated either by F_y or E_z and translate the input image to the label mentioned. The D takes the original and the fake data to classify real/fake. Rather than using domain classification, D has multiple output branch each resulting in a binary classification of real/fake for each domain.

Multiple loss values are calculated to update and optimize all the neural network architecture mentioned above.

Adversarial loss: A random latent code z and domain label y are used to generate target style code $\tilde{s}= F_{\tilde{y}}(z)$ which is used to transform the input image x into the target domain.

$$\ell_{adv} = \mathbb{E}[logD_y(x)] + \mathbb{E}[log(1 - D_{\tilde{y}}(G(x, \tilde{s})] \tag{3.16}$$

Style reconstruction: The L1 norm is calculated between the target style code and the style code of the newly generated image using the style encoder E_y

$$\ell_{sty} = \mathbb{E}[||\tilde{s} - E_{\tilde{y}}(G(x, \tilde{s})||] \tag{3.17}$$

Style diversification: to further enable the generator G to produce diverse data, STARGAN V2 regularize the generator. It generates two style code on the same domain y and try to maximize the difference between them to produce diverse quality of images.

$$\begin{aligned} \tilde{s_1} &= F_y(z_1) \\ \tilde{s_2} &= F_y(z_2) \\ \ell_{ds} &= \mathbb{E}[||G(x, \tilde{s_1}) - G(x, \tilde{s_2})||] \end{aligned} \tag{3.18}$$

Preserving source characteristics: To maintain the consistency of the original data from the input image. the Style code of the input image is extracted using the style encoder and used to recreate the original image from the augmented image. The L1 norm is calculated between the original image and reconstructed image.

$$\begin{aligned} \hat{s} &= E_y(x) \\ \ell_{cyc} &= \mathbb{E}[||x - G(G(x, \tilde{s}), \hat{s})||] \end{aligned} \tag{3.19}$$

Total loss: All the loss values are added together along with the learning rate λ_{sty}, λ_{ds} and λ_{cyc} which are hyper parameters. Generator, Mapping network and style encoder optimize the weights by minimizing the total loss and the Discriminator optimize by maximizing the weights.

$$\underset{G,F,E}{min}\,\underset{D}{max}(\ell_{adv} + \lambda_{sty}\ell_{sty} - \lambda_{ds}.\ell_{ds} + \lambda_{cyc}.\ell_c yc) \tag{3.20}$$

3.6 HCR Classifier

A simple HCR classifier has been modelled using convolution 1D layers and Linear layers for the purpose of evaluation of the synthetic data generated from the models mentioned in section 3.5. The HCR classifier takes the time series data, down samples the data to extract the necessary features and processes them through a linear layer to predict the context label the data belongs to.

The classifier takes the generated time series data as input and uses the convolution layer to down sample the data to extract necessary features of the data. After a series of down sampling using convolution layers with different filter sizes and strides, the data is flattened to give input for the Linear layers. The final layer of the liner layers represents the same number of units as the number of context labels used. Equation 3.21 represents the cross entropy loss function used in which $y_{o,c}$ represents the binary indicator whether class c has been predicted correctly or not for observation o and $p_{o,c}$ represents the predicted probability for observation o in class c.

$$-\sum_{c=1}^{M} y_{o,c} log(p_{o,c}) \tag{3.21}$$

3.7 Evaluation Metrics

3.7.1 Shannon Entropy

Shannon Entropy is used as a data balance measure to quantitatively evaluate. The Shannon entropy takes the number of instances under each label as input to computer a value that ranges from 0 to $log(k)$ where k represents the number of classes. The higher the value represents more balanced the dataset is. Equation 3.22 represents the Shannon entropy formula where p represents the probability of the class and k represents each class in the dataset.

$$entropy = -\sum_{k=1}^{N} P(k) * log(P(k)) \tag{3.22}$$

3.7.2 KL Divergence

The Kullback–Leibler (KL) divergence is calculated by taking each data point in real data and calculates the smallest KL divergence with each data point in the Synthetic data. the KL divergence of each data point in real data is calculated with the respective data point along the same axis. Resulting in KL divergence value along all the features. the total KL divergence is calculated by averaging all the values along all axes.

$$D_{KL}(P(z)||Q(z)) = \int_z P(z) log \frac{P(z)}{Q(z)} dz \tag{3.23}$$

3.7.3 Frechet Inception Distance (FID) score

The FID calculates the difference between two Gaussians (Real and Synthetic data). The algorithm uses the Inception V3 model prior to the last pooling layer preceding the classification layer to get the distribution of the real and Synthetic data. The difference of the distribution is calculated to measure how realistic the synthetic data is compared to the real data

$$FID = ||\mu - \mu_w||_2^2 + tr(\Sigma + \Sigma_w - 2(\Sigma^{1/2}\Sigma_w\Sigma^{1/2})^{1/2}) \tag{3.24}$$

3.7.4 Classifier Metrics

Various instances of the same classifier architecture were trained on real data and synthetic data from each translation model. All the classification models will be trained on the training data and the validation set used to check its intermediate performance. After training, the performance metrics of F1 score, recall, precision and accuracy of each model have been recorded and compared to evaluate the impact of synthetic data on the classifier's performance. Table 3.2 represents the confusion matrix for a binary classification, which is a tabular representation of how many values the model has predicted are correct and false compared against the real data. For a multiclass classification problem, for each label the table and classification metrics are calculated with the label as True and all the other label values as False. Accuracy is the metric that estimates the proportion of the correctly predicted values in both True and False scenario among the total number of cases. Precision represents the proportion of True positive values among all the predicted values. Recall is the proportion of True positive values among all the actual Positive values. F1 score is a harmonic mean of precision and recall. The formulas of each classification metric are given in Equation 3.25.

	Predicted Yes	Predicted No
True Yes	True Positive	False Negative
True No	False Positive	True Negative

Table 3.2: Confusion Matrix

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\ Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F1score &= 2.\frac{Precision * Recall}{Precision + Recall} \end{aligned} \tag{3.25}$$

4. IMPLEMENTATION OF GAN MODELS

Each translation model has multiple hyperparameters that can be modified to alter the training of the model and optimize its performance. For each model 3 rounds of hyperparameter tuning have been performed to identify the best performing model.

4.1 DOS-GAN

DOSGAN consists of 4 deep learning models: Domain-specific encoder, Domain-independent encoder, Generator and Discriminator. First a classifier model was trained on the training and validation set for 200,000 iterations with the loss values being logged every 10,000 iterations and the model weights saved for 10,000 iterations. After training the classifier, the output of the second to the last layer is used as the domain-specific features for any sensor data and the classifier is turned into a Domain-Specific encoder. The training of GAN is performed for 200,000 iterations and all the model stored for every 10,000 iterations.

Table 4.1 describes the hyper parameter for the Generative model of DOSGAN along with the default values used for first training. These hyper parameters are common for the training of all the models for DOS-GAN as the pre-train classifier, encoder, generator, and discriminator. The default values were used as the first set of hyperparameters for the training.

Hyper Parameter	Description	Default Values
conv_filters	Number of convolution filters in first later	64
n_blocks	Number of Res. blocks	1
Batch size	Batch size for training	32
λ_{rec}	learning rate for Reconstruction (X_{AA})	10
λ_{rec2}	learning rate for Reconstruction (X_{ABA})	10
λ_{gp}	Learning rate for Gradient Penalty	10
λ_{fs}	Learning rate for Domain specific reconstruction	5
feature-size	The encoded feature space size	1024
λ_g	Learning rate for Generator (Adversarial)	0.0001
λ_d	Learning Rate for Discriminator (Adversarial)	0.0001
n-critic	Frequency of training for Generator	5

Table 4.1: Hyper parameters for DOS-GAN

Table 4.2 summarizes the results of the different sets of hyperparameter tuning performed on the DOSGAN model. Three different models were trained with these sets of hyperparameters, and the validation data was used to generate synthetic data for each model.

	Hyper parameter
Default values	Default values
HP 1	Batch size = 256
HP 2	λ_g = 0.01, λ_d = 0.01, λ_{rec} = 3, λ_{rec2} = 3, λ_{gp} = 3, $\lambda_{fs} = 0.5$

Table 4.2: Hyper parameters for DOS-GAN

4.2 GMM-UNIT

GMM-UNIT is like DOS-GAN and consists of same models. But unlike DOS-GAN, GMM-UNIT uses a normal encoder for domain-Specific encoder and trains it along with other models. All the models are trained in sync for 200,000 iterations and all the model weights are stored for every 10,000 iterations. Based on the loss functions using the sensor data from Generator and the results from discriminator, all the models train and update. Table 4.3 describes various hyperparameters of the model that are common to all the models.

Hyper Parameter	Description	Default Values
n_down and n_up	Number of Down sample and Up sample layer	2
Batch size	Batch size for training	32
λ_{rec}	learning rate for Reconstruction (X_{AA})	1
λ_{recx}	learning rate for Reconstruction (X_{ABA})	10
λ_{adv}	Learning rate for Adversarial	1
λ_{cyc}	Learning rate for Cyclic Consistency	10
λ_{kl}	Learning rate for KL Divergence loss	0.1
feature-size	The encoded feature space size	64
lr	Learning rate	0.0001
Scheduler type	Scheduler Type for Training	Cosine
n-critic	Frequency of training for Generator	5

Table 4.3: Hyper parameters for GMM-UNIT

Table 4.4 summarizes results of the hyperparameter tuning performed to create three different models with different training weights. Each model is used to generate synthetic data using the validation set.

	Hyper parameter
Default values	Default values
HP 1	lr = 0.01, batch size 16,
HP 2	feature size = 128, λ_{adv} = 0.01

Table 4.4: Hyper parameters Tuning for GMM

4.3 STARGAN-V2

The STARGAN V2 architecture consists of a style encoder, Generator, and discriminator. the style encoder is used to generate the features of the target sensor data which is appended with the raw sensor source data and given as input to the generator. All the models are trained together for 200,000 iterations on the training data and all the models are saved for every 10,000 iterations. Table 4.5 lists all the hyper parameter of the model that can be changed to alter the training process.

Hyper Parameter	Description	Default Values
Batch size	Batch size for training	32
λ_{cls}	learning rate for Reconstruction (X_{AA})	1.8
λ_{id}	learning rate for Reconstruction (X_{ABA})	1
λ_{cyc}	Learning rate for Cyclic Consistency	1.2
feature-size	The encoded feature space size	64
lr	Learning rate	0.0001
n-critic	Frequency of training for Generator	5

Table 4.5: Hyper parameters for GMM-UNIT

the sets of hyperparameters in Table 4.6 were used to train three different models of STARGAN-V2. After training, synthetic data was generated by all three models and saved for evaluation purposes.

	Hyper parameter
Default values	Default values
HP 1	feature size = 128, n-critic = 3

Table 4.6: Hyper parameters Tuning for STARGAN

5. EVALUATION AND RESULTS

This section presents and compares results achieved using three different GANs for generating synthetic data augmenting an HCR dataset.

5.1 Shannon Entropy

Two methodologies were explored for the purpose of synthetic data generation to evaluate which methodology produces a balanced dataset. The first methodology randomly samples the data labels while taking the random target data. The data loader randomly picks a label from the available labels and then takes a random data instance with that label as target data to translate the given source data. The next methodology is to directly randomly sample the data, taking the entire dataset and randomly select a data as target data for the given source data. Table 5.1 describes the entropy of the original dataset and the synthetic dataset using different methods. The Shannon entropy of the original dataset and synthetic data from random sampling of data are similar, since the data for target data is randomly sampled from an imbalanced data, the data generated is equally imbalanced as the original data. The synthetic data from random sampling of the labels has produced a balanced dataset since the labels are sampled randomly with the assumption that all the labels are equal with no imbalance, and after the labels have been chosen, the data is randomly sampled from that data. This methodology has been able to mitigate the data imbalance issue that exists in the original dataset and has been employed for data generation of all the generative models.

Datasets	Shannon Entropy
Real Dataset	2.911
Synthetic Dataset using Random sampling of label	3.93
Synthetic Dataset using Random sampling of Data	2.911

Table 5.1: Shannon Entropy of Real and Synthetic dataset

5.2 Domain Specific GAN (DOS-GAN)

The results for the data augmentation are calculated based on the quality of the data generated from the test data both on quantitative measures and practical measures. Evaluation metrics included GAN-specific metrics such as KL Divergence and FID score, as well as a measure of how well the synthetic data generated improved HCR results. Although DOS-GAN was primarily utilized for image-to-image translation in prior work, in this work it was repurposed for use with HCR sensor data.

For each hyper parameter setting in Table 4.2, the classifier, Generator, Encoder and Discriminator were trained and saved for every 10,000 iterations. With the trained model, the testing of the model is performed from the validation set of the Extrasensory Data. For each sensor data in validation set, a random target sensor has been used from the same dataset to translate and augment the data. After completion of testing, random samples of data from each context from real and synthetic data was taken and quantitative measures were calculated for each label. KL divergence had varying values across the labels with few labels recording very little KL divergence and few labels having high values. The average of each metric has been calculated in which the Hyper parameter set 2 has the least KL divergence and FID score of 6.37 and 30.34 indicating the synthetic data from the model with the Hyper parameter has produced quality data which is close to the real world data. From the default hyperparameter set, the KL divergence is least for context labels such as TALKING, OUTSIDE, PHONE in hand and highest for context labels DRIVING, SITTING.

But the FID score is quite consistent across all the labels. Table 5.2 describes the KL divergence and FID score for a few labels and the average of all labels.

Table 5.2: FID and KL score for DOSGAN

Model	Labels	KL	FID score
Default	IN CLASS	0.45	23.28
	FIX RUNNING	2.24	24.00
	PHONE IN HAND	5.28	22.38
	SITTING	10.82	21.63
	SLEEPING	27.77	22.83
	ELEVATOR	494.98	22.85
	Mean	851.28	22.92
Hyper Parameter 1	IN CLASS	2.56	22.87
	FIX RUNNING	29.54	23.20
	PHONE IN HAND	0.58	21.83
	SITTING	2.47	22.20
	SLEEPING	57.98	23.79
	ELEVATOR	16.89	21.58
	Mean	98.08	22.51
Hyper Parameter 2	IN CLASS	4.25	30.42
	FIX RUNNING	24.65	30.01
	PHONE IN HAND	0.61	29.92
	SITTING	1.50	30.58
	SLEEPING	7.65	31.62
	ELEVATOR	1.82	28.33
	Mean	**6.37**	**30.34**

5.3 GMM-UNIT

From the synthetic data generated from each model of GMM-UNIT, random samples of each label are taken from real and synthetic data. Quantitative measure algorithms are performed on the random samples to evaluate the quality of the data generated. Table 5.3 describes the KL Divergence and FID score calculated for few labels and an average KL divergence and FID score is also calculated across all labels. The average of each metric has been calculated in which the Hyper parameter set 2 has the least KL divergence and FID score of 4.11 and 21.91 indicating the synthetic data from the model with the Hyper parameter has produced quality data which is close to the real world data. From the optimized hyperparameter set, the KL divergence is least for context labels such as ALCOHOL, HOME, BEACH in hand and highest for context labels Lying Down, Phone in bag.

Table 5.3: FID and KL score for GMM-UNIT

Model	Labels	KL	F1 score
Default	IN CLASS	4.40	22.30
	FIX RUNNING	1.19	21.98
	PHONE IN HAND	3.74	22.20
	SITTING	2.14	21.83
	SLEEPING	16.31	22.63
	ELEVATOR	0.438	21.89
	Mean	6.24	22.04
Hyper Parameter 1	IN CLASS	0.57	25.32
	FIX RUNNING	31.06	25.39
	PHONE IN HAND	0.44	25.32
	SITTING	1.05	25.53
	SLEEPING	0.35	25.60
	ELEVATOR	0.67	24.42
	Mean	6.36	25.37
Hyper Parameter 2	IN CLASS	2.81	21.82
	FIX RUNNING	1.45	22.47
	PHONE IN HAND	0.62	21.17
	SITTING	3.33	22.17
	SLEEPING	21.59	21.67
	ELEVATOR	0.64	21.84
	Mean	**4.11**	**21.91**

5.4 STARGAN-V2

.

From the synthetic data generated from each model of STARGAN-V2, random samples of each label are taken from real and synthetic data. Quantitative measure algorithms are performed on the random samples to evaluate the quality of the data generated. Table 5.4 describes the KL Divergence and FID score calculated for few labels and a, average KL divergence and FID score is also calculated across all labels. The average of each metric has been calculated in which the Hyper parameter set 1 has the least KL divergence and FID score of 5.24 and 114.35 indicating the synthetic data from the model with the Hyper parameter has produced quality data which is close to the real world data. From the optimized hyperparameter set, the KL divergence is least for context labels such as Outside, exercise, cleaning and highest for context labels Indoor, cooking., Strolling. Unlike all the other generative model, the FID score for the synthetic data from STARGAN is the most varying with high different between different labels. The total average as well as individual labels are also high. For the optimized Hyper parameter, the FID score is highest for the context labels Computer work, workplace and lowest for Singing, Sleeping.

Table 5.4: FID and KL score for STARGAN-V2

Model	Labels	KL	F1 score
Default	IN CLASS	1.56	220.19
	FIX RUNNING	20.46	233.43
	PHONE IN HAND	3.04	390.96
	SITTING	23.57	388.24
	SLEEPING	41.76	408.89
	ELEVATOR	0.92	455.75
	Average	16.07	610.69
Hyper Parameter 1	IN CLASS	2.39	126.72
	FIX RUNNING	0.64	21.098
	PHONE IN HAND	0.50	175.38
	SITTING	2.58	354.79
	SLEEPING	1.063	2.26
	ELEVATOR	0.55	24.62
	Average	**5.24**	**114.35**

5.5 Quantitative comparison

The average of all quantitative measures was calculated across all labels for all synthetic data generated. Table 5.5 compares the quality of data between each synthetic data from different models.

Table 5.5: FID and KL score comparison

Model	Hyper parameter	KL	F1 score
DOSGAN	Default	851.28	22.92
	HP 01	98.08	22.51
	HP 02	6.37	30.34
GMM-UNIT	Default	6.24	22.04
	HP 01	6.36	25.37
	HP 02	**4.11**	**21.91**
STARGAN	Default	16.07	610.69
	HP 01	5.24	114.35

5.6 Classifier performance

The next set of evaluations focused on evaluating how much HCR performance improved when synthetic data was used to augment the HCR dataset. The test dataset was split into training and validation sets for the purpose of classifier training. Multiple classifier models were initialized with the same architecture and hyper parameters and trained on real and synthetic data of each GAN model. The model training loss has been recorded. After the model being trained on real and synthetic data, the model classification performance metrics Accuracy, Precision, Recall and F1 scores have been calculated on the validation set. From Table 5.6, the model trained on GMM-Unit on hyper parameter set 1 can produce similar results with a minor improvement compared to the model trained on the original dataset. The classifier performance trained on synthetic data from DOSGAN with default hyper parameters is the least, which coincides with the quantitative evaluation results with the data having the highest KL Divergence. An oversampling and under sampling dataset has been created in which the data from all the labels has been over sampled and under sampled to an equal amount of data. All the data from the labels with less data points has been over sampled and the data from the labels with a higher number of data points is under sampled. The model trained on this over sampled and under sampled data is used as a baseline for data augmentation without deep learning architecture.

Model	Hyper parameter	Accuracy(%)	F1 score	Precision	Recall
Real	Original	53.71	9.65	53.71	16.31
	Over sampling and Under sampling	48.52	7.61	48.52	13.15
DOS-GAN	Default	0.71	0.02	0.71	0.04
	HP1	9.8	0.5	9.8	1.06
	HP2	1.03	0.03	1.03	0.07
GMM-GAN	Default	53.71	9.57	53.71	16.02
	HP1	**53.71**	**9.79**	**53.71**	**16.31**
	HP2	38.71	5.40	38.71	9.33
STARGAN-V2	Default	7.61	2.71	7.61	3.99
	HP1	14.5	3.12	14.5	5.13

Table 5.6: Performance metrics of different classifier trained on synthetic data from generative models

6. DISCUSSION

This chapter gives answers to our research questions proposed in chapter 1 and concludes some experimental remarks.

Performance of Image-to-Image models for Sensor based time series data on the task of data augmentation: : In this thesis, the proposed GAN models have been trained by employing Conv1D networks for down sampling and up sampling of the sensor data in the Generator and Discrimina- tor. The implemented models have generated synthetic and balanced data. The Shannon entropy of the dataset has increased, showing that the imbalance in the dataset has decreased. All the context labels have a similar number of data points available. From Table 5.2, Table 5.3 and Table 5.4,different labels have high and low KL Divergence and FID scores with no context label always having low or high KL Divergence. The FID score of STARGAN-V2 is higher and more variable than other generative models.

Context labels that GAN models generated best and effects of hyperparameters: Based on the KL Divergence values calculated from the synthetic data created from the models implemented, various GAN models achieved varying performance for generating different context labels. Table 5.2 shows that the most KL Divergence consistent hyper parameter set for DOS-GAN is Set 2. The data generated from DOSGAN with default values had the lowest data quality and highest KL Divergence. With the increase in learning rate and feature space size, the performance of the generative model improved to create quality data with less KL Divergence from Hyper param- eter set 2. From Table 5.3representing Quantitative evaluation of data from GMM-UNIT, shows that all the GAN models with different hyperparameters were able to generate quality data that had both KL divergence and FID scores consistent. With higher learning rates, the model was able to further optimize performance. The data from the model with Hyperparameter set 2 achieved the lowest KL Divergence and FID scores. From Table 5.4 epresenting quantitative evaluation for STARGAN-V2, the initial model had a very high KL Divergence and FID score. The FID score for the data from STARGAN-V2 is unexpectedly high and varying. For all the other models, the FID score remained consistent with very few variations, but the STARGAN-V2 GAN model generated data with higher variations. With hyperparameter tuning, the KL Divergence and FID scores are both reduced, but the FID score is high compared to other models. From the quantitative evaluation of all generative models, no context label had consistently high or low KL Divergence or FID score From Table 5.5, the data from GMM-UNIT achieved the lowest KL Divergence and FID scores compared to other generative models.

How much did the performance of the HCR classifier model improve when trained on synthetic data: from Table 5.6, the classifier performance trained on data from GMM-unit is comparatively higher than results achieved by training on synthetic data generated by the other GAN models. However, even this synthetic data was only able to achieve similar performance compared to the classifier trained on the original (real data) dataset. The GMM-unit with hyperparameter set 1 had the highest performance metrics even though Hyper parameter set 2 achieved the lowest KL Divergence and FID score. The generative model GMM-UNIT has consistently proven to have generated quality synthetic data both in quantitative evaluation and performance evaluation.

6.1 Limitations:

One of the major limitations of the comparison models developed is to augment data belonging to different context label such as SITTING and LOC_HOME representing a person is sitting at home. To augment such data from any source label, a data must already exist with the same combination of the context labels. Another major limitation of the models developed is the translation of data from one user to another, the models developed have a context label centric and not user centric. The data and model must modify and re-trained to accommodate such tasks to augment data for a user that has lesser data from data from other users.

6.2 Future work

The study has developed and compared models that are based on context labels. The models can be explored to accommodate user information so that in future context data for any user can be augmented using data from other users. The data rather than being collected for the new users for a elongated period of time, the generative model can be used to generate sufficient data for all the context labels and the classifier model can be trained on the newly synthesized data to produce similar performance results as the classifier for existing user. The model can also be explored to take context labels as input along with the feature values generated by the encoder in order to augment data for a combination of context labels. This could help to augment data for context label combinations that are not available in the dataset. The evaluation of these generative models has been done only on the Extrasensory dataset. In future, the proposed GAN models can be evaluated on other HCR dataset which has collected in-wild data from smart phones or smart watches such as Warfighter Analytics for Smartphone Healthcare (WASH). Apart from HCR data, the models can also be implemented to evaluate the synthetic data for other time series data.

7. CONCLUSION

Implementing HCR is a crucial and difficult task due to the data privacy and data scarcity issues that arise. Collection of this data is typically from volunteers and is expensive. Moreover, various people visit different contexts and unequal frequencies, causing imbalanced in HCR labels col- lected. Such an imbalance reduces the performance of HCR machine learning classification models. The proposed thesis tackles the issue of Data Augmentation of the data to create a more balanced dataset that can be used by other deep learning models to improve performance. GANs for image-to-image translation were repurposed and used for HCR sensor data augmentation. All three models have performed well by creating quality and diverse data in quantitative measures. Each model has an optimal hyper parameter set that has produced optimal results. The data generated from the models has produced quality data that has been used to improve a classifier in different metrics. The GMM-UNIT has been able to generate quality data with the consistency of KL Divergence 4.11 and FID Score of 21.91. The data generated from the same model has been able to get similar classifier performance as the model trained on real data with a improvement of 0.72% in accuracy and 0.30 in F1 score.

REFERENCES

[1] 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.

[2] M. Alzantot, S. Chakraborty, and M. Srivastava, “Sensegen: A deep learning architecture for synthetic sensor data generation,” in 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 188–193, IEEE, 2017.

[3] 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.

[4] S. Norgaard, R. Saeedi, K. Sasani, and A. H. Gebremedhin, “Synthetic sensor data generation for health applications: A supervised deep learning approach,” in 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1164–1167, IEEE, 2018.

[5] 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.

[6] X. Li, J. Luo, and R. Younes, “Activitygan: generative adversarial networks for data augmentation in sensor-based human activity recognition,” in Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers, pp. 249–254, 2020.

[7] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in

Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3234–3243, 2016.

[8] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in CVPR Workshop on the Future of Datasets in Vision, vol. 2, sn, 2015.

[9] Y. Vaizman, K. Ellis, G. Lanckriet, and N. Weibel, "Extrasensory app: Data collection in-the-wild with rich user interface to self-report behavior," in Proceedings of the 2018 CHI conference on human factors in computing systems, pp. 1–12, 2018.

[10] D. Blachon, D. Coşkun, and F. Portet, "On-line context aware physical activity recognition from the accelerometer and audio sensors of smartphones," in European Conference on Ambient Intelligence, pp. 205–220, Springer, 2014.

[11] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in International conference on intelligent computing, pp. 878–887, Springer, 2005.

[12] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), pp. 1322–1328, IEEE, 2008.

[13] 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.

[14] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

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

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

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134, 2017.

[18] A. Yu and K. Grauman, “Fine-grained visual comparisons with local learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 192–199, 2014.

[19] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in European conference on computer vision, pp. 597–613, Springer, 2016.

[20] Z. Liu, P. Luo, X. Wang, and X. Tang, “Large-scale celebfaces attributes (celeba) dataset,” Retrieved August, vol. 15, no. 2018, p. 11, 2018.

[21] H.-W. Ng and S. Winkler, “A data-driven approach to cleaning large face datasets,” in 2014 IEEE international conference on image processing (ICIP), pp. 343–347, IEEE, 2014.

[22] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool, “Combogan: Unrestrained scalability for image domain translation,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 783–790, 2018.

[23] A. Alotaibi, “Deep generative adversarial networks for image-to-image translation: A review,” Symmetry, vol. 12, no. 10, p. 1705, 2020.

[24] Z. Yongxin, “A survey of image to image translation with gans,”

[25] H. Guan, X. Ma, and S. Shen, “Dos-gan: A distributed over-sampling method based on generative adversarial networks for distributed class-imbalance learning,” in International Conference on Algorithms and Architectures for Parallel Processing, pp. 609–622, Springer, 2020.

[26] Y. Liu, M. De Nadai, J. Yao, N. Sebe, B. Lepri, and X. Alameda-Pineda, “Gmm-unit: Unsupervised multi-domain and multi-modal image-to-image translation via attribute gaussian mixture modeling,” arXiv preprint arXiv:2003.06788, 2020.

[27] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8188–8197, 2020.