Sampling Strategies for Tackling Imbalanced Data in Human Activity Recognition

Fayez Alharbi
Department of Computing, Goldsmiths, University of London

First Supervisor Dr Jamie A Ward
Second Supervisor Dr Lahcen Ouarbya

This dissertation is submitted for the degree of Doctor of Philosophy
June 2021

Declaration

I declare that this thesis titled "Sampling Strategies for Tackling Imbalanced Data in Human Activity Recognition" has been composed by myself. It also has not been submitted for any previous application for a degree or other qualifications.

Signature

Date: Fayez Alharbi

ACKNOWLEDGEMENT

I would first like to thank God for bestowing upon me all the strength, good health, and mental state to conduct this research.

This research would not have been accomplished without my first supervisor's support and guidance, Dr Jamie A Ward. I would like to express my greatest and sincerest gratitude and appreciation to Dr Ward for giving me the opportunity to work under his supervision.

I would like to thank him for allocating a lot of his time, providing generous support, and sharing his ideas and research knowledge which helped me every step of the way, I learn a lot from him.

The year 2020 was a challenging one with the COVID-19 pandemic crisis; I insist on thanking Dr Ward for his support throughout the difficult time. He is a caring and incredibly supportive supervisor.

Next, special thanks go to my second supervisor, Dr Lahcen Ouarbya. He has always been so supportive, offering me with his full support, ideas, and suggestions to improve my work. I am incredibly blessed to have a supervisor who cared about my academic development, and I am really grateful to Dr Ouarbya.

I would like to thank the National Health Service and other key workers for their efforts and for putting themselves at risk in the COVID-19 pandemic in order to keep everyone safe in the United Kingdom.

I would next like to thank the Goldsmiths, University of London members, and especially the Computing Department. They have always been supportive, and they showed extraordinarily effort to minimize the impact of the COVID-19 pandemic on students.

I would like to thank the members of the Kingdom of Saudi Arabia's embassy in London for their supports when the COVID-19 pandemic began and for offering their financial and logistical support, especially in the form of a free flight ticket to return home.

I would like also to thank my previous supervisors who departure Goldsmiths, University of London for other careers for their advice and guidance.

Special thanks go to my family. Thank you to my mother and father, my dear wife, my sisters, and my brothers for all the love and encouragement they have offered me along the way.

Most importantly, I say thank you for the beat of my heart, my daughter Rema, and my son Salman. Their existence in my life is the energy of my soul that pushes me every single day forward to work hard to complete my studies and always do better.

Abstract

Human activity recognition (HAR) using wearable sensors is a topic that is being actively researched in machine learning. Smart, sensor-embedded devices, such as smartphones, fitness trackers, or smartwatches that collect detailed data on movement, are widely available now. HAR may be applied in areas such as healthcare, physiotherapy, and fitness to assist users of these smart devices in their daily lives. However, one of the main challenges facing HAR, particularly when it is used in supervised learning, is how balanced data may be obtained for algorithm optimisation and testing. Because users engage in some activities more than others, e.g. walking more than running, HAR datasets are typically imbalanced. The lack of dataset representation from minority classes, therefore, hinders the ability of HAR classifiers to sufficiently capture new instances of those activities.

Inspired by the concept of data fusion, this thesis will introduce three new hybrid sampling methods. Thus, the diversity of the synthesised samples will be enhanced by combining output from separate sampling methods into three hybrid approaches. The advantage of the hybrid method is that it provides diverse synthetic data that can increase the size of the training data from different sampling approaches. This leads to improvements in the generalisation of a learning activity recognition model.

The first strategy, known as the *distance-based method* (DBM), combines synthetic minority oversampling techniques (SMOTE) with Random_SMOTE, both of which are built around the k-nearest neighbours algorithm. The second technique, called the *noise detection-based method* (NDBM), combines Tomek links (SMOTE_Tomeklinks) and the modified synthetic minority oversampling technique (MSMOTE). The third approach, titled the *cluster-based method* (CBM), combines cluster-based synthetic oversampling (CBSO) and the proximity weighted synthetic oversampling technique (ProWSyn). The performance of the proposed hybrid methods is compared with existing methods using accelerometer data from three commonly used benchmark datasets. The results show that the DBM, NDBM and CBM can significantly reduce the impact of class imbalance and enhance F1 scores of the multilayer perceptron (MLP) by as much as 9 % to 20 % compared with their constituent sampling methods. Also, the Friedman statistical significance test was conducted to compare the effect of the different sampling methods. The test results confirm that the CBM is more effective than the other sampling approaches.

This thesis also introduces a method based on the Wasserstein generative adversarial network (WGAN) for generating different types of data on human activity. The WGAN is

more stable to train than a generative adversarial network (GAN) and this is due to the use of a stable metric, namely Wasserstein distance, to compare the similarity between the real data distribution with the generated data distribution.

WGAN is a deep learning approach, and in contrast to the six existing sampling methods referred to previously, it can operate on raw sensor data as convolutional and recurrent layers can act as feature extractors. WGAN is used to generate raw sensor data to overcome the limitations of the traditional machine learning-based sampling methods that can only operate on extracted features. The synthetic data that is produced by WGAN is then used to oversample the imbalanced training data. This thesis demonstrates that this approach significantly enhances the learning ability of the convolutional neural network (CNN) by as much as 5 % to 6 % from imbalanced human activity datasets.

This thesis concludes that the proposed sampling methods based on traditional machine learning are efficient when human activity training data is imbalanced and small. These methods are less complex to implement, require less human activity training data to produce synthetic data and fewer computational resources than the WGAN approach. The proposed WGAN method is effective at producing raw sensor data when a large quantity of human activity training data is available. Additionally, it is time-consuming to optimise the hyperparameters related to the WGAN architecture, which significantly impacts the performance of the method.

List of Publications

F. Alharbi and K. Farrahi, "A convolutional neural network for smoking activity recognition," in 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services, Healthcom 2018, 2018.

F. Alharbi, L. Ouarbya, and J. A. Ward, "Synthetic Sensor Data for Human Activity Recognition," Proc. Int. Jt. Conf. Neural Networks, 2020.

F. Alharbi, L. Ouarbya, and J. A. Ward, "Comparing Sampling Strategies for Tackling Imbalanced Data in Human Activity Recognition," this work is in submission to Multidisciplinary Digital Publishing Institute (MDPI) Sensors journal 2021.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, in section 1.1, we discuss the importance of and define the main objective of human activity recognition. We also talk about the main approaches that are often used to recognise human activity. We then introduce the main applications of human activity recognition that are related to this thesis.

The motivation of this thesis is presented in section 1.2, and the aim in section 1.3. The thesis's contribution is in section 1.4. Finally, in section 1.5, the thesis outline is given.

1.1 The Impotance of Human Activity Recognition

The main purpose of human activity recognition (HAR) is to automatically recognise human physical activities [2]. This is an active research topic in mobile and ubiquitous computing [3]. Activity recognition is useful tool because it provides information on an individual's behaviour that allows computing systems to monitor and analyse, and assists individuals with their day-to-day tasks [1]. In order to be able to benefit form recognising human activities, different techniques have been commonly applied. For example, video-based systems, sensor-based systems that include wearable, also known as body-worn sensors, as well as ambient sensors [3]. These sensors collect what is known as time-series data. Video-based systems which is rely on using images or videos taking by utilise cameras to identify individuals' behaviours or physical activities.

Sensor-based systems include wearable sensors such as the accelerometer and gyroscope that are embedded in most smartphones and smartwatches [4]. They are one of the most popular ways to monitor user activity as individuals most commonly wear these motion sensors. Sensor-based systems also use ambient sensors. The interaction between humans and the smart environment can be captured by embedding ambient sensors in people's environment, for example, pressure sensors and temperature sensors [5].

However, the video-based systems raise some concerns, most commonly due to privacy and high cost [6]. If we consider installing cameras in individuals' spaces to monitor and record their activities, some individuals may perceive that as a threat to privacy. Another issue that may arise is complexity, due to video processing techniques that are computationally intensive and expensive [6]. Therefore, these issues hinder video-based systems from being suitable for HAR.

The limitations mentioned above have resulted in a shift towards using wearable sensors as the main tool to recognise human activities because they can directly and efficiently capture body movements [3]. Advances in wearable sensor technology now enable sensors such as the accelerometer and gyroscope to be integrated into smartphones and smartwatches [3]. For example, smartphones and smartwatches main features are their portable, computational power, communication capability, and their embedded sensors [5]. These characteristics have allowed smart devices to become the primary platform for HAR because of their capability to obtain contextual information from various real-world settings [5]

The main objective of HAR systems is not only to monitor and analyse human activities to interpret ongoing events using wearable sensors, but also to serve as an essential step in several applications [3]. There are numerous application domains where HAR models are applied, for example, activities of daily living, health [6], transportation mode [7] and behavioural analysis such as in smoking activity detection [8].

The activities of daily living (ADL) refers to activities that are performed daily and

are necessary for independent living, such as personal care, eating and mobility [9]. The first work carried out related to ADL proposed a specific set of activities including *bathing, dressing, toileting, transferring* and *feeding* that called activities of daily living in order to provide a standardized way to estimate the physical well-being of the elderly and their need for assisted living [9]. In this thesis, we use three datasets that correspond to the ADL applications, including Opportunity [10], Physical Activity Monitoring (PAMAP2) [11]. In addition, Activity Recognition from a Single Chest-Mounted Accelerometer dataset which we refer to as activities of daily living (ADL) as the data participants perform different activities of daily living such as *walking* and *standing* [12]. The three datasets are recorded to help researchers evaluate their methods for ADL applications. The three datasets comprise various classes of activities of daily living which motivated us to use them in this thesis to evaluate the performance of the sampling methods that are based on traditional machine learning. A detailed account of these datasets is provided in section 4.3.1 .

Physical activity is significant for individuals' physical and mental well-being and the absence of such activity may negatively influence it [5]. Transportation mode applications can provide related information that designates individual mobility status during travel (i.e. *walking, cycling, taking a bus* or *driving a car*) [13]. Transportation mode applications enable individuals to monitor their physical activity to improve their well-being [14]. Such applications can determine an individual's transportation mode when outside, which enables individuals to plan their modes of transportation based on their aims of physical activity [15]. Gjoreski et al. [16] indicated that a limited datasets are available for locomotion and transportation modes application. That motivated them to collect and annotate a large-scale dataset known as the University of Sussex- Huawei Locomotion-Transportation (SHL). SHL dataset provides researchers with a richly annotated dataset for transportation modes application studies. This dataset is used in this thesis to assess the performance of the WGAN model. The WGAN needs large amount of training data. The SHL dataset contains large quantities of data which makes it suitable for this experiment. More details about the SHL dataset will be presented in section 5.2.9.

Another potential type of human activity recognition application are applications that can recognize more complex activities and assist individuals in tracking their routines and patterns [17]. For example, one could develop systems that help individuals track their routines and behavioural patterns, particularly their smoking patterns [8]. This type of application can be used to collect data from individuals for smoking cessation programs where individuals can self-report their smoking activity [8]. It can be beneficial for individuals to track their smoking patterns, if one is able to know the number of cigarettes they smoke and at what time [18]. It will help them to be more consciously aware of their behaviours [19]. With such tools, individuals may be able to improve their well-being by reducing their smoking or by

stopping completely [19]. Shoaib et al. [17] collected a large dataset for smoking activity and made it available for researchers to evaluate approaches for smoking detection applications. We used this dataset to introduce a model for smoking activity detection and a version of the study shows in the appendix A. Section 5.2.9 provides more details about the smoking activity dataset.

The WGAN model requires a large quantity of training data in order to train; therefore, we also use the smoking activity dataset here due to its high quantity of data.

There are various applications of human activity recognition. This thesis focuses on applications that are among the more common in the domain and more useful, including the ADL and transportation mode as well as behavioural analysis, particularly smoking detection applications. The following section will provide the reasons that motivate this thesis.

1.2 Motivation

The human activity recognition (HAR) problem is usually considered as a typical pattern recognition problem [20]. A typical HAR system for time-series data collected from wearable sensors such as accelerometer and gyroscopes is often developed using the following steps (phases): pre-processing, segmentation, feature extraction, and activity classification [1]. The pre-processing stage includes operations such as filtering for noise elimination and data cleaning [5].

It is a challenging task to retrieve valuable information from a continuous stream such as sensor data [21]. A popular way to deal with this is to break up continuous data into discrete chunks or segments. The segmentation step is required in order to collect meaningful information from sensor data [22]. A segmentation approach is applied to divide the sensor data into individual segments. Labels can then be assigned to each segment [5].

The feature extraction phase is responsible for representing the activities by extracting features from each segment in order to best characterise sensor data [23]. Typical features might include mean and standard deviation [24].

The final phase is the classification. Here a supervised learning algorithm is often implemented and uses the extracted features to make a decision as to which activity the data belongs [2].

Supervised learning algorithms often require a substantial quantity of annotated data for training and evaluation [25]. However, the availability of a large volume of training labelled sensor data for human activity recognition is scarce for two reasons [26]. Namely, the sensor data collection is costly, as well as the labelling of sensor data procedure is time-consuming. In addition, in real scenarios, activities often have a different number of samples because of the differences in their duration (e.g. a regular individual *sleep* time will definitely be longer than their *eating* or *drinking* time) [1]. Some human activity also has a lower frequency of occurrence than other performed activities such as *walking* or *sitting* might occur more than *running* . Consequently, a major issue is introduced which is known as a class imbalance [4]. When any class (activities) is underrepresented in the dataset, a dataset is considered to be imbalanced [27].

Most of human activity recognition studies adopt a supervised model approach [28]. These approaches often need immense amount of labelled sensor data in order to train [25]. Alternately, datasets for human activity recognition are often imbalanced, which create a significant challenge when constructing and training a supervised model [29]. When a supervised model is implemented to recognise human activity by utilizing imbalanced sensor data, the performance tends to be biased towards learning and identifying the more represented activities [30]. These activities are known as majority classes and underrepresented activities are identified as minority classes [27].

HAR research often relies on the quantity and the quality of the used sensor data [25]. Sensor data quality is usually inadequate and frequently occurs alongside missing data [31]. This occurs due to a number of factors, for instance, an individual not wearing a sensor or malfunctioning [31]. Similarly, the sensor data may often be highly imbalanced due to significant individual variations, with limited labels for certain activities [32].

Obtaining data from real-life can be challenging [25]. First, individuals may not be able to have several devices due to issues as cost [33]. Secondly, issues can arise related to privacy concerns as some individuals may prefer to use specific or fewer sensors [6]. Individuals might also not choose to enable multiple sensors as it can increase battery consumption of their devices [34].

In addition, other sensor data of certain activities can be difficult to come by, for example, falls activities related to the older people) [25]. Recording sensor data in unrestricted environments can introduce class imbalance [25].

As a result, finding solutions that can increase the number of samples of human activity sensor data can efficiently increase the performance of HAR systems. It has been also emphasised in [1] and recently in [25] that the class imbalance in human activity is a potential problem that needs to be addressed and resolved.

The imbalance of class distribution is not the only issue that hinders a supervised model's performance [30]. Despite this imbalance, other issues that affect the performance might arise, such as small sample size, class overlapping, and within-class imbalance [35].

There are many explanations as to why imbalanced class distribution occurs, one of which is that some activities are performed less often than others [30]. As a result, smaller sample sizes for these activities occur, so a supervised model might not have enough data to learn a pattern in an activity adequately [36].

Class overlapping can occur when applying the sliding window approach to segment sensor data [37]. The label within a segment is often taken from the majority vote of constituent samples [21]. This can lead to ill-defined class boundaries resulting in overlap [38].

A within-class imbalance might occur due to intraclass variability. The intraclass variability is a situation where the same activity is often performed in different way by the same individual [1].

There are usually two methods to approach the issue of class imbalance: data level (sampling) and algorithm level methods [27]. Data level techniques' main objective is to solve the problem by changing a training set's class distribution, which includes oversampling, undersampling, and combined sampling methods (using oversampling and undersampling techniques). In term of the second technique, during this process, algorithm level methods adjust existing learning algorithms to focus more on the minority classes. Both methods are capable of decreasing the degree of class imbalance [1]. We aim to focus on this problem by using

data level methods, which are often not only useful and less complex to configure but can be integrated with any learning algorithm [29]. In the next section, we indicate the reasons of the proposed solutions of this thesis.

1.3 Proposed Solutions

Motivated by the concept of data fusion, this thesis proposes three new hybrid sampling methods. The fusion of several data sources and sensor modalities has been widely researched and is used well in HAR (e.g. [39], [40], [2] and [41]). Multimodal feature fusion is also typically used in HAR as an efficient way to improve recognition performance [42].

Similarly, the fusion of multiple, diverse, weak learners to produce a strong ensemble is a popular approach and is a very effective technique in machine learning [43].

The diversity of the synthesised samples is enhanced by combining output from separate methods into three hybrid approaches DBM, NDBM and CBM. The hybrid sampling method is beneficial not only because it provides diverse synthetic data, but it also increases the size of the training data from different sampling methods. This can lead to improvements in the generalisation of a learning activity recognition model.

Variational auto-encoders (VAEs) and generative adversarial networks (GANs) use Kullback-Leibler (KL) divergence and the Jensen-Shannon (JS) divergence, respectively, in their cost function [44]. KL and JS are probability metrics that measure the similarity between two probability distributions [45]. Arjovsky et al. [44] contended that KL and JS often hinder the performance of VAEs and GANs. For example, the VAE is sometimes incapable of generating good quality samples [46].

In terms of GAN, JS causes training instability that often makes training a GAN cost function a challenging task [47]. When the discriminator network is trained optimally, particularly in the early stages of the training process, the cost function falls to zero [47]. Consequently, there probably is no gradient left to update during the training processes [47]. This is because the discriminator network will always be able to differentiate between real and synthetic samples. Therefore, the vanishing gradient problem is introduced [48]. In addition, if the discriminator network is not trained well, this could result in the generator not performing properly [49]. This is because the generator does not receive appropriate feedback from the discriminator, and the learned cost function might not be capable of generating accurate samples [50].

The Wasserstein GAN (WGAN) is introduced to overcome the training difficulty of the original GAN [48]. For the original GAN, the cost function of the discriminator network is determined by the binary classification of real and synthetic samples, while the cost function of the discriminator (also called critic) in WGAN is represented by the Wasserstein distance between real and synthetic distributions [44]. The discriminator of WGAN is formulated as a regression task that compares with GAN, which was a classification task [50]. In other words, the discriminator/critic network does not directly discriminate between real and synthetic samples but estimates the Wasserstein distance between synthetic and real sample distributions [44].

Arjovsky et al. [44] presented the mathematical argument that the Wasserstein distance possesses the properties of being both continuous and differentiable and it can provide a reliable and usable gradient for the cost function even after the critic has been well trained. For these reasons, this thesis decided to use WGAN to generate sensor data. The aim of this thesis and the questions it seeks to answer will be presented in the next section.

1.4 Aim

In this thesis, we aim to use sampling methods to overcome the class imbalance issue in human activity recognition in order to improve the generalisation ability of a learning algorithm.

This thesis focuses on two different ways in order to introduce four sampling methods and to overcome the challenge of class imbalance. The first way is to use sampling techniques that generated data of minority classes or eliminate some majority class samples based on shallow machine learning. In order to use a sampling method such as Synthetic Minority Over-sampling Technique (SMOTE), to produce synthetic activity samples (e.g. sensor features), we must extract features from the raw sensor data such as mean and standard deviation. The reason is that these sampling methods including SMOTE do not operate directly on time series data such as raw sensor data (more details in [51]).

The second way to deal with the challenge of class imbalance is to utilize an approach based on deep learning, such as Generative Adversarial Networks (GANs), which does not require hand-crafted features, such as time-domain features including mean or maximum. In a case where GAN is applied to generate synthetic samples, the raw sensor data can be used directly with the GAN to generate synthetic human activity data [51].

For small sets of imbalanced human activity data, we apply both oversampling and undersampling methods that are capable of handling typical issues related to small sample size, class overlap and within-class imbalance. In total, we use six different sampling methods to construct three different novel hybrid oversampling methods (which combine different sampling approaches) in order to decrease the effect of class imbalance. We show how this approach can enhance the performance of human activity recognition on three public datasets including Opportunity [10], Physical Activity Monitoring (PAMAP2) [11], and Activity Recognition from a Single Chest-Mounted Accelerometer [12]. The participants performed different daily living activities such as *Standing Up, Walking and Going up/downstairs*; hence, we call the Single Chest-Mounted Accelerometer dataset as (ADL).

The six basic sampling methods that we use to build our proposed approach included Synthetic Minority Over-sampling Technique (SMOTE) [52], Random_SMOTE algorithm [53], Smote with Tomek links (SMOTE_Tomeklinks) [54], Modified Synthetic Minority Over-sampling Technique (MSMOTE) [55], Cluster-Based Synthetic Oversampling algorithm (CBSO) algorithm [56], and Proximity Weighted Synthetic Oversampling Technique (ProWSyn) [57].

The hybrid methods we propose are as follows. The first method is based on SMOTE and Random_SMOTE algorithms, which we call distance-based method (DBM), to handle the small sample size in the training data. The second proposed technique is built using SMOTE_Tomeklinks and MSMOTE algorithms, which we call noise detection-based method (NDBM), and we develop this method to handle class overlapping problem in the training

data during producing of synthetic samples. The third propose method is combined CBSO and ProWSyn algorithms, which we call cluster-based method (CBM). We explore how the cluster-based method considers the within-class imbalance issue in the training data whilst generating synthetic samples.

Generative adversarial networks (GANs) are an approach that is able to generate synthetic data and are adopted in several fields, for example, language generation and speech recognition. [28]. However, few works adopt GANs in the domain of human activity recognition.

We investigate and build models for producing a number of types of human activity sensor data by implementing a Wasserstein generative adversarial network (WGAN) [44]. The WGAN is used in this thesis because it is more stable than GAN [49]. The WGAN effectively might deal with not only intraclass variability that might produce within-class imbalance but aslso class overlapping issues because it is based on deep learning methods that can automatically capture different patterns within raw sensor data such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). WGAN requires substantial amounts of data to train, therefore, we use two large public human activity datasets to demonstrate how this fourth approach could increase the performance of human activity recognition. We use the Sussex-Huawei Locomotion (SHL) [16], and the Smoking Activities Dataset (Smoking) [17].

We evaluate the synthetic data by applying two regularly apply classification algorithms: CNN and LSTM.Then, the quality and diversity of the artificial data are assessed by training on synthetic data and testing on real sensor data, and vice versa. The synthetic sensor data then are used to oversample the imbalanced training set.

It is also important to indicate that we focused on ambulation and daily and transportation activities because they are common activities in real-life.

This thesis deals with the class imbalance problem for human activity recognition, and we address the following questions:

1. Can the performance of a supervised model be enhanced by using the existing sampling methods when training with imbalanced human activity data?

In order to answer this question, we compare six different sampling methods to determine how they can enhance the performance of a supervised model when imbalanced human activity dataset is used. The sampling methods are: Synthetic Minority Over-sampling Technique (SMOTE), Random_SMOTE algorithm, Smote with Tomek links (SMOTE_Tomeklinks), Modified Synthetic Minority Over-Sampling Technique (MSMOTE), Cluster-Based Synthetic Oversampling algorithm (CBSO), and Proximity Weighted Synthetic Oversampling Technique (ProWSyn) .

2. Can combining different sampling methods to generate synthetic human activity data enhance a learning algorithm performance of learning from imbalanced human activity

dataset?

To address the second question, we propose three sampling methods to explore their feasibility in enhancing the performance of a supervised model.

The first introduce approach is developed by combining SMOTE and Random_SMOTE. It is called distance-based method (DBM) because SMOTE and Random_SMOTE methods use K nearest neighbours in the process of oversample data.

The second approach, noise detection-based method (NDBM), is created by combining sampling methods the SMOTE_Tomeklinks and the MSMOTE where these algorithms can perform both oversampling and undersampling.

The third method combines two sampling methods consisting of CBSO and ProWSyn. The propose approach is called cluster-based method (CBM) because CBSO and ProWSyn incorporate cluster algorithms to produce synthetic data. Questions 1 and 2 are addressed in chapter 4.

3. Can the Wasserstein generative adversarial network (WGAN) frameworks be utilised to generate similar sensor data of human activity?

4. Can new sensor data, generated from WGAN networks, enhance recognition performance?

The Generative Adversarial Networks (GANs) are type of method based on deep learning which have been most successful for image generation. To answer the third and fourth questions, we consider using an improved variation of GAN, namely the Wasserstein Generative Adversarial Network (WGAN5) to propose the fourth method for raw human activity data generation, and WGAN has been demonstrated to enhance stability when training generator and discriminator networks. Questions 3 and 4 are answered in chapter 5. Next, the thesis contributions are introduced in the following section.

1.5 Contributions

To date, there are several studies such as [58], [4], [1] and [25] that highlight that the class imbalance in human activity recognition should be investigated. This thesis fills the gap by offering a comprehensive assessment of the benefit of using sampling methods to increase the performance of human activity recognition models when an imbalanced human activity dataset is used for training. The sampling methods in this thesis are divided into two categories based on the intrinsic properties of sampling methods in order to introduce the four proposed methods. The first category is where sampling methods are based on shallow machine learning algorithms, and the second category is a sampling method based on deep learning algorithms.

We implement and compare six different sampling methods based on shallow machine learning algorithms in order to handle the class imbalance challenge in the human activity dataset in order to determine their applicability for improving the performance of a supervised model. These methods include Synthetic Minority Over-sampling Technique (SMOTE), Random_SMOTE algorithm, Smote with Tomek links (SMOTE_Tomeklinks), Modified Synthetic Minority Over-Sampling Technique (MSMOTE), Cluster-Based Synthetic Oversampling algorithm (CBSO), and Proximity Weighted Synthetic Oversampling Technique (ProWSyn).

We demonstrate that using these existing sampling methods to introduce three different combined sampling approaches for sensor features generation to handle the class imbalance in human activity is more useful than implementing a single sampling method. The three proposed sampling approaches are named the DBM, NDBM and CBM.

We also compare the performance of these proposed and existing sampling methods using several shallow machine learning algorithms such as the K-nearest neighbours, Logistic regression, Random forest and Support vector machine, and also a deep learning algorithm, the Multilayer perceptron.

Finally, we investigate and assess the potential of using the Wasserstein Generative Adversarial Network (WGAN) that is based on deep learning, to introduce the fourth sampling method to generate raw synthetic human activity data using raw sensor data. We compare the performance of this proposed sampling method in improving learning from imbalanced human activity data by using both Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). We demonstrate that the WGAN only works when we used large human activity datasets. It is a promising approach when used as a sampling method to deal with the class imbalance problem as it enhanced the performance of CNN in human activity recognition. It is hard to use the WGAN when the training data of human activity is small since large quantity of training data is required.

1.6 Outline of the Thesis

Chapter 2 highlights particular aspects of human activity recognition (HAR) systems, including types of human activities, examples of HAR applications, the pipeline of developing a HAR system, and an overview of basic concepts and algorithms in machine learning. It then focuses on various already implemented in HAR systems relevant to our research.

Chapter 3 provides a review of the class imbalance problem and reviews the solutions proposed by other studies in order to deal with this problem. The chapter is outlined three solutions, including data level solutions that implement several methods. For example, sampling methods to balance the dataset. This chapter also shows a short introduction of the algorithmic level approach that adjust the learning algorithm in order to handle the class imbalance issue. The other solutions are then reviewed, which work by combining undersampling and oversampling to improve a supervised model's performance. This chapter follows to presents the Generative Adversarial Networks (GAN) and how it can solve the class imbalance problem. Finally, the chapter highlights studies that observe and resolve the challenge of class imbalance in the domain of human activity recognition.

Chapter 4 introduces the experiment and obtain results of the proposed DBM, NDBM and CBM and compares six different sampling methods to overcome the problem of class imbalance in human activity recognition. These six sampling methods are Synthetic Minority Over-sampling Technique (SMOTE), Random_SMOTE algorithm, Smote with Tomek links (SMOTE_Tomeklinks), Modified Synthetic Minority Over-Sampling Technique (MSMOTE), Cluster-Based Synthetic Oversampling algorithm (CBSO), and Proximity Weighted Synthetic Oversampling Technique (ProWSyn).

Chapter 5 presents the experiment and achieve results of the proposed Wasserstein Generative Adversarial Networks (WGAN) to generate synthetic human activity data in order to deal with the problem of class imbalance in human activity recognition. It also presents the attain results when comparing the proposed DBM, NDBM and CBM to WGAN method.

Chapter 6 discusses the findings, their limitations, and their relevance to other studies in HAR field and provides plans for future work.

Chapter 2

Human Activity Recognition Overview

2.1 Introduction

This chapter provides context to the process of human activity recognition that is essential to this thesis. Section 2.2 discusses applications of human activity recognition. In section 2.3, the type of human activity is defined. Section 2.4 provides a background of the process of developing a human activity system that comprises data collection, pre-processing, segmentation, feature extraction, and classification.

Section 2.5 gives an overview of the main types of learning algorithm, including shallow algorithms such as the K-nearest neighbours (KNN), Logistic regression (LR), Random forest (RF) and Support vector machine (SVM), and deep learning algorithms including the Multilayer perceptron (MLP), convolutional neural network (CNN) and Long Short-Term Memory (LSTM). Moreover, it presents the evaluation methods which have been adopted for human activity recognition. This section also reviews works where researchers have used shallow algorithms and deep learning methods for human activity recognition. Finally, the chapter is summarised in section 2.6.

2.2 Human Activity Recognition Applications

Human Activity Recognition (HAR) automatically identifies what an individual is doing from sensor data. Wearable sensors are widely used to recognise human activity in various situations [25]. HAR systems' capabilities to estimate the movement of an individual lead to develop a broad variety of applications [59]. For example, falling can cause serious injury, especially for the elderly. Health-related applications such as fall detection have been widely used to monitor individuals at higher risk of falling [60]. HAR model applied using senor data can automatically send an alert for help if the subject has fallen [61].

Other researchers showed that clinicians and practitioners could adopt activity recognition approaches to assess patient's physical activity and provide promptly recommendations to help physical well-being [62]. Analysing movement information about subject's *gait* and *walking* is useful because it disclose sign and indications of Parkinson's disease [63]. Early on indications of neurodegenerative movement disorders such as Parkinson's disease can be distinguished by analysing movement information about subjects'*gait* and *walking* [63].

The type of activities often determines a human activity recognition application to identify because they might influence the way applications are created and applied [64]. Consequently, the categorisation of activities in types can help one to select suitable methods to identify human activities [1]. The following section will introduce the types of human activities.

Table 2.1: Human activity according to the type and repetition

Activity	Type	Repetition	Example
Ambulation	Static	Less-repetitive	*standing*, *walking*, *lying*
	Dynamic	More-repetitive	*Running*
Transportation	Static	Less-repetitive	*Riding a bus*
	Dynamic	More-repetitive	*Cycling*
Daily Activity	Gestures	Less-repetitive	*Wave hands*
	Hand-to-mouth gestures (HMG)	Less-repetitive	*Eating, drinking* and *brushing teeth*

2.3 Human Activity

The human activity model's performance relies on the type of activities we identify, because certain activities influence how models are constructed and applied [6]. By categorising activities, we can simplify the process of choosing suitable approaches to develop a human activity system, for instance, selecting classification methods [64]. It is important to highlight that in the literature these terms and categories are vary. A common way in the literature to categorize the activity is to group them according to the type of activity and repetition/periodicity [64].

Table 2.1 illustrates the categorisation of activities according to their type and repetition. There are mainly three groups of human activity: ambulation, daily and transportation activities [6], [8] and [64].

First, ambulation activity that is performed in longer durations which comes in two difference forms: static (less-repetitive) such as *standing*, or dynamic (more-repetitive), for example, *running*. Shoaib et al. [65] described these activities as simple activity. Second, the transportation activity which can be static such as a person *riding a bus*, or dynamic, where a person is *cycling*. Third, the daily activity that might consist of hand gestures such as *waving hands* or hand-to-mouth gestures (HMG), for example, *eating* or *drinking* [8]. Daily activities are not as repetitive as ambulate dynamic activities, and these daily activities often are concurrent with each other due to their similar gestures such as *eating*, *drinking*, and *brushing teeth* [6]. These confounded activities often called complex activities [65].

Once one identifies the type of activities, one can start to design a human activity recognition system. Standard steps are often applied to develop a human activity recognition system presents in the following section.

2.4 Human Activity Recognition Pipeline

2.4.1 Introduction

Human activity recognition (HAR) intends to provide information on human physical activity and identify human behavioural patterns by using sensor data. The availability of wearable sensors in devices that can record subjects' behaviours, such as, smartphone and smartwatch, has allowed the advancement of an array of applications as we mentioned earlier such as healthcare and physical well-being [2]. It allows computer systems to monitor, analyse, and assist individuals' daily life to improve their quality of life [25]. There are typical steps that need to be taken in order to build a human activity system, also known as Activity Recognition Chain (ARC) [1] (Figure 2.1 illustrates the ARC process). The terms (steps, phases, and stages) are frequently used interchangeably in the literature to descript the process of ARC. These phases are data collection, pre-processing, segmentation, feature extraction and classification of human activity.

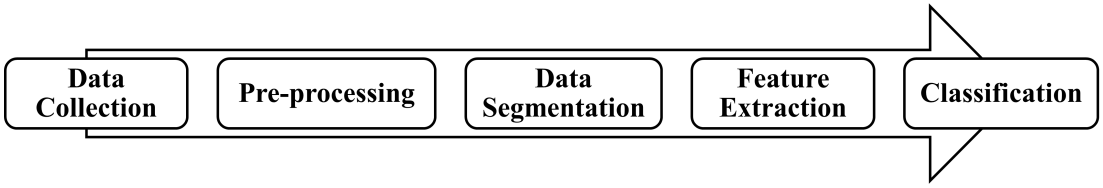

Figure 2.1: The process of developing human activity recognition system [1]

The first stage is data collection, which acquires activity data from sensors that are worn by subjects [1]. Secondly, is the pre-processing stage, which consists of tasks for example, noise removal from the dataset [5].

The third stage consists of data segmentation. Sensor data is a time series where the data is collected and arranged chronologically [5]. Here, instead of evaluating each single data point, the sensor data is split into pieces/segments (windows) where each window has a corresponding activity (class) [21]. In addition, by dividing a continuous stream of data it can help to identify useful information from each window [1]. Consequently, features will be extracted from data in each window.

The fourth stage is feature extraction. In this stage, data will be characterised in an appropriate form to differentiate different classes (activities) from each segment [1]. Time-domain, and frequency-domain features, which are the most common approaches, can be extracted from sensor data (window) [5]. Then, these extracted features are utilised as input for the recognition systems to determine which class (activity) belongs to the data [1].

After extracting certain features, the final step is the classification stage, where machine learning techniques are applied to construct a classifier that recognises human activity. There are two approaches based on machine learning techniques that have been utilised in human

activity recognition, shallow algorithms (e.g. k-Nearest Neighbours and Decision Tree) and deep algorithms (e.g. Convolutional Neural Network and Recurrent Neural Networks) [4].

The next sections explain in detail; the process of developing a human activity system that includes data collection, pre-processing, segmentation, feature extraction, classification.

2.4.2 Data Collection

In the first stage, the wearable sensors (body-worn sensors) are attached to different body locations to obtain raw sensor data in human activity recognition. They can effectively capture body movements such as gestures, movement, and location [25]. Wearable sensors often include inertial measurement units (IMUs), they are worn by users by integrating the sensors into smartphones, and smartwatches, and embedded with sensors (e.g. accelerometer and gyroscope) [5].

Lara et al. [6] indicated three forms of data acquisition used in human activity recognition according to the level of naturalness, including natural, semi natural, and laboratory settings.

The natural setting is where subjects operate their everyday activities, generally with no interference to their behaviour by the application.

In semi natural settings subjects carry out their everyday activities as usual. However, they can be required to alter their behaviour in modest ways and perform specific activities from the experiments; specifically, the subject ensures that all research activities have been performed.

The last setting is within a laboratory. The subjects perform particular organised activities in a controlled environment and steps to perform activity are previously pre-planned.

The number and diversity of individuals are essential to consider when collecting human activity data. Both the number of subjects present in the data and the data's diversity can lead to a higher generalisation of human activity models. The study of human activity recognition often does not rely on a specific number of subjects, but previous studies indicated the number often did not exceeded 49 [5].

In short, to obtain reliable results, it is imperative to have a substantial number of subjects in order to evaluate the human activity model [5]. It is also essential to consider diverse subjects when acquiring data such as age, health, height, weight, and other factors [1]. In addition, subject selection can always be restricted based on applications to ensure representativity.

Data annotation is a fundamental step during data collection to label data into different activities. Different methods are available to annotate sensor data. For example, by using custom software developed to synchronise data and annotate them from video footage [66]. Another approach is to use a software which allow individuals to label the recorded data in real-time. For instance, DataLogger was a software that was developed to enable individuals

to annotate their performed activities instantaneously [67].

Data acquired from inertial sensors is a type of time series data, for instance, in smartphones are recorded sequentially at a specific sampling rate [5]. Wang et al. [3] described a sensor reading as:

s = (d_1, d_2, ..., d_t, ..., d_n), where d_t is a sensor reading at time t [3].

The accelerometer is the most common sensor that calculates the acceleration along the x, y, and z axes of a moving or vibrating body, and it is an essential element in identifying different types of human movement [2]. A linear acceleration is often used to measure the acceleration impact of the sensor movement, by eliminating the effect of gravitational force on the sensor [68].

In addition, a gyroscope is a commonly used sensor and it measures the rate of rotation around a specific axis, and it is able to provide orientation information [2].

Many studies carried out to compare and evaluate a human activity system's performance use an individual sensor including an accelerometer, a gyroscope, or a combination (sensors fusion) [42]. Shoaib et al. [69] suggested that in most cases, a fusion of these sensors performs better than the individual in identifying body movements such as ascending and descending stairs. Ward et al. [41] also demonstrated the feasibility of fusing data from microphones and accelerometers. The sensors were mounted to the user's body in a wood workshop. The aim was to recognise activities characterised from not only a hand motion but also an associated sound. In this thesis, we use sensor data from different modalities such as an accelerometer and gyroscope from different body positions for comparison purposes.

One element that should be considered in data acquisition is the frequency of data collection. Thus, the frequency includes related information to the human body's movement [67]. The frequency refers to the total number of sample data gathered every second (Hertz). For example, if the sample rate is 50, then it means for each 1 second we collected 50 data samples.

The position of the wearable device, such as in smartphones or smartwatch attached to an individual's body, is an additional element and significantly impacts the condition of the data gathered as well as the human activity recognition model performance. For instance, data gathered using the smartphone located at an individual's hand provides distinct single readings (patterns) than a smartphone positioned in a individua's waist [70]. Specifically, if data collected from a smartwatch is located at a user's hand, this data can be used by an activity algorithm that aims to recognise activity related to hand movements such as *eating* or *drinking*.

2.4.3 Pre-Processing

The second stage is pre-processing, where various techniques can be applied, such as noise reduction of the sensors and handling missing values [42]. The raw sensor data might consist

of noise produced by anomalies during data collection, such as changes in body acceleration movement. Some study applies noise reduction (filtering) techniques in order to minimise noise. For example, techniques primarily used for sensor data are low-pass Butterworth and Kalman filters [5]. Applying a filter to reduce noise can improve sensor data quality and increase the HAR model's performance [5]. In addition, handling missing values is applied at this stage, using several missing data imputation methods [31].

2.4.4 Data Segmentation

An activity recognition system is applied to identify activities using sensor data and it is a time series containing temporal events when they occur at the time an activity is performed [2]. Therefore, sensor data events are separated into sub-sequences (also known as windows or segment), where each subsequence contain sufficient characteristics of a single activity [71].

In other words, rather than evaluating single data points, the sensor signal data is separated into meaningful smaller windows to contain sufficient characteristics where each window has a corresponding activity. This allows the identification of human activity at a given interval (segment) [21].

In human activity recognition, different windowing approaches are applied such as sliding window, event-defined window, activity-defined window [72]. First, a sliding window (called fixed-sized windows) where sensor data is separated into fixed-length windows [71].

Second, event-defined windows, a pre-processing step is first required in order to locate specific events, and then they are used to determine consecutive data segmentation [71].

Third, activity-defined windows, data segmentation here works by relaying one the detection of activity changes [72]. We use the sliding window approach as it is computationally efficient [2], so the next section elaborates more on this approach.

There are two typical ways to apply a fixed-sized windows approach: overlapping and non-overlapping windows [5]. With non-overlapping, the sensor data is split in a way where each window's values do not intersect with other windows' values. With overlapped windows, the sensor divided data are represented by a percentage that determines the number of data points from the prior window, which intersects the data points from the following window. The main challenge with the sliding window method is to identify optimal window size [71].

2.4.5 Extracting and Selecting Features

The fourth stage is extracting and selecting features. In the context of HAR, feature extraction aims to discover relevant information of the raw sensor data from each window about different movement patterns of subject's physical activities [73]. The input data will be decreased to a reduced set of features representing the different movement characteristics of a user's physical

activities known as the feature vector [1].

Features are generally separated into time-domain as well as frequency-domain features [73]. For instance, using statistical data information of raw sensor data in a window allows us to identify the difference between *"walking"* and *"running"* activity [5].

Time-domain features are typically statistical measures [74]. These Features can be calculated directly from the sensor data in each segment and describe how the sensor data changes with time. Time-domain features are frequently adopted in the human activity recognition model because they are inexpensive to calculate. The mean, median, variance, skewness, and kurtosis are among the time-domain features widely used in activity recognition.

Frequency-domain features are based on frequency analysis is also applied to human activity recognition [74]. In order to calculate frequency-domain features (e.g. Spectral Energy and Information Entropy), each data segment of the signal must be converted into a frequency-domain. The fast Fourier transform (FFT) is the method that used for sensor data transformation [75]. We adopt time-domain features in this thesis as they proved to be efficient and less costly to compute [72].

2.4.6 Activity Classification

The final stage is the activity classification. Numerous machine learning algorithms have been applied for human activity recognition, where most of these algorithms are supervised learning algorithms [25].

Supervised learning algorithms (classification algorithms) learn a mapping function from input data to a class (activity) label. The algorithm discovers a hidden relationship between the input data features or raw human activity data and the class by minimising a defined loss function of the pairs of input data and the corresponding a class/label [2].

There are two groups of classification algorithms [76]. First, the conventional/shallow machine learning algorithms, for instance, Support vector machine (SVM), K-nearest neighbours (KNN) and Random forest (RF), then the second is deep learning algorithms, including convolutional neural network (CNN) and Long Short-Term Memory (LSTM). In the next section, the classification algorithms that are related to the scope of this thesis are briefly reviewed.

2.5 Machine Learning

2.5.1 Introduction

Shallow learning and end-to-end learning are terms often used to describe machine learning [77]. When machine-learning techniques cannot perform feature extraction and classification from training data, they are often called (shallow learning approaches). For example, support vector machine (SVM) or K-nearest neighbours (KNN) are types of the Shallow algorithms [77]. In contrast, deep learning methods are called (end-to-end learning models) because they cannot only perform feature extraction but also classification such as convolutional neural network (CNN) [78].

Machine algorithms are characterised by the category of input utilised for training and its predictable outcome [2]. The most popular forms of machine learning are supervised and unsupervised learning [77]. This section reviews the most regularly applied shallow and deep learning of supervised learning approaches for human activity recognition.

In supervised learning input data are generally comprised of a set of the input data (x) and its class label (y) [77]. The task of the learning algorithm is learning to map the input data to class label [2].

The shallow supervised learning approach includes [2] the K-nearest neighbours (KNN), Logistic regression (LR), Random forest (RF) and Support vector machine (SVM). Multilayer perceptron (MLP), Convolutional Neural Network (CNN) and Long short-term memory (LSTM) are the supervised learning methods based on deep learning [5].

In unsupervised learning where the machine learning method is applied in order to learn from a dataset containing only input data, not including their associated label [45]. There are different tasks to perform based on unsupervised learning, such as data clustering and dimensionality reduction of features [5].

An unsupervised learning method might be applied to discover certain similarities or find a distinctive structure within the input data, such as the clustering method K-means [5].

Dimensionality reduction can be achieved by an unsupervised learning approach, for example, principal component analysis (PCA). PCA is capble to decrease the dimension of features contained in a dataset without losing information [5]. More details on how unsupervised learning methods that have been applied in human activity studies can be found [5].

The next section briefly describes shallow learning methods to allow for a more in-depth exploration of supervised deep learning methods, a process that used more often in this thesis.

2.5.2 Shallow Machine Learning

Among most used shallow supervised learning algorithms in human activity recognition are the K-nearest neighbours (KNN) [79], Logistic regression (LR) [2], Random forest (RF) [80] and Support vector machine (SVM) [81].

KNN is one of the popular methods that can perform classification tasks based on finding similarity measures between data using distance metrics such as Euclidean distance [79].

LR is capable of estimating the probability of samples to determine which sample belongs to a certain class by using a logistic function [2].

RF is a common classification method that is developed using an ensemble of decision trees (DT) [80]. RF operates by building different decision trees at the training phase where a predicted class is selected based on the most often occurring amongst each DT's output.

SVM is another usually applied classification method that seeks to discover the hyperplanes, also known as the decision boundaries, that separates the data into classes [81].

Several previous studies have adopted shallow supervised learning algorithms where they depend on feature extraction methods in human activity recognition, however due to the achievement of deep learning researchers have recently started to learn features using deep learning algorithms [25]. The next section will introduce deep learning in depth.

2.5.3 Deep Learning

Deep learning is a subfield of machine learning algorithms based on artificial neural networks that have been widely used for human activity recognition [5]. We apply the Multilayer perceptron, Convolutional neural network, and Long Short-Term Memory in this thesis. Therefore, the following sections are describing those deep learning algorithms.

2.5.3.1 Multilayer perceptron

The Multilayer perceptron (MLP) is an artificial neural network defined as computational systems that process information via interconnected computational nodes called units (or neurons) [45]. MLP incorporates multiple layers with neurons, particularly an input layer, an output layer and one or more a hidden layer between the input layer and an output layer.

A deep neural network has more than one hidden layer [45]. These layers interact with weighted connections whereby every layer is fully connected to the next following layer. MLP can also be called a feedforward neural network because input data flows in one direction from the input layer toward the output layer [45]. The input layer contains the input value and passes input values to the next layer. The hidden layers will gather the weighted inputs from the input layer and forward their output data to the following layer [45]. Ultimately, the output

layer will encompass the classification results for input data. The output of a single neuron y is represented as [45]:

$$y = \sigma(\sum_{1}^{n} x_i w_i + b) \tag{2.1}$$

Where σ is the activation function (non-linear) defines the output of that neuron given a set of input data [45]. There are popular activation functions which can be applied on hidden layers such as sigmoid, which is a non-linear activation function that outputs a value of between 0 and 1, with an S-shaped distribution curve [4]. Another non-linear function is the hyperbolic tangent or known as (tanh) [77]. Its output distribution spans the range of -1 to +1, whilst the Rectified Linear Unit [4] is also non-linear and permits only positive values to pass through it; in contrast, negative values will be mapped to zero. In the output layer, the commonly use activation function is SoftMax [45], which can output probabilities in cases of class predictions.

The w_i represents weight in the layer, x_i the input data, which can be the output of the prior layer, and b is the bias [45]. Each neuron possesses a unique set of weights and bias.

2.5.3.2 Convolutional Neural Network

A convolutional neural network (CNN) is another common deep learning algorithm that automatically performs feature extraction and classification tasks [82]. CNN has been widely implemented for several applications, such as human activity recognition, object recognition and image recognition [83].

The CNN feature extraction task's main characteristics for human activity is their ability for capturing local dependency as well as scale invariance in human activity data [84]. Local dependency means the CNN can detect local movement patterns within an activity from input data [84]. Scale invariance refers to CNN's ability to recognise activity patterns even when the activity motion varies in some way [84]. For example, a person might *run* with different motion intensity.

A typical CNN model form of a convolutional layer or more, a pooling layer, and a fully-connected layer [82]. A convolutional layer containing a kernel (also known as a filter) is the element for automatically extracting features from the input sensor data by performing a convolution operation of input sensor data with a kernel [84]. The convolutional layer's output is the produced features (also known as a feature map) that will be passed to the next layer [45].

There are two types of the kernel (filter) in a convolutional layer, one-dimensional (1D-CNN) [84], which is often used for time-series data and two-dimensional (2D-CNN) kernel, which is commonly applied on image data [85].

The equation of the 1D-CNN convolution process is described as bellow [86]:

$$z(i) = x(i) * k(i) = \sum_{n=-a}^{a} x(n) * k(i-n) \tag{2.2}$$

Where $x(i)$ is the input data to be convoluted with the kernel $k(i)$ of a size a , produces a new matrix $z(i)$ which is a feature map.

The equation of the 2D-CNN convolution process is defined as following [86]:

$$z(i,j) = x(i,j) * k(i,j) = \sum_{n=-a}^{a} \sum_{m=-b}^{b} x(n,m) * k(i-n, j-m) \tag{2.3}$$

Where $x(i,j)$ represents the input data matrix to be convolved with the kernel matrix $k(i,j)$ of size $a \times b$ result in a new matrix $z(i,j)$ that representing a feature map.

The pooling layer, which commonly follows the convolutional layer, decreases the dimensions of feature maps. Many pooling methods have been implemented, such as max pooling or average pooling [45]. Fully connected layers are the typical feedforward neural network layer and incorporate a non-linear activation function, such as SoftMax [45], which can output probabilities in cases of class predictions.

2.5.3.3 Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) network is a specific type of deep learning network designed for applications of sequence and time-series data [45]. LSTM is useful for time-series data as information from earlier in the time-series might be important to discover pattern within time-series data [87]. The main computation unit of the LSTM is known as a memory cell or the cell [88]. There are two states of the LSTM's cell to store and remove information, including the long-term state c_t (cell's state at time step t) and the short-term state h_t[88].

LSTMs use several internal computational units in the memory cell known as gating [86]. These gate are responsible to control what operation can be performed on the memory cell, including input gate, output gate, and forget gate [83]. Therefore, the LSTM algorithm is suitable for learning temporal dependencies from a sequence and time-series data [83]. The cell enables the LSTM to remember information from earlier in the sequence by learning what information to store, erase and output information in the long-term state c_t and the short-term state h_t [89].

The g_t (equation 2.7) is the main input x_t to the cell which is computed from the input of the current time step t and state of the previous short-term state h_{t-1}.

The first gate is the forget gate that is controlled by f_t (equation 2.5) and decides which parts of previous information will be forgotten from the previous time step c_{t-1}.

The second gate is the input gate which is controlled by i_t (equation 2.4) and determines which new information from the current input x_t should be added in c_t.

The last gate is the output gate, and it is controlled by o_t(equation 2.6) which determines what information to output based on the current input x_t and information in c_t. The process of each cell component is formalised as follows [88]:

$$i_t = \sigma_i(b_i + w_{xi}x_t + W_{hi}h_{t-1}) \tag{2.4}$$

$$f_t = \sigma_f(b_f + w_{xf}x_t + W_{hf}h_{t-1}) \tag{2.5}$$

$$o_t = \sigma_o(b_o + w_{xo}x_t + W_{ho}h_{t-1}) \tag{2.6}$$

$$g_t = tanh(b_g + w_{xg}x_t + W_{hg}h_{t-1}) \tag{2.7}$$

$$c_t = f_t c_{t-1} + i_t g_t \tag{2.8}$$

$$h_t = tanh(c_t)o_t \tag{2.9}$$

Where i_t, f_t, o_t, g_t, c_t represent input gate, forget gate, output gate, main input to the LSTM cell, the long-term and short-term state, respectively [88]. The term x_t is the input to the LSTM cell at time step t. $w_{xi}, w_{xf}, w_{xo}, w_{xg}$ are weights of each cell element for their connection to the input xt, where $w_{hi}, w_{hf}, w_{ho}, w_{hg}$ represent each cell component for their connection to the prior short-term state c_{t-1}. $\sigma_i, \sigma_f, \sigma_o$ are activation functions [88].

A number of evaluating metrics have been adopted in order to measure the recognition performance of shallow and deep learning classification approaches [1]. The next section introduces the evaluation metrics that commonly employed in human activity recognition.

2.5.4 Evaluation Metrics

Several performance metrics have been implemented to evaluate classifier performance in human activity recognition such as F1 score, recall, and precision [1]. These metrics are usually used for binary class classification. In order to use these metrics for multiclass classification cases, there are two approaches one can use, a one versus one (OVO) and a one versus all (OVA) (more details about these approaches in [90]). In this thesis we used OVA because it is the most popular applied approach [90].

Accuracy is often applied to measure classifier performance [35]. However, if a dataset is imbalanced, accuracy is improper since it is skewed towards more represented classes [83]. One approach to overcome this, precision and recall are utilized for each class, and then the weighted mean of these overall classes (the F1 score) [2]. The Precision records the proportion of class predictions that are correct, and the Recall records the proportion of actual class instances that are correct [91].

$$Recall = \frac{true\ positive}{true\ positive + false\ negative} = \frac{TP}{TP + FN} \tag{2.10}$$

$$Precision = \frac{true\ positive}{true\ positive + false\ positive} = \frac{TP}{TP + FP} \tag{2.11}$$

We use an example to explain True Positive, True Negative, False Positive and False Negative [90]. Consider a dataset that incorporates data from two different activities (e.g. *run* and *walk*), and the activity *run* is the activity of interest. A classification algorithm is applied for recognising the activities. True Positive (TP) refers to the case when the actual data of activity *run* correctly recognised as activity *run*, where True Negative (TN) is that actual data of activity *walk* correctly recognised as activity *walk*. The False Positive (FP) is when actual data of activity *walk* incorrectly recognised as class *run*. The False Negative (FN) is when actual data of activity *run* incorrectly recognised as activity *walk*. The balanced F1 score applied here treats classes equally (Macro F1 score), regardless of how frequently a class appears [91]:

$$F1\ score = \frac{1}{m} \sum_{i=1}^{classes} \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{2.12}$$

Another approach to calculating the F1 score is the weighted average [91], where the F1 score is calculated for each label. The average value is computed with each class owning a weight corresponding to the proportion of true samples in each label [83].

2.5.5 Machine Learning for Human Activity Recognition

The study in [25] has indicated that the two common approaches among other methods used to improve an activity recognition models' performance. Rather than exploring and applying sampling methods, comparing different classification models and features extraction from training data are widely investigated.

Akbari et al. [40] compared a deep learning model, which was CNN, with traditional machine learning algorithms for human activity recognition using the Physical Activity Monitoring (PAMAP2) dataset. The dataset including ADL such as *walking, running and vacuum cleaning*. They used accelerometer and gyroscope data from a sensor attached on the wrist. Akbari et al., for this comparison, extracted statistical features including mean and standard deviation and used SVM and KNN classifiers. Their study demonstrated that generally, CNN outperformed the traditional machine learning algorithms that accomplished a 4.5% higher accuracy on average than SVM and KNN models. Akbari et al. argued that the reason for this

might be that the features created by the convolutional layers in deep neural networks were more useful than the hand-crafted features.

The authors in [92], compared how the type of features influence the performance of different classification methods, including RF, KNN and SVM, to recognise daily living activities from the Activity Recognition from a Single Chest-Mounted Accelerometer dataset (ADL), such as *working at a computer, standing*, and *walking*. The extracted features included time-domain (e.g. maximum and minimum), and frequency-domain features such as FFT coefficients and energy from an accelerometer data. They fused time-domain and frequency-domain features to evaluate the combined features' influence on classifier performance. They concluded that an RF classifier with the combined features provided the highest accuracy of 0.88% in recognising the daily living activities.

Erdas et al. [93] also utilized the ADL dataset in order to compare the performance of deep learning algorithms such as CNN and a hybrid based on CNN and LSTM layers to recognize several human activities, such as *working at the computer, standing up, walking* and *going up/down stairs*. They indicated that the applied algorithms showed similar performances as the accuracy score was 0.91% from both CNN and the combining CNN and LSTM layers.

Ordóñez et al. [83] proposed a hybrid (DeepConvLSTM) approach by combining CNN and LSTM layers to improve the activity recognition. They used the opportunity dataset including using multimodal wearable sensors (e.g. accelerometer and gyroscope). They showed that the CNN could discover the key features from the sensors data automatically, whereas the LSTM identify temporal patterns from these features. The stated that the proposed hybrid approach recognised complex daily activities, such as *open drawer* and *close drawer* with a high F1 score of 0.93%.

Antar et al. [94] and Jeyakumaret et al. [89] used a part of the original large scale Sussex-Huawei Locomotion (SHL) dataset in order to recognise locomotion as well as transportation activities. The activites are including *still, walk, run, bike, car, bus, train*, and *subway*.

Antar et al. [94] used several time-domain features, including mean, variance, standard deviation, maximum and others to compare performances of different classifiers such as KNN, SVM, and RF. They also used data from several sensors for instance an accelerometer, gyroscope, magnetometer, and linear acceleration. They showed that the accuracy of KNN was 0.91%, SVM was 0.87%, and RF the best of the classifiers, was 0.92%.

Jeyakumar et al. [89] compared various traditional and deep learning algorithms using data from different sensors including an accelerometer, gyroscope, and magnetometer to recognise different types of locomotion in addition to transportation activities. They also extracted and fused 18 time-domain features along with 5 frequency-domain features to use for training an MLP classifier. They reported that the MLP classifier achieved a F1 score of 0.93%.

Shoaib et al. [17] collected the smoking activities dataset and introduced results on both two-layered as well as single layer classifiers, including SVM, RF, and Decision Tree (DT) for smoking activity detection. They driven time-domain features in order classify the sensor data. In this case, the F1 score fell between 0.83-0.94% when utilizing two-layer classifiers to detect smoking activities. Nevertheless, Shoaib et al. also indicated recognising concurrent activities such as *smoking* and *drinking* was challenging.

We also explored the influence of different data pre-processing methods that impact the CNN classification performance of smoking activity with time-domain hand-crafted features as well as raw sensor data utilizing the smoking activity dataset collected by Shoaib et al. [17]. The CNN model with raw input was better to differentiate confound activities (e.g. *smoking* and *drinking*) compared with using manually hand-crafted features as an input to the proposed CNN model. We show a version of this work in the appendix A.

2.6 Summary

This chapter covered some applications which used in human activity recognition and offered the background required to contextualise the problem of human activity recognition. For example, the chapter showed the type of human activity, including ambulation, daily and transportation activities. It included an explanation of the common steps for developing HAR systems such as data collection, pre-processing, segmentation, feature extraction, and human activity classification as well as evaluation.

This chapter also discussed the essential aspect of machine learning, including comparing the main types of learning algorithms, namely supervised and unsupervised. Furthermore, the shallow algorithms KNN, LR, RF, SVM and deep learning approaches such as MLP, CNN and LSTM were explained. The chapter finally provided a comparison of some studies where researchers applied shallow algorithms and deep learning algorithms for human activity recognition.

Combined, these studies suggest that CNN and LSTM are useful methods for recognising human activity compared with shallow methods, such as SVM. This is because convolutional and recurrent layers can act as feature extractors [4]. Consequently, CNN and LSTM were used to develop WGAN. They were also used to design activity recognition models to evaluate the WGAN networks. Further explanation is presented in section 5.2.2. KNN, LR, RF, SVM, and MLP are widely implemented for human activity recognition [76] [2]. This thesis decided to use these to evaluate how the DBM, NDBM and CBM may solve the problem of class imbalance.

The next chapter will define and discuss the main aspects related to the class imbalanced problem.

Chapter 3

Class Imbalance

This chapter presents the class imbalance problem and discusses existing researchers' proposed solutions that are in the public domain and of particular relevance to the field of human activity recognition. Section 3.1 presents a brief overview of the problem of class imbalance. Then, the main standard solutions for class imbalance, including data level methods, are review in section 3.2, and follow with a brief review of the algorithm level solution in section 3.3. Furthermore, Generative Adversarial Networks (GAN) are introduced in section 3.4.

As it is relevant to this thesis, this chapter reviews the work carried out by earlier researchers on tackling the class imbalance problem using sampling approaches in human activity recognition in section 3.5. Finally, the chapter is summarised in section 3.6.

3.1 Introduction

In machine learning, classification tasks are extensively employed in multiple applications. [88]. The class imbalance problem is among the most common challenges encountered when performing classification tasks [95]. A dataset is considered imbalanced if the number of samples in the dataset for one or more classes (majorities) significantly outnumbers the other classes (minorities) [96]. Consequently, a classification algorithm can become severely biased toward the majority class [35].

There are many different applications where the class imbalance issue presents, such as medical diagnosis and fraud detection [95]. In most cases, the class of concern in medical diagnosis [97] and fraud detection [95] probably might be the minority class. If a classification model misclassifies the underrepresented class, the cost of that is risky. For example, if a fraud detection system is biased toward the most represented class, it might not be able to identify a fraud transaction [27]. This thesis focuses on studying the problem of class imbalance in human activity recognition. More details about this problem in other domain can be found in [35].

In addition, there are other issues related to the class imbalance that might present in imbalanced data, for example, small sample size, class overlapping, and within-class imbalance which also can deteriorate a supervised model's performance [27].

The small sample size refers to a lack of samples in a particular class or classes [95]. When there is lack of training samples, a challenge arises in the classification task. Namely, the classification ability to generalise is degraded as there are not enough samples to discover these classes' underlying pattern.

Class overlapping, also called class separability, refers to the degree of separability among classes in the data [27]. When the input features of samples (data points) from different classes share similar characteristics in the feature space, it becomes challenging to define discriminative rules to separate the classes [98]. Alternatively, the class overlapping presents in imbalanced data, which might cause stronger classifier performance degradation [35].

The within-class imbalance and (also called a small disjunct) appear when some classes might consist of several sub-clusters of different amounts of samples [99]. When a classifier is trained with data that is imbalanced and contains the within-class issue that might hinder the classifier performance, as it might too suffer from within-class imbalance [36].

There are two approaches to solving the class imbalance issue: data level (also known as sampling or external approach) and algorithm (also called internal approach) level techniques [95].

Data level techniques aim to solve the problem by manipulating (balancing or rebalancing) a training set's class distribution using sampling methods, including oversampling, undersampling, and combined sampling methods (using oversampling and undersampling

techniques) [95]. Specifically, balancing or rebalancing classes refer to applying sampling methods in order to enlarge the frequency of the minority class or decrease the majority class's frequency in a dataset [38]. In addition, the number of samples to oversample or undersample for each class is often empirically selected [96].

Algorithm level methods adjust existing learning algorithms to focus more on the minority classes[100]. Both methods are capable of decreasing the degree of class imbalance [38].

It is important to mention that we will focus on the class imbalance problem by using data level methods. The reason is that they are not only useful and less complex to configure but also can be integrated with any learning algorithm [27].

Recently, several data generation approaches have emerged based on deep neural networks. The generative adversarial network (GAN) is one of these methods that has attracted much interest in many applications such as generating images [101]. GAN is learned by competition between two neural network models. The two neural networks are known as the generator and the discriminator. The generator network during the learning process is used to produce new data samples by capturing the distribution of the real data, and the discriminator network is employed to distinguish whether data samples are real or synthetic. Section 3.4 provide more details about the GAN method.

The next section describes different data level methods in order to deal with the issue of class imbalance.

3.2 Data Level Solution

This section introduce several data level methods that are often implemented for the issue of class imbalance. We first describe the random oversampling and random undersampling methods and follow to categorise the sampling methods into three sections based on their intrinsic properties in order to produce synthetic data, which is then used to rebalance a dataset.

Sampling methods are a straightforward approach to tackle the issue of a class imbalance [102]. These methods are applied at the pre-processing stage to rebalancing the dataset [35]. The original class frequencies (the number of samples for each class) is modified [38]. Consequently, a machine learning classifier's ability to identify the decision boundary between the most represented class and the unrepresented class enhance, then their generalisation ability improves [27] and [95].

The first category of sampling methods is the distance-based sampling approaches: Synthetic Minority Over-sampling Technique (SMOTE) [52] and Random_SMOTE algorithm [53]. These methods mainly use K nearest neighbours in the process of oversample a dataset.

The second category is noise detection-based methods, including SMOTE with Tomek links (SMOTE_Tomeklinks) algorithm [54] and Modified Synthetic Minority Over-sampling Technique (MSMOTE) algorithm [55]. These algorithms include mechanisms to identify noisy instances in a dataset which reduce the risk of introducing artificially noisy instances.

The last is a cluster-based sampling algorithm, which includes the cluster-Based Synthetic Oversampling (CBSO) algorithm [56] and Proximity Weighted Synthetic Oversampling Technique (ProWSyn) [57]. Here, the clustering approaches are included in the process of oversampling a dataset.

3.2.1 Random Oversampling and Random Undersampling

The random oversampling method randomly generates replicate data of the minority class in order to rebalance the distribution of a dataset [103]. Consider a training set comprising 10 samples from the minority class and 50 samples from the majority class. This sampling method each time will randomly select a sample from the minority class and duplicate it until the number of minority samples become equal to the number of majority samples. Then, the training set would contain 100 samples (50 majority class samples and 50 minority class samples). It has been shown that this method leads to overfitting [95].

Overfitting refers to a situation where a machine learning model perfectly shows good performance on the training data but lacks generalisation ability to unseen data [77].

The random undersample is another sampling approach that decreases the number of the majority samples by randomly removing majority class samples to rebalance the dataset distribution [95]. Given a training set comprising 50 samples from the minority class and 200

samples from the majority class. In this sampling method, 150 samples from the majority class are selected randomly and eliminated in order to generate a rebalanced class distribution. The training set will then contain 100 samples that include 50 samples from the majority class and 50 samples from the minority class.

Previous research has established that the random undersampling approach disadvantage can eliminate valuable information from the data [27].

3.2.2 Distance-Based Methods

SMOTE and Random_SMOTE methods principally use K-Nearest Neighbours (KNN) in the process of oversampling. In the training dataset, SMOTE [52] takes an instance of the minority class x, and then computes its K nearest neighbours, identified as the lowest Euclidean distance between itself and other minority instances. In order to produce a synthetic sample of x, SMOTE randomly selects the nearest neighbours from the minority class, let's say y, and computes the difference of $y - x$. The new synthetic sample, x_{new}, is computed by multiplying a random number between 0 and 1 using the below equation [52]:

$$x_{new} = x + (y - x) \times rand\,(0, 1) \tag{3.1}$$

The new synthetic instance x_{new} will be an instance along the line between x and y.

Random_SMOTE [53] is an oversampling algorithm that enlarges the decision regions. To generate a synthetic sample, the first step is that each sample (data point) x of the minority class, two samples y_1 and y_2 are randomly chosen from the minority class. Therefore, a triangle is formed with x, y_1 and y_2. Then, it generates temporally synthetic sample t_i, using equation 3.2, along the line between the two selected y_1 and sample y_2 of the minority class:

$$t_i = y_1 + (y_2 - y_1) \times rand\,(0, 1) \tag{3.2}$$

A synthetic sample x_{new} is then created along the line between the temporally sample t_i and sample x using equation 3.3 [53].

$$x_{new} = x + rand\,(0, 1) \times (t_i - x) \tag{3.3}$$

3.2.3 Noise Detection-Based Methods

Real world data often contain noise [104]. Frenay et al. [105] described noise as inconsistencies that obscures the relationship between the features of a sample and its label. The present of noise in data can negatively affect it quality and then might hinder the performance of a learning algorithm [38]. Noise can be produced by different reasons such as faulty sensors

or other sources [106]. Frenay et al [105] also indicated that class noise (also known as label noise) is one of the most harmful noises in machine learning. Class noise refers to altering labels that is assigned to samples [105]. For instance, by incorrectly assigning the label of a minority class sample to the majority class label [27].

SMOTE-Tomek Link [54] and modified synthetic minority oversampling technique (MSMOTE) algorithms integrate processes to detect noisy samples in a dataset in order to minimize the risk of creating synthetically noisy samples [55].

SMOTE-Tomek Link is a sampling approach that performs oversampling by using SMOTE algorithm and integrating the Tomek link concept as a step of data cleaning [54].

We explain the idea of Tomek link as follows [107]. Consider two samples x and y belonging of opposite classes, where $d(x, y)$ is the Euclidean distance between x and y. A (x, y) pair ,which are each other's nearest neighbours, is identified as a Tomek link, if there is no sample z meets the following conditions, $d(x, z) < d(x, y)$ or $d(y, z) < d(x, y)$.

In other words [38], the sample x and y will form a Tomek Link in the following cases. First, sample x nearest neighbour is y. Second, sample y nearest neighbour is x. Finally, samples x as well as y belong to opposite classes. This definition show that samples that are in Tomek Links are boundary samples or noisy samples. According to Saez et al. [98] borderline or boundary samples defined as samples that are close to the decision boundary between the minority class and the majority class or lying in the region which surrounding the class boundaries where the classes overlap. Hoens et al. [29] indicated that this is because not only boundary samples, but also noisy samples will be closer neighbours, who are belong different class.

SMOTE-Tomek Link produces synthetic data in two steps [54]. First, the original minority training data is oversampled by using SMOTE method. Second, Tomek links are recognized in the training data and removed to produce a rebalanced data set.

The MSMOTE is an improved algorithm of SMOTE [55]. In this algorithm, there are two steps to be made in order to produce synthetic samples [55]. First, the samples of the minority class are assigned into three types, safe, border and noise samples, which are based on distances among all samples using the KNN classification algorithm. In order to decide the type of samples, that is if the sample' label is in the minority class is the same as the labels of its k near neighbours, then the sample is a member of the safe samples [55]. If their labels are entirely different, then the sample identifies as a noise. Finally, if the sample is neither a safe sample nor noise, then it recognises as a border sample. For example, the sample k near neighbours' labels have both majority class label and minority class label.

The second step is to produce a new sample. The MSMOTE uses a different approach than SMOTE to select the nearest neighbours for producing new samples. In the SMOTE, the nearest neighbours are randomly selected. In contrast, in MSMOTE, the nearest neighbours'

selection is according to the categories assigned to the sample, such as safe.

For samples that are identified as safe, the MSMOTE will pick at random a sample from the K nearest neighbours, whereas samples recognized as border samples, the algorithm only selects the nearest neighbour. Finally for the samples where they are labelled as noise, the MSMOTE disregards them.

3.2.4 Cluster-Based Methods

The cluster-based sampling methods include Cluster-Based Synthetic Oversampling (CBSO) and Proximity Weighted Synthetic Oversampling Technique (ProWSyn).

CBSO integrate clustering technique with SMOTE algorithm. CBSO uses agglomerative clustering to first cluster minority class samples with the aim of identifying those minority samples which are close to the majority samples border [108]. CBSO produces samples only in the neighbourhood of minority samples which are close to majority neighbours using equation 3.1. For instance, in order to produce a new sample, CBSO will select a sample x from the minority class and randomly choose a minority sample y from x's cluster using SMOTE to produce a new sample.

ProWSyn is another cluster-based sampling method [57]. This algorithm computes the distance between underrepresented samples and the highly-represented samples in order to assign higher weights to the underrepresented samples.

The purpose of these weights is to assign greater significance to these samples during learning. To identify the importance of underrepresented samples for learning, ProWSyn operates in two phases. The first phase splits the underrepresented data into a number of partitions (P) according to their distance from the class boundary. The ProWSyn assigns a proximity level to each partition. The level increases if the distance from the boundary is increased. When Underrepresented class samples are assigned with lower proximity levels, then they are considered more important for learning because they are close to boundary. However, in cases where they are assigned higher proximity levels then they are considered less important [57].

To do this from each represented sample y, ProWSyn identifies the nearest K underrepresented samples based on Euclidean distance, and then the set of these underrepresented samples create a partition P_1 (of proximity level L_1). Level L_1 is the nearest level of samples from the classes boundary.

Then, ProWSyn identifies the following K underrepresented samples using the distance from each represented class sample in order for these underrepresented samples to form together the second partition, P_2 (of proximity level L_2). The procedure is repeated in order to partition the underrepresented samples sequentially.

In the second phase, ProWSyn creates synthetic samples of the underrepresented class samples by utilising proximity information to ensure that it only produces a new sample in

regions close to the boundary. To perform this, ProWSyn finds the partition P_1 of underrepresented sample x and randomly choose another underrepresented sample y from P_1 partition. Then it generates a synthetic sample x_{new}, according to equation 3.1.

An alternative approach used for class imbalance challenge is at the algorithmic level. Here a learning algorithm is altered to deal with the issue. The algorithmic level approaches briefly introduces in the coming section because this thesis aims to focus on data level methods.

3.3 Algorithm Level Solution

Instead of altering the dataset distribution, algorithm level methods (known as internal approaches) for handling class imbalance modify existing learning methods or decision process of algorithms in order to alleviate their bias of only focus more on learning from the majority class [95].

In the learning stage of a learning algorithm, unique misclassification costs are given for each class. In other words, the cost of incorrectly classified underrepresented samples is higher compared with the cost of misclassifying represented samples. For more details about cost-sensitive learning see [27].

Generative Adversarial Networks (GANs), a major deep learning technique to generate data [101], is explained in depth in the following section.

3.4 Generative Adversarial Networks

Generative Adversarial Networks (GANs) have largely been used to produce synthetic samples in several applications, such as image synthesis and text generation [109].

GAN is based on the game theory concept of a *minimax game*, where two networks, the generator (G) and the discriminator (D), are trained in an adversarial fashion [101]. The objective of G is to generate synthetic data that D would be unable to differentiate from real data. Contrarily, the aim of D is to distinguish real data from generated synthetic data. Consequently, the objective function of GAN is identified as [110]:

$$\min_G \max_D E_{x\sim P_r}[log(D(x))] + E_{z\sim P_Z}[log(1 - D(G(z))] \tag{3.4}$$

Where x is real data, z is sampled data form random noise, such as Gaussian distribution or uniform distribution [44]. Namely, P_r is the real data distribution and P_z is the generated data distribution.

Kullback-Leibler (KL) divergence and Jensen-Shannon (JS) divergence [48] are significant probability measurement metrics that GAN uses when the discriminator is optimised. These metrics estimate the distribution distance between the real samples and the produced samples. Mode collapse [48] is the problem that constrains the capability of the generator model, which occurs by only allowing the generator models to generate a partial range of samples from the original data distribution. GAN suffers from mode collapse, and this potential limitation leads to learning instability. Arjovsky et al. [44] showed that a possible source of mode collapse is from the use of KL in GAN training.

To overcome mode collapse, Arjovsky et al. [44] proposed the Wasserstein GAN (WGAN), which uses the Wasserstein distance instead of KL to measure the distance between the original sample and the created sample. WGAN enhances the stability of learning and overcomes the difficulty of mode collapse. A Wasserstein distance is defined as [44]:

$$W(P_r, P_z) = \inf_{\gamma\in\prod(P_r,P_z)} E_{(x,y)\sim\gamma}[||x - y||] \tag{3.5}$$

Where $\prod(P_r, P_z)$ represents the set of all joint distributions $\gamma(x, y)$, the distance to transform the distribution P_r into the distribution P_z is represented by $\gamma(x, y)$ [44].

Due to the fact that the Wasserstein distance equation is intractable in practice, Arjovsky et al. transformed it into a more computable form by using to the Kantorovich Rubinstein duality [44]. Then, from the Wasserstein distance they derive the WGAN objective function as:

$$\min_G \max_{D\in L} \underset{x\sim P_r}{E}[D(x)] - E_{z\sim P_Z}[D(G(z))] \tag{3.6}$$

In order to meet the duality requirement, Arjovsky et al apply the Lipschitz constraint on

the discriminator, which is also called the critic in WGAN, they suggest implementing the parameter to clip the weights of the discriminator. The weights of the discriminator have to be within a specific range $[-c, c]$, where c controls hyperparameters [44].

A major variance between WGAN and original GAN is the role of D [44]. The D purpose in GAN is applied as a binary classifier, which differentiates between real and generated samples. However, the function of D in WGAN is to estimate the Wasserstein distance between the generated, and the actual data distribution, which is a regression task. Hence, in the last layer of D, in the WGAN, the sigmoid function is eliminated.

Having presented the major approaches to deal with class imbalance, next section will discuss the attempts that have been shown in the human activity recognition field in order to further investigate the problem of class imbalance.

3.5 Class Imbalance in Human Activity Data

This section introduces studies that attempted to tackle the issue of class imbalanced and generating synthetic senor data in human activity recognition.

3.5.1 Related Work

Classification algorithms that used to recognise human activity can demonstrate better performances when large labelled training data is used [25]. However, human activity data is often unequally distributed. This issue is identified as the class imbalance [36]. The more represented activities make the majority classes (activities), whereas those underrepresented samples form minority activities [29].

Popular research emphasised that class imbalance in human activity is vital to hinder classification algorithms' generalisation capabilities for human activity recognition [1]. When a dataset is skewed, a classification algorithm might predict the majority class and simply shows great accuracy, whereas the minority classes are highly possible to be misclassified [103].

A recent study by [25] strongly suggested the importance of addressing class imbalance in order to alleviate the problem of activity unequally distributed in human activity recognition. Resolving class imbalance can also help overcome the effect of imbalanced class distribution which enhances the generalisation of classification algorithms [95].

During our research, we have found that there are different ways to improve the problem of class imbalanced. The first is a sampling technique which generated more data samples of underrepresented classes or eliminated some samples of the represented classes. In order to apply a sampling approach to time series data for producing synthetic activity samples such as SMOTE, one is required to extract features from the raw sensor data (e.g. time-domain features) [51]. The sampling methods are based on shallow machine learning algorithms and they cannot operate on time series data [51].

The second technique to solve class imbalance is using a method based on deep learning, such as GAN, which does not require one to derive features for example, mean or standard deviation from the raw sensor data [28]. In a case where GAN is used to produce fake samples, the raw sensor data can be used directly with the GAN method to generate synthetic human activity data.

Only a few studies have adopted sampling methods in human activity recognition. SMOTE has been the most applied approach that is used to mitigate the influence of class imbalance [26].

In [111] a framework was introduced by Ni et al. to recognise different human daily activities, such as static and dynamic activities using different sensor modalities, using data

from accelerometer and gyroscope. The performance of the model dropped as the class (activity) distribution of samples was unbalanced. Therefore, they used sampling techniques such as SMOTE and random undersampling to improve the unbalanced samples problem. The sampling methods were applied to the entire original data. They did not divide the data into partitions, such as the training and testing sets. Based on their experimental result, the sampling technique (SMOTE) successfully enlarged underrepresented human activities samples and improved recognition performance compared to random undersampling as it could discard some important information.

Likewise, several aspects that could influence daily human activity recognition performance, including class imbalance, were investigated by [58]. Data from several sensors (e.g.accelerometer and gyroscope) were utilized in order to identify different activities for instance, *walking, jogging,* and *jumping*. The distribution of human activities was unequal. Therefore, they reported that the classifier always showed a good performance to recognise the majority class, whereas its performance was inadequate for the minority classes. To overcome the problem of class imbalance SMOTE method was applied to increase the underrepresented activities' presence. They concluded that the classifier's performance improved since more training data was obtained.

Despite this, implementing oversampling before splitting a dataset into different partetions such as train and test data can result in information leakage, from the original test data to the newly produced training data. This can lead to overly optimistic classification performance [112]. In other words [113], the performance of a learning algorithm can be similar in both in the train and test data, which cannot be interpreted as the learning algorithm is able to generalise to the test data appropriately. However, it is due to the fact that are similar patterns in both train and test data from the information leakage [112]. Consequently, in this thesis, we avoid implementing the sampling method to the entire original data. Instead, we only apply the sampling methods to the train set. In addition, we apply and compare six sampling approaches including SMOTE, Random_SMOTE, SMOTE_Tomeklinks, MSMOTE ,CBSO , and ProWSyn.

GANs have been largely used to produce synthetic samples in several applications, such as image synthesis and text generation [47]. However, few works have been done to develop GANs models for the aim of producing sensor data [26].

SenseGen [114] was the first effort at using GANs to synthesise sensor data. However, the proposed model trained both the generator and the discriminator separately. Subsequently, during the training process, the generator did not learn from the feedback of the discriminator.

Recently, Wang et al. [28] have developed a model called SensoryGAN for generating sensor data. SensoryGAN models are capable of capturing the distribution of the original sensor data of human activity and enabled them to generate synthetic sensor data. Nevertheless,

Wang et al. indicated that SensoryGAN suffered from instability while training.

In this work, we use an extended variation of GAN, called the Wasserstein Generative Adversarial Network (WGAN), which has been shown to improve stability when training generator and discriminator networks [44]. We aim to generate synthetic sensor data based on the approach of a generative adversarial model and assessing the quality of the artificial sensor data by applying a supervised classifier. This thesis attempts to shed light on utilizing WGAN for producing sensor data.

3.6 Summary

The literature review related to the class imbalance issue was introduced in this chapter to summarise related work and show the importance of tackling this issue for human activity recognition. The chapter also highlighted a significant problem with activity recognition algorithms when classifying imbalanced human activity dataset. These classification algorithms might be biased towards the most represented class that could reduce the performance of these models [4]. The solutions related to class imbalance chapter also reviewed, including data level (i.e. sampling), algorithm level learning, and generative adversarial networks (GANs). In addition, the chapter covered studies where other researchers addressed the class imbalance for human activity recognition.

In summary, human activity recognition technologies' potential contribution might be limited due to the nature of the human activity data problem that is imbalanced and noisy [115]. Many studies on human activity recognition have primarily concentrated on optimizing supervised algorithms in order enhance the performance of activity recognition systems [25] and [28]. However, few attempts have been made to explore the potential of generating sensor data using sampling methods [26]. In order to boost the performance of human activity recognition, further research should examine the efficiency of the data generation methods. Accordingly, throughout, this thesis will explore various sampling methods for sensor data generation.

The next chapter focuses on the problem of class imbalance in human activity recognition as well as presents the proposed solution to overcome it.

Chapter 4

Comparing Sampling Methods to Generate Sensor Features for Human Activity

4.1 Introduction

A large number of human activity recognition research often implemented a supervised model method and they are depend on a large amount of labelled sensor data to train. The availability of a substantial and balanced number of training labelled human activity data is limited because obtaining and labelling sensor data processes are costly as well as time-consuming [25]. As a result, datasets for human activity recognition are regularly not only small but also imbalanced, which produce a major challenge when forming and training a supervised model. When a supervised model is applied to recognise human activity utilising imbalanced sensor data, the performance leans to be biased for learning and recognising the majority class.

Other issues that are related to class imbalance can deteriorate a supervised model's performance such as small sample size, class overlapping, and within-class imbalance.

Some activities are performed less often than others which can cause the small sample sizes for these activities. Therefore, a supervised model often has few data samples to learn a pattern in these activities effectively.

When the sliding window approach is applied to segment the sensor data, class overlapping might occur. The label within a segment is often selected on the majority vote, which may possibly lead to class overlap.

A within-class imbalance often happen because of the intraclass variability. For instance, the same activity might be frequently performed differently by the same individual.

Data level technique's objective is to solve the problem of class imbalance by altering the frequency of the classes distribution of a training set, which consist of oversampling, un-

dersampling, and collective sampling methods (using oversampling and undersampling techniques). Sampling methods can decrease the degree of class imbalance. Therefore, we focused on utilising data level methods, which are useful, not complex to configure and can be integrated with any learning algorithm.

In our research, we consider applying oversampling and undersampling methods capable of considering human activity data issues, such as small sample size, class overlap, and the within-class imbalance. We compared six different sampling methods to evaluate their influence on the imbalanced human activity dataset. The six basic oversampling methods that we used to build our proposed approach include Synthetic Minority Over-sampling Technique (SMOTE), Random_SMOTE algorithm, Smote with Tomek links (SMOTE_Tomeklinks), Modified Synthetic Minority Over-sampling Technique (MSMOTE), Cluster- Synthetic Oversampling algorithm (CBSO) algorithm, and Proximity Weighted Synthetic Oversampling Technique (ProWSyn). Across this chapter, we are going sometimes to refer to these sampling methods as (the six sampling methods).

We also utilised these six sampling methods to construct three different novel hybrid oversampling methods (which combine different sampling approaches) to reduce the impact of class imbalance by generating sensor data features. We show how this approach can enhance the performance of human activity recognition on three public datasets. The three hybrid methods we proposed are as follows.

The first proposed method was developed on using SMOTE and Random_SMOTE algorithms, which we called distance-based method (DBM), to handle the small sample size in the training data. The second, proposed technique was built on using SMOTE_Tomeklinks and MSMOTE algorithms, which we called noise detection noise detection-based method (NDBM) which examines how this proposed method consider the class overlapping problem in the training data during producing synthetic samples. The last proposed method was designed by combining CBSO and ProWSyn algorithms, which we called cluster-based method (CBM). We investigate how this method consider the within-class imbalance issue in the training data while generating synthetic samples. Across this chapter, we will often refer to these proposed sampling methods as (the three proposed sampling methods).

The next section explains the proposed oversampling methods that we applied to build three different, unified oversampling models that decrease the influence of class imbalance and improve the performance of the human activity recognition model.

This thesis also analyses the influence of applied sampling methods in the performance of different machine learning approaches activity recognition, such as Random Forest, K-Nearest Neighbour, Artificial Neural Network, Support Vector Machine, and Logistic Regression.

4.2 Proposed Methods

The main objective is to improve the performance of a learning algorithm from an imbalanced human activity dataset. When imbalanced human activity data is used to train a supervised model, it can exhibit poor performance. This being because a supervised model might be biased to learn from the majority classes and thus, performs poorly in minority classes. In addition, there are related issues that can present in an imbalanced human activity dataset, including small size sample, class overlapping plus within-class imbalance.

Because of this, we introduce three novel hybrid sampling methods that consider the small sizes of samples, class overlapping and within-class imbalance while creating more human activity data samples from the original training data. To do so, we combined different sampling methods to create a unified method that can empower the sampling algorithms' performance and increase the training data's variability and then increase the generalisation ability of a supervised model. Thus, the three novel hybrid sampling methods handle imbalanced training data and generate a variety of data samples and rebalance the training data. As a result, the human activity classification model generalisation might significantly improve. Then, we compared the proposed approaches to applying a single oversampling method directly to the training data.

We called the first method distance- method (DBM), which was on a combination of distance-based methods SMOTE and Random_SMOTE. SMOTE is still the state-of-art sampling method [116]. Therefore, it was important to form a method that combined SMOTE with a similar sampling method in order to examine if they could complement each other and to solve the small size sample issue. Then, the DBM was applied to balance the training data distribution of activities and enhance the human activity classifier's performance.

The second method noise detection-based method (NDBM) was developed on the oversampling and undersampling techniques SMOTE_Tomeklinks and MSMOTE. We explored whether the NDBM could handle overlapping issues within the training data when the algorithm would include the filtering step.

The last method was built by using cluster-based method, including CBSO and ProWSyn and was called cluster-based method (CBM). We investigated how this CBM handled the within-class issues and improved the human activity classifier's performance.

The proposed approaches are shown in Figure 4.1. As Figure 4.1 illustrates, the first step was to divide the sensor data into a training set and test set. Secondly, we selected the sample method from the three proposed approaches DBM, NRBM, and CBM to oversample the training set which increased all classes' presence. It was important to mention that we avoided oversampling the entire dataset before splitting it into train and test partitions to avoid overly optimistic classification performance.

Then, the oversampled training set was passed onto the activity recognition model (clas-

sifier) to train the model by using a training set that did not suffer from class imbalance and instead, increased the classifier's ability to generalise and distinguish different classes (activities). Finally, the test set was used for the final prediction.

The DBM, NDBM and CBM are taken from work in submission to the Multidisciplinary Digital Publishing Institute (MDPI) Sensors journal 2021.[1]

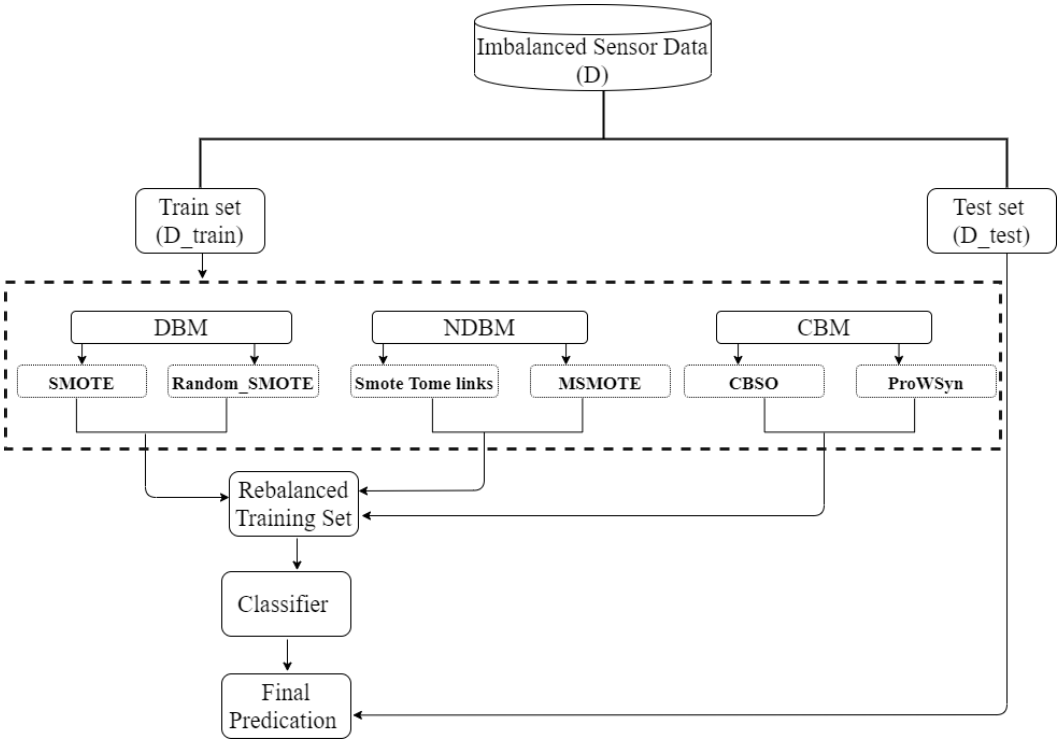

Figure 4.1: The proposed sampling methods. The original dataset is split into train and test sets. A sampling method from the three proposed approaches is then selected to enlarge the training set and mitigate the class (activity) imbalance negative impact. Once the training data is oversampled, it is used to train the recognition model (classifier) to identify different human activities. Finally, the test set is used for the final prediction.

[1] F. Alharbi, L. Ouarbya, and J. A. Ward, *"Comparing Sampling Strategies for Tackling Imbalanced Data in Human Activity Recognition"*, this work is in submission to Multidisciplinary Digital Publishing Institute (MDPI) Sensors journal 2021.

4.3 Experiment Settings

4.3.1 Datasets

We used three datasets that have commonly been utilised within the field, which represent typical tasks for human activity recognition. The reason we selected these datasets because they comprise many individuals performing numerous types of human activities such as complex activities and daily living activities [117]. Daily living is an activity performed daily by a subject [25]. Complex activities are less repetitive and might involve several hand gestures [65]. These datasets also contained continuous sensor readings from a diverse range of wearable sensors such as an inertial measurement unit (IMU) or a smart device (smartphone or smartwatch) worn by individuals in various scenarios at different positions on their bodies. Those sensor readings were recorded by various sensors such as accelerometer, gyroscope, and magnetometer.

The Opportunity dataset was collected from 72 sensors, and there were different types of sensor modalities integrated into the environment, object sensors and body-worn sensors [10]. There were four subjects who performed daily living scenarios by carrying out morning activities in a simulated kitchen environment. The dataset encompassed around 6 hours of recordings and was sampled at 30 Hz. The activities were annotated at different levels: locomotion and gestures. For example, *cleaning up* and *open* door were labelled as gestures activities the locomotion was comprised of posture activities such as *sitting* and *lying*. It is imperative to highlight that we have focused this thesis on using gestures activities because they were often more complexly recognised by a human activity recognition model. Thus, we were able to examine the effectiveness of the applied sampling methods. As Figure 4.2 shows there are 17 imbalanced activities categorised as gestures, and include: *Open Door1, Open Door2, Close Door1, Close Door2, Open Fridge, Close Fridge, Open Dishwasher, Close Dishwasher, Open Drawer1, Close Drawer1, Open Drawer2, Close Drawer2, Open Drawer3, Close Drawer3, Clean Table, Drink from Cup, Toggle Switch*. In addition, the dataset contains several on-body and object sensors, but in this thesis, we utilised only the accelerometer sensor of the IMU attached to the right lower arm.

The Physical Activity Monitoring dataset (PAMAP2) was collected from 9 participants who performed 12 activities for over 10 hours and it was sampled at 100 Hz. Data were recorded by using IMUs placed on the hand, chest and ankle [11]. The dataset included different types of human activities. Figure 4.3 shows the activities distribution, and it can be seen that the dataset is imbalanced. It contains both simple and sporting activities such as *walking, running, cycling, Nordic walking,* and *rope jumping*. It also includes posture activities such as *lying, sitting*, and *standing*. Furthermore, it comprised of activities of daily living (*ascending stairs, descending stairs*), and households activities such as *vacuum cleaning* and *ironing*. We

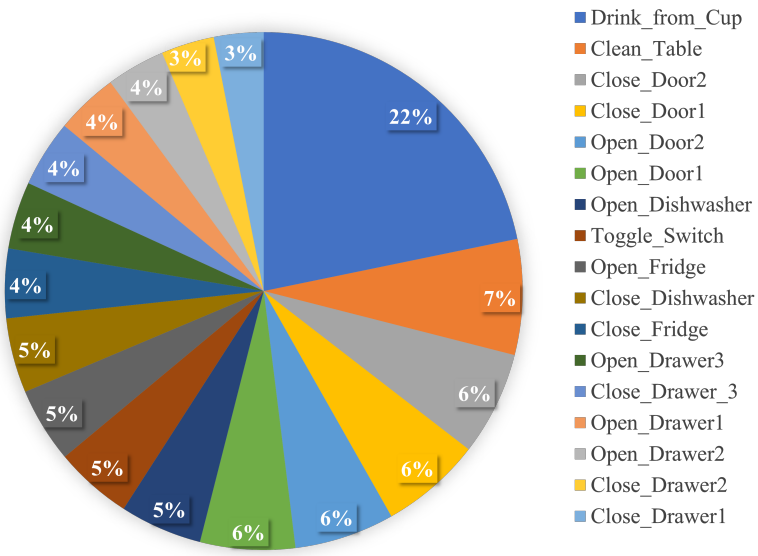

Figure 4.2: Class (activity) distribution of the Opportunity dataset

used only the accelerometer sensor of the IMU placed on the hand.

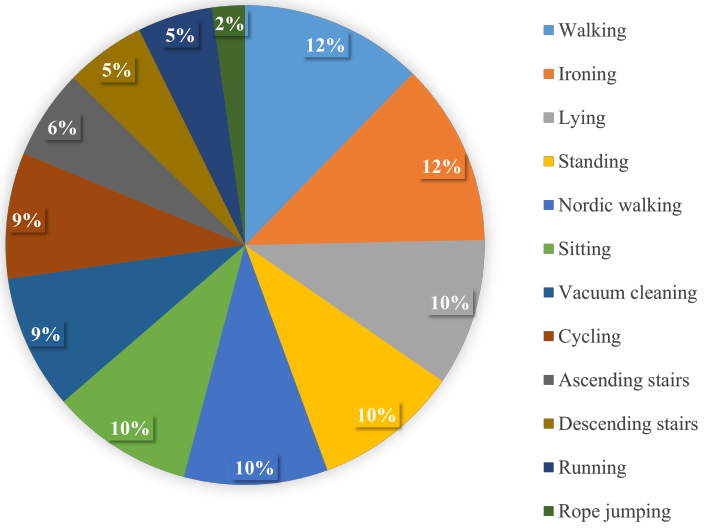

Figure 4.3: Class (activity) distribution of the PAMAP2 dataset

Activity Recognition from a Single Chest-Mounted Accelerometer is a public dataset used in this thesis [12]. Here, dataset was collected data from a wearable accelerometer that was mounted on the chest of 15 participants. The sampling rate of the accelerometer was 52 Hz. The participants performed seven daily living activities, so we refer to this dataset as (ADL). The activities were *Working at Computer (WAC), Standing Up, Walking and Going Up/Downstairs (SWGUDS), Standing, Walking, Going Up/Downstairs (GUDS), Walking and*

Talking with Someone (WATWS), Talking while Standing (TWS). Figure 4.4 shows the activities distribution of the ADL dataset which indicates that the dataset is imbalanced.

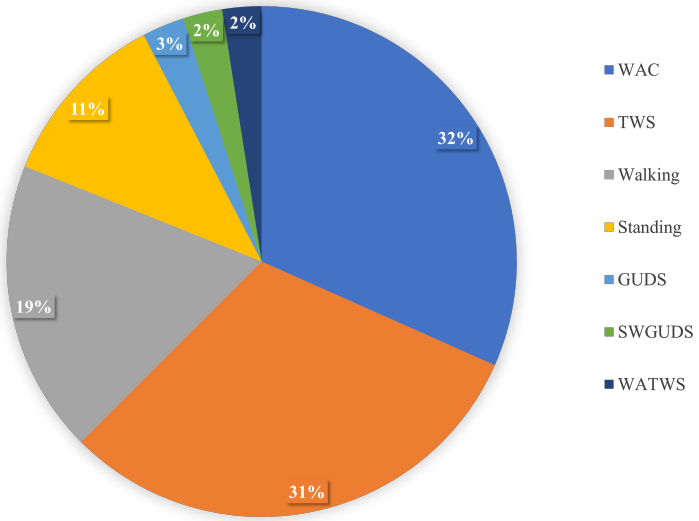

Figure 4.4: Class (activity) distribution of the ADL dataset

4.3.2 Data Pre-processing

As we mentioned in our motivation section, activity recognition models' performance is improved when multiple sensors are used. However, in real-life scenarios, utilising many sensors may not be possible as a subject might not feel comfortable wearing multiple sensors. It is costly as well as time-consuming to collect and label data from several sensors. It is also difficult to achieve the highest performance of a human activity model when using a single sensor. Therefore, we explored and showed how our proposed sampling method could enhance a human activity model's performance in a similar real-life scenario where a single sensor may be available.

First, we calculated the root-sum-squared magnitude ($\sqrt{x^2 + y^2 + z^2}$) for each 3-axis sensor to ensure the data is invariant to the shifting orientation of the sensors [118].

Then, we applied a non-overlapping sliding window to segment the data. Then, we extract six time-domain features because they are effective and inexpensive to calculate, including *mean, standard deviation, minimum, maximum, median*, and *range* [2]

Table 4.1 provides more details such as the number of subjects, sampling rate, the window size, sensor position, and the type of sensor we used.

Table 4.1: Dataset details

Dataset	Number of subjects	Sample Rate	Window size (second)	Sensor position	Number of sensors used
Opportunity	4	32	2	Right Arm	1 accelerometer
PAMAP2	8	100	3	Dominant Wrist	1 accelerometer
ADL	15	52	10	Chest	1 accelerometer

4.3.3 Parameters Setting

We used the gird search to find the parameters for Support vector machine (SVM), Logistic regression (LR), K-Nearest Neighbour (KNN), and SVM, but for Random forest (RF) and Multilayer perceptron (MLP) we used the default setting. The reason was that the SVM, LR and KNN did not show good performance when using their default setting compared to MLP and RF. Also, it is beyond the scope of this thesis to explore the parameters of the classifier. We were more interested in exploring the influence of sampling methods on these classifiers.

The percentage of samples to be created by a sampling method was set to 100%, which means that the number of minority samples in the training set will be equal to the number of majority samples in the training set after sampling. We implemented our work using Python and R packages [119], [108] and [120].

4.3.4 Evaluation Method

In this chapter we ran 30 trials for each model of all the experiments [121]. The mean and standard deviation of weighted F1 score, recall, and precision were used to measure the performance of the classifiers. In addition, a cross-validation approach with three folds was employed. This thesis did not adopt more than three folds because most of the activities had a very low number of samples in some datasets.

In line with previous works in literature [122] and [123] in term of using statistical analysis of sampling methods, we use the mean f1 scores for our analysis. A statistical analysis was performed to discover whether there were significant differences among the sampling methods in terms of their performance metric (the mean F1 scores) across the five classifiers.

The normality of the data was then examined [124]. Data here refers to the the mean F1 scores that were obtained from 30 trials of our experiments. In section 4.4.6, the data normally was explored by using the Anderson-Darling normality test [125]. This test is commonly used in the literature so we used it in this thesis [124]. The Anderson-Darling normality test uses the p-value (α = 0.05) to examine normality in data [126]. The null hypothesis is that the data is normally distributed [125]. However, if the p-value of this test is less than (α = 0.05), the alternative hypothesis is confirmed, which is that the data is non-normal [126]. More details about the Anderson-Darling normality test are discussed in [125].

In line with prior study [127] the Friedman test [128] was conducted on the mean F1

scores [129]. This test does not presume normality of the data [103]. Consequently, it was chosen in this thesis.

The Friedman test ranks the data of each classifier for each sampling technique and then analyses the values of the ranks [122]. This test then provides a summary of the ranks for each sampling approach [130]. This can be helpful in identifying the most useful sampling method [131]. The section 4.4.6 provides more details about the raking process. Also, more explanation of the Friedman test ranking is in [132].

The null hypothesis of the Friedman test is that all sampling methods perform in the same way [122]. The alternative is that at least one or more of the sampling methods significantly performs better than others [130].

The thesis also used the analysis of variance (ANOVA) test [133]. The reason we used it because it can be implemented when normality is presented in data [134].

Here, the null hypothesis is that the results of all the sampling methods are the same [122]. The alternative hypothesis is that at least one sampling method is significantly different [135]. The ANOVA test uses the p-value (α = 0.05) to examine null hypothesis [123]. If the null hypothesis is rejected, another test is applied to determine which sampling method that is significantly different [136]. For example, in the literature the post hoc test are often applied [123]. For more details about ANOVA test see [133]. The next section will present the results of this chapter.

4.4 Results

4.4.1 Preliminary (Baseline)

As a baseline for evaluation, we applied different recognition methods, including Multilayer Perceptron (MLP), Support-Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), and K-Nearest Neighbour (KNN). We used the F1 score, recall, and precision to report these common classification method's performances. Here, we did not apply any oversampling methods (baseline).

When the baseline classifiers trained using the original imbalanced training data without applying any oversampling method, the F1 scores of baseline classifiers varied drastically. Figure 4.5 presents a general overview of the obtained F1 score of baseline classifiers over all used datasets.

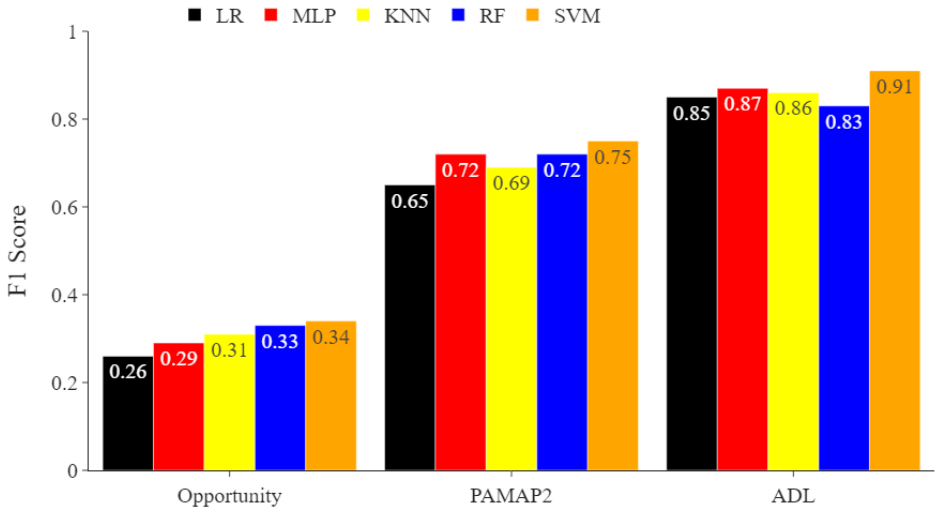

Figure 4.5: The mean F1 score of all classifiers on multiple datasets The reported mean of F1 scores were obtained from 30 repetitions

It can be seen that SVM and then MLP performed comparatively better than the other classifiers. In contrast, KNN, RF and LR showed the lowest performances.

Table 4.2 indicates that all of the evaluation metrics results are below 0.35% in the Opportunity dataset, which shows that none of the classifiers have achieved acceptable results. SVM obtained the most significant F1 score than all other classifiers, and it was 0.34%. Also, the recall and precision of SVM were better than other classifiers. For instance, the recall was 0.39% and precision 0.34%. In contrast, the LR classifier performed the worse, and the F1 score was 0.26%, recall 0.33%, and precision 0.27%.

In the PAMAP2 dataset, the greatest results belong to the SVM classifiers, and the

Table 4.2: Comparing the performance of the baseline classifiers on the Opportunity dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls, and precisions were obtained from 30 repetitions

Classifier	F1 Score	Recall	Precision
SVM	0.34 ($\pm$ 0.01)	0.39	0.34
MLP	0.29 ($\pm$ 0.02)	0.34	0.30
RF	0.33 ($\pm$ 0.03)	0.36	0.34
LR	0.26 ($\pm$ 0.01)	0.33	0.27
KNN	0.31 ($\pm$ 0.05)	0.35	0.32

Table 4.3: Comparing the performance of the baseline classifier on the PAMAP2 dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls, and precisions were obtained from 30 repetitions

Classifier	F1 Score	Recall	Precision
SVM	0.75 ($\pm$ 0.06)	0.76	0.78
MLP	0.72 ($\pm$ 0.08)	0.73	0.75
RF	0.72 ($\pm$ 0.06)	0.72	0.75
LR	0.65 ($\pm$ 0.09)	0.67	0.67
KNN	0.69 ($\pm$ 0.03)	0.69	0.72

lowest to LR. Table 4.3 shows SVM obtained results for the F1 score was 0.75%, the recall was 0.76%, and precision was 0.78%. In contrast, the obtained F1 score of LR was 0.65%, and the recall and precision were 0.67%, respectively.

Interestingly, we observed that the MLP and RF achieved similar performances. For example, for both classifiers, the F1 score was 0.72%, and precision 0.75%. KNN showed modest results compared to the worse classifier (LR). KNN's F1 score and recall were 0.69%, and precision was 0.72%.

In the ADL dataset, the best classifier was the SVM classifier. As shown in Table 4.4 it achieved a similar F1 score and recall, which was 0.91%, and the precision was 0.92%. However, the RF showed the lowest performance compared to other classifiers. The obtained F1 score was 0.83%. The recall was 0.82%, and the precision was 0.85%.

The baseline classifier's ability to recognise human activities indicated that the SVM and MLP showed better performance in most of the cases than LR, RF, and KNN.

Table 4.4: Comparing the performance of the baseline classifiers on the ADL dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls, and precisions were obtained from 30 repetitions

Classifier	F1 Score	Recall	Precision
SVM	0.91 ($\pm$ 0.04)	0.91	0.92
MLP	0.87 ($\pm$ 0.05)	0.87	0.89
RF	0.83 ($\pm$ 0.04)	0.82	0.85
LR	0.85 ($\pm$ 0.03)	0.86	0.85
KNN	0.86 ($\pm$ 0.04)	0.85	0.88

Next, we presented class-specific recognition results for the SVM and MLP in order to identify how these classifiers suffer to recognise the less represented activities. We only used the SVM and MLP because they performed better than LR, RF, and KNN.

4.4.2 Class-Specific Recognition Results for MLP and SVM Classifiers

Class imbalance significantly influences a classification algorithm's performance because the majority classes are likely to negatively affect the performance of a classifier compared to the minority classes. In our thesis's datasets, Opportunity, PAMAP2 and ADL are suffering from class imbalance because the frequency occurrence of some activities is lower comparing with other activities presented in the datasets. Here, we showed the F1 score, recall, and precision of the MLP and SVM per activity to demonstrate how class imbalance negatively influences the MLP and SVM performance.

For example, on the Opportunity dataset, *Drink_from_Cup* and *Clean_Table* activities are more represented, but *Close_Drawer2, Close_Drawer2, Close_Drawer3, Open_Drawer2, Open_Fridge*, and *Open_Drawer1* activities are less frequent. As Table 4.5 shows, the MLP and SVM performance was higher on the F1 score, recall and precision for the most represented activities compared to the less presented activities such as *Open_Fridge*, where the classifiers were not able to recognise successfully.

Similarly, on the PAMAP2 dataset, the *Sitting* and *Walking* activities are among the activity classes that are frequency occurred compared to the less performed activities such as *Rope Jumping*. Therefore, MLP and SVM demonstrated a higher performance, as Table 4.6 illustrates, on the F1 score, recall and precision for *Sitting* and *Walking* activities.

Likewise, on the ADL dataset, the *Working at Computer (WAC)* and *Talking while Standing (TWS)* activities appeared more often, whereas the *Walking and Talking with Someone (WATWS)* and *Walking and Going Up/Downstairs (SWGUDS)* activities were less frequently performed in the dataset. Consequently, Table 4.7 indicates that MLP and SVM revealed better performances on the F1 score, recall and precision for more represented activities, such as WAC. For example, the F1 score of MLP to recognise WAC activity was a high 0.97%, but for less represented activities, including *SWGUDS*, inadequate 0.09%.

Together these class-wise recognition results provide important insights into the negative influence of unequally class distribution on the human activity classifiers. Therefore, it is now well established that class imbalance needs to be addressed in order to enhance the performance of human activity recognition.

Next, we will apply different sampling methods and compare their impact on human activity recognition method's performance.

Table 4.5: Class-wise recognition results for Opportunity dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls, and precisions were obtained from 30 repetitions. Note that high F1 score achieved for the most represented activities compared to the less presented activities

Classifier	MLP			SVM		
Activity	F1 Score	Recall	Precision	F1 Score	Recall	Precision
Clean_Table	0.45 ($\pm$ 0.00)	0.53	0.42	0.57 ($\pm$ 0.00)	0.64	0.54
Close_Dishwasher	0.04 ($\pm$ 0.00)	0.03	0.06	0.12 ($\pm$ 0.00)	0.09	0.22
Close_Door1	0.08 ($\pm$ 0.00)	0.08	0.09	0.18 ($\pm$ 0.00)	0.21	0.17
Close_Door2	0.24 ($\pm$ 0.00)	0.20	0.30	0.40 ($\pm$ 0.00)	0.35	0.46
Close_Drawer1	0.10 ($\pm$ 0.00)	0.08	0.17	0.05 ($\pm$ 0.00)	0.04	0.06
Close_Drawer2	0.29 ($\pm$ 0.00)	0.28	0.57	0.26 ($\pm$ 0.00)	0.28	0.23
Close_Fridge	0.01 ($\pm$ 0.00)	0.01	0.02	0.06 ($\pm$ 0.00)	0.05	0.11
Drink_from_Cup	0.67 ($\pm$ 0.00)	0.93	0.54	0.70 ($\pm$ 0.00)	0.93	0.58
Open_Dishwasher	0.31 ($\pm$ 0.00)	0.30	0.32	0.33 ($\pm$ 0.00)	0.29	0.40
Open_Door1	0.17 ($\pm$ 0.00)	0.16	0.20	0.27 ($\pm$ 0.00)	0.23	0.33
Open_Door2	0.22 ($\pm$ 0.00)	0.18	0.33	0.36 ($\pm$ 0.00)	0.37	0.37
Open_Drawer2	0.05 ($\pm$ 0.00)	0.03	0.13	0.05 ($\pm$ 0.00)	0.03	0.20
Open_Drawer3	0.26 ($\pm$ 0.00)	0.31	0.23	0.27 ($\pm$ 0.00)	0.29	0.27
Open_Fridge	0.00 ($\pm$ 0.00)	0.00	0.00	0.00 ($\pm$ 0.00)	0.00	0.00
Open_Drawer1	0.18 ($\pm$ 0.00)	0.15	0.23	0.23 ($\pm$ 0.00)	0.19	0.32
Toggle_Switch	0.12 ($\pm$ 0.00)	0.12	0.18	0.12 ($\pm$ 0.00)	0.11	0.15
Close_Drawer_3	0.23 ($\pm$ 0.00)	0.19	0.30	0.21 ($\pm$ 0.00)	0.17	0.29

Table 4.6: Class-wise recognition results for PAMAP2 dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls, and precisions were obtained from 30 repetitions. Note that high F1 score achieved for the most represented activities compared to the less presented activities

Classifier	MLP			SVM		
Activity	F1 Score	Recall	Precision	F1 Score	Recall	Precision
Nordic Walking	0.63 ($\pm$ 0.00)	0.58	0.73	0.69 ($\pm$ 0.00)	0.66	0.74
Ascending Stairs	0.46 ($\pm$ 0.00)	0.41	0.54	0.48 ($\pm$ 0.00)	0.44	0.57
Cycling	0.72 ($\pm$ 0.00)	0.75	0.72	0.73 ($\pm$ 0.00)	0.78	0.71
Descending Stairs	0.87 ($\pm$ 0.00)	0.83	0.93	0.88 ($\pm$ 0.00)	0.82	0.95
Ironing	0.80 ($\pm$ 0.00)	0.86	0.74	0.82 ($\pm$ 0.00)	0.86	0.78
Lying	0.71 ($\pm$ 0.00)	0.77	0.7	0.72 ($\pm$ 0.00)	0.73	0.75
Rope Jumping	0.44 ($\pm$ 0.00)	0.52	0.51	0.48 ($\pm$ 0.00)	0.57	0.53
Running	0.70 ($\pm$ 0.00)	0.71	0.88	0.82 ($\pm$ 0.00)	0.80	0.89
Sitting	0.95 ($\pm$ 0.00)	0.98	0.92	0.94 ($\pm$ 0.00)	0.99	0.90
Standing	0.67 ($\pm$ 0.00)	0.64	0.78	0.72 ($\pm$ 0.00)	0.71	0.80
Vacuum Cleaning	0.54 ($\pm$ 0.00)	0.60	0.50	0.59 ($\pm$ 0.00)	0.65	0.55
Walking	0.90 ($\pm$ 0.00)	0.90	0.89	0.91 ($\pm$ 0.00)	0.91	0.90

Table 4.7: Class-wise recognition results for ADL dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls, and precisions were obtained from 30 repetitions. Note that high F1 score achieved for the most represented activities compared to the less presented activities

Classifier	MLP			SVM		
Activity	F1 Score	Recall	Precision	F1 Score	Recall	Precision
GUDS	0.51 ($\pm$ 0.00)	0.41	0.72	0.68 ($\pm$ 0.00)	0.77	0.63
SWGUDS	0.09 ($\pm$ 0.00)	0.12	0.22	0.29 ($\pm$ 0.00)	0.22	0.54
Standing	0.77 ($\pm$ 0.00)	0.83	0.72	0.86 ($\pm$ 0.00)	0.93	0.81
TWS	0.95 ($\pm$ 0.00)	0.94	0.96	0.95 ($\pm$ 0.00)	0.94	0.96
WAC	0.97 ($\pm$ 0.00)	0.98	0.96	0.98 ($\pm$ 0.00)	0.98	0.97
WATWS	0.26 ($\pm$ 0.00)	0.22	0.41	0.24 ($\pm$ 0.00)	0.22	0.41
Walking	0.88 ($\pm$ 0.00)	0.84	0.94	0.94 ($\pm$ 0.00)	0.92	0.96

4.4.3 Comparing the Proposed Sampling Methods to the Existing Sampling Methods

We proposed in this chapter three sampling techniques, DBM, NDBM and CBM, to rebalance the training dataset to improve the human activity classifier's performance. In this section, we aimed to compare the proposed methods to the existing sampling methods, including SMOTE, Random_SMOTE, SMOTE_Tomeklinks, MSMOTE, CBSO and ProWSyn.

It is important to point out that we only used the MLP classifier because its performance was always significantly enhanced with every proposed oversample method. The results of applying the sampling methods to the Support vector machine (SVM), Random forest (RF) Logistic regression (LR), and K-nearest neighbours (KNN) are presented in the appendix B.

To find out why these classifiers did not produce a significant performance with the sampling methods was beyond this work's scope and will be considered for future work. To provide a comprehensive evaluation of the used methods, the performance results are shown in terms of the F1 score, recall, and precision. The following section will compare the obtained results from applying distance-based method (DBM), SMOTE and Random_SMOTE, and comparing their influence on the performance of the MLP.

4.4.3.1 Distance-Based Method (DBM)

The DBM was the first proposed sampling technique which was on combining two distance-based oversampling approaches: SMOTE and Random_SMOTE.

The problem of a small sample size occurred when there were a small number of samples (data points) for certain classes resulting in a lack of information and then a learning algorithm trained with these few samples that might not be enough for making generalisations in unseen samples. [27]. Therefore, we proposed the DBM to handle the small sample size problem in the training data to enlarge its size.

Furthermore, we compared the proposed DBM with the existing techniques such as baseline, SMOTE and Random_SMOTE. We also evaluated the performance of DBM, SMOTE, and Random_SMOTE by using the MLP classification method to recognise human activity.

The MLP classifier was applied to classified human activity. Figure 4.6 shows the classifier F1 score performance of the baseline with different sampling methods. It can be seen in Figure 4.6 that the DBM was better than SMOTE and Random_SMOTE across all used datasets.

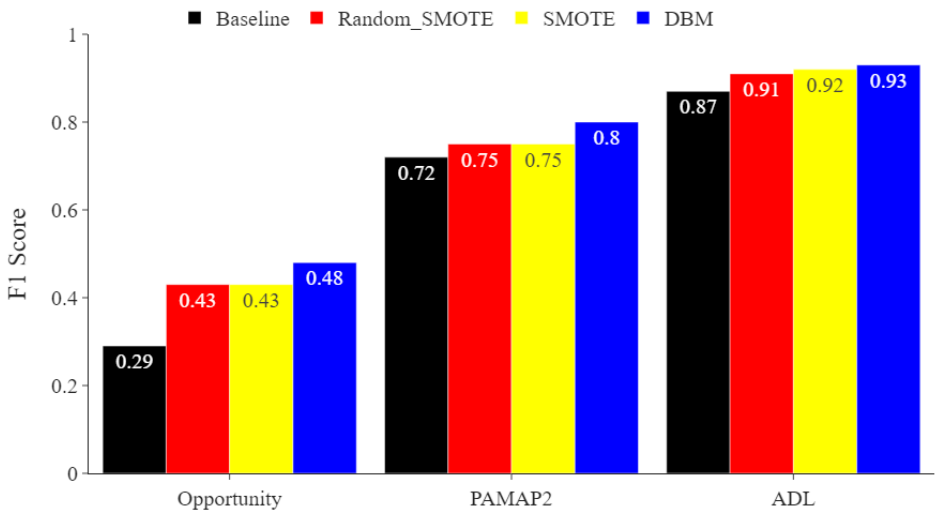

Figure 4.6: Comparing the mean F1 score of baselines (MLP), and the proposed DBM, SMOTE and Random_SMOTE on multiple datasets. The reported mean of F1 scores were obtained from 30 repetitions

Moreover, Table 4.8 shows that by using the DBM, the F1 score, recall, and precision significantly increased by more than 13% on the Opportunity dataset compared to the SMOTE and Random_SMOTE approaches. It is important to point out that even though the DBM was superior on the Opportunity dataset, the SMOTE and Random_SMOTE methods effectively improved the classifier performance compared to the baseline.

It is evident that the DBM produced better performance on the F1 score, recall, and precision than SMOTE and Random_SMOTE techniques, on the PAMAP2 dataset. For instance, on the PAMAP2 dataset, the F1score was improved by 8%, and both the recall and precision by 7%.

On the ADL dataset, the DBM showed acceptable improvements on the F1 score and the recall when compared to the SMOTE and Random_SMOTE techniques. However, both the DBM and SMOTE method showed similar performances on the precision. For instance, the F1 score of the MLP classifier improved by 6% and the recall by 5% with the DBM , whereas

Table 4.8: Comparing the performance of MLP, and proposed DBM, SMOTE and Random_SMOTE on multiple datasets. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 repetitions. The largest obtained scores are in bold font

Data	Method	F1 Score	Recall	Precision
ADL	Baseline	0.87 ($\pm$ 0.05)	0.87	0.89
	SMOTE	0.92 ($\pm$ 0.07)	0.91	0.94
	Random_SMOTE	0.91 ($\pm$ 0.09)	0.90	0.93
	DBM	**0.93** ($\pm$ 0.08)	**0.92**	**0.94**
Opportunity	Baseline	0.29 ($\pm$ 0.02)	0.34	0.30
	SMOTE	0.43 ($\pm$ 0.04)	0.42	0.46
	Random_SMOTE	0.43 ($\pm$ 0.04)	0.42	0.46
	DBM	**0.48** ($\pm$ 0.05)	**0.48**	**0.51**
PAMAP2	Baseline	0.72 ($\pm$ 0.08)	0.73	0.75
	SMOTE	0.75 ($\pm$ 0.06)	0.75	0.78
	Random_SMOTE	0.75 ($\pm$ 0.05)	0.75	0.78
	DBM	**0.80** ($\pm$ 0.05)	**0.80**	**0.82**

the precision increased by 5% with both the DBM and SMOTE.

The obtained results showed that the proposed DBM had the potential, when compared to SMOTE and Random_SMOTE, to improve the performance of MLP's ability to learn from an imbalanced human activity dataset.

The next section compares the second proposed approach, which we called noise detection-based method (NDBM) to SMOTE_TomekLinks and MSMOTE sampling methods for improving learning from imbalanced human activity data.

4.4.3.2 Noise Detection-Based Method (NDBM)

The NDBM was the second proposed technique which was on combining two oversampling and undersampling based approaches, including SMOTE_TomekLinks and MSMOTE.

Class overlap occurs when samples of more than one class share similarities in feature space. However, when they belong to different classes, the class overlap issue hinders a classification algorithm's performance from learning to differentiate between these classes.

It is possible that the issue of class overlap in human activity was introduced when to applying the sliding window method while pre-processing human activity data. When a label is assigned to a segment, the majority vote is used to select a label (as explained in section 2.4.4). In addition, the class overlap issue might be caused by inadequate labelling while annotating the sensor data. For example, when one assigns a class label to a wrong senor data sample.

The NDBM was proposed to handle the class overlapping challenge in the training data. We rebalanced the training data using NDBM, SMOTE_TomekLinks, and MSMOTE approaches. Then, we compared the NDBM with the existing methods, namely, baseline,

SMOTE_TomekLinks, and MSMOTE.

Figure 4.7 demonstrates the MLP classifier F1 score performance of baseline and different oversampling methods. It indicates the NDBM was more efficient than SMOTE_TomekLinks, and MSMOTE across the employed datasets.

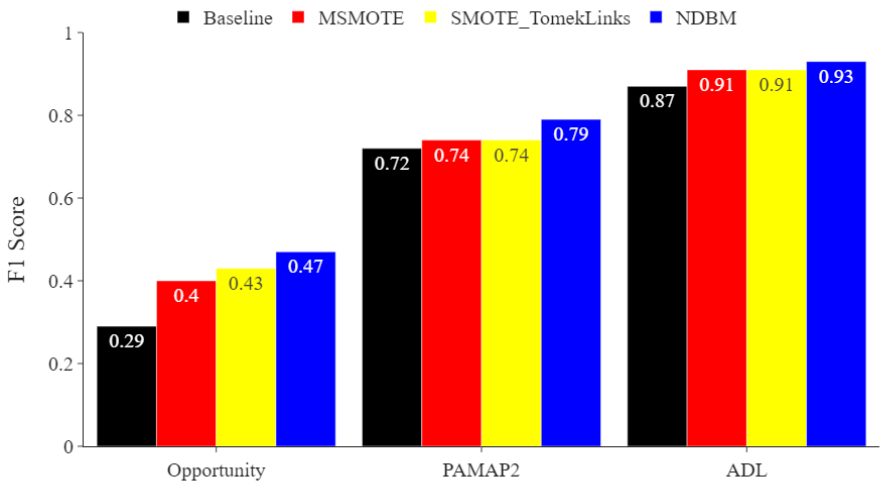

Figure 4.7: Comparing the mean F1 score of baselines (MLP), and the proposed NDBM, MSMOTE and SMOTE_TomekLinks on multiple datasets. The reported mean of F1 scores were obtained from 30 repetitions

In addition, Table 4.9 indicates that the recall and precision were improved with the NDBM. For example, on the Opportunity dataset, compared to SMOTE_TomekLinks, and MSMOTE, NDBM led the MLP's F1 score significantly enhanced by 18% and the recall by 13%, where the precision increased by 19%.

Likewise, on the PAMAP2 dataset, the F1 score improved by 7%, and the recall and precision by 6%.

Table 4.9 demonstrates that NDBM was better than SMOTE_TomekLinks, and MSMOTE techniques at improving the F1 score, recall, and precision on the ADL dataset. For instance, the F1 score and precision increased by 6% and the recall by 5%.

These results indicated that the NDBM was more useful in improving the MLP's ability (on all used datasets) to learn from imbalanced human activity data, rather than the SMOTE_TomekLinks and MSMOTE techniques.

The following section compares the last proposed method in this chapter, which was named cluster-based method (CBM), to the CBSO and ProWsyn sampling algorithms to explore how they enhance the MLP's ability to learning from imbalanced human activity data.

Table 4.9: Comparing the performance of MLP, proposed NDBM, MSMOTE and SMOTE_TomekLinks on multiple datasets. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 repetitions. The largest obtained scores are in bold font

Data	Method	F1 Score	Recall	Precision
ADL	Baseline	0.87 ($\pm$ 0.05)	0.87	0.89
	MSMOTE	0.91 ($\pm$ 0.07)	0.90	0.93
	SMOTE_TomekLinks	0.91 ($\pm$ 0.07)	0.91	0.94
	NDBM	**0.93** ($\pm$ 0.07)	**0.92**	**0.95**
Opportunity	Baseline	0.29 ($\pm$ 0.02)	0.34	0.30
	MSMOTE	0.40 ($\pm$ 0.07)	0.40	0.42
	SMOTE_TomekLinks	0.43 ($\pm$ 0.04)	0.43	0.45
	NDBM	**0.47** ($\pm$ 0.07)	**0.47**	**0.49**
PAMAP2	Baseline	0.72 ($\pm$ 0.08)	0.73	0.75
	MSMOTE	0.74 ($\pm$ 0.06)	0.74	0.77
	SMOTE_TomekLinks	0.74 ($\pm$ 0.05)	0.75	0.77
	NDBM	**0.79** ($\pm$ 0.05)	**0.79**	**0.81**

4.4.3.3 Cluster-Based Method (CBM)

The CBM was developed by combining two cluster-based oversampling approaches, using CBSO and ProWsyn. The CBSO and ProWsyn cluster the minority class samples before generating synthetic samples. Therefore, CBM was constructed to consider the within-class imbalance problem in the training data, while generating synthetic samples as the CBSO and ProWsyn algorithms cluster the minority class samples before generating synthetic samples.

The within-class imbalance occurs when a class is comprised of a number of various subclusters where these subclusters do not include the same number of samples [99]. The within-class imbalance may arise in human activity because of intraclass variability. This corresponds to a case where the an activity is performed in a different way either by individuals or by an individual.

Figure 4.8 presents the change of the MLP's F1 score for baseline, the proposed CBM, CBSO and ProWsyn on the three different datasets. Figure 4.8 shows a clear improvement on the F1 score by using the proposed CBM rather than the CBSO and ProWsyn methods.

As shown in Table 4.10, on the Opportunity dataset, when we applied our CBM to oversample the training data, the MLP demonstrated the maximum obtained F1 score, which had improved from 0.29% to 0.49%. In fact, the F1 score increased by 20%. In contrast, with the CBSO and ProWsyn methods, the F1 score only increased by 14% at best.

Table 4.10 indicates that the highest achieved recall and precision scores were achieved by applying the CBM belong to the Opportunity dataset. The baseline recall was 0.34% which then increased to 0.49%. This is a significant improved to the recall of 15%, meanwhile the precision score was enhanced by 21%. Interestingly, the CBSO and ProWsyn methods

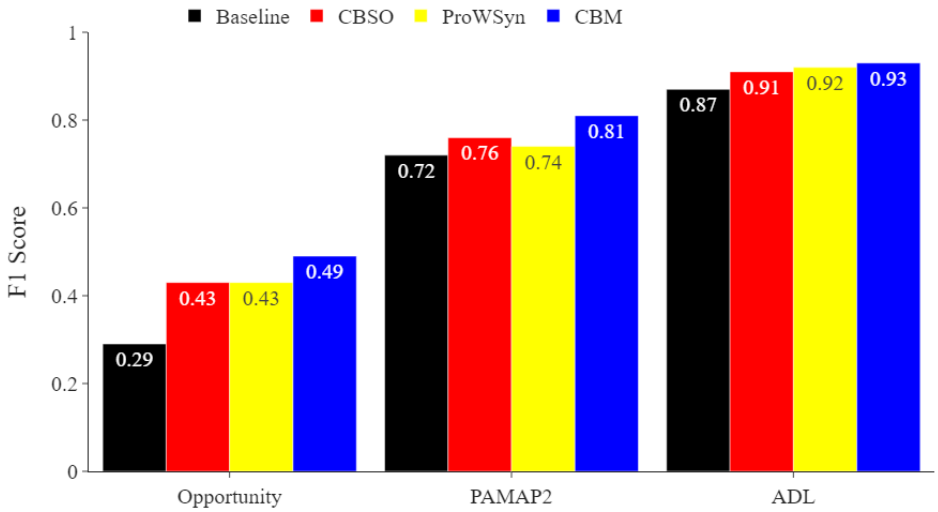

Figure 4.8: Comparing the mean F1 score of baselines (MLP), and the proposed CBM, CBSO and ProWsyn on multiple datasets. The reported mean of F1 scores were obtained from 30 repetitions

showed notable improvement on the F1 score, recall and precision (by more than 8% on the Opportunity dataset).

Table 4.10 highlights that, on the PAMAP2 dataset, the proposed CBM exhibited a significant improvement on F1 score, recall, and precision compared to the CBSO and ProWsyn methods. For example, the F1 improved by 9%, and both recall and precision by 8%. In contrast, the CBSO and ProWsyn methods were unable to effectively enhance the performance of the MLP.

Table 4.10 also illustrates that the CBM's performance showed better results than the CBSO and ProWsyn on the ADL dataset. For instance, with the CBM, the F1 score and precision improved by 6%, and the recall increased by 5%. By comparing the CBSO and ProWsyn oversampling approaches, it is evident that the ProWsyn method is more useful in enhancing the MLP classifier's performance. For instance, both the F1 score and precision were enhanced by 5%, and the recall by 4%.

Compared to the CBSO and ProWsyn techniques, we can see that the CBM is superior because, it significantly improves the MLP's ability to learn from multiple imbalanced human activity datasets.

The next section presents a comparison of the proposed the DBM, NDBM and CBM , and their ability to increase the capacity of the MLP in learning from imbalanced human activity data.

Table 4.10: Comparing the performance of MLP, and proposed CBM, CBSO and ProWsyn on multiple datasets. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 repetitions. The largest obtained scores are in bold font

Data	Method	F1 Score	Recall	Precision
ADL	Baseline	0.87 ($\pm$ 0.05)	0.87	0.89
	CBSO	0.91 ($\pm$ 0.09)	0.90	0.94
	ProWSyn	0.92 ($\pm$ 0.09)	0.91	0.94
	CBM	**0.93** ($\pm$ 0.09)	**0.92**	**0.95**
Opportunity	Baseline	0.29 ($\pm$ 0.02)	0.34	0.30
	CBSO	0.43 ($\pm$ 0.02)	0.43	0.45
	ProWSyn	0.43 ($\pm$ 0.06)	0.43	0.45
	CBM	**0.49** ($\pm$ 0.04)	**0.49**	**0.51**
PAMAP2	Baseline	0.72 ($\pm$ 0.08)	0.73	0.75
	CBSO	0.76 ($\pm$ 0.04)	0.75	0.78
	ProWSyn	0.74 ($\pm$ 0.05)	0.74	0.78
	CBM	**0.81** ($\pm$ 0.05)	**0.81**	**0.83**

4.4.4 Comparing the Performance of the Proposed sampling Methods

In this section, we compared the performance of the three proposed sampling methods (DBM, NDBM and CBM), when attempting to improve the MLP's ability to generalise. This section also provided a comparison for the three proposed sampling methods in terms of running times of training sets.

The performance results were demonstrated in terms of the F1 score. The standard deviation of the F1 scores for the test folds (we performed 3-fold cross-validation as described in section 4.3.4), were recall, and precision. This was because we wanted to ensure that we provided a comprehensive evaluation of the proposed methods.

The results revealed that the three proposed sampling methods made a positive impact on human activity recognition. Therefore, integrating more than one sampling method into a recognition model certainly positively influences the classifier performance on all evaluation metrics.

Figure 4.9 shows that using the proposed sampling methods led to a potential improvement on the F1 score across all used datasets.

Also, Table 4.11 shows small standard deviation of recognition scores. They were consistent across each of the three test folds. Overall, it reveals our sampling method's effectiveness.

As Table 4.11 shows, on the Opportunity dataset, due to applying the three sampling methods, there is a significant improvement not only on the F1 score, but also on the recall and precision. The most significant improvement in the evaluation metrics was achieved with the CBM. For example, the F1 score was enhanced by 20%, the recall by 15% and the precision

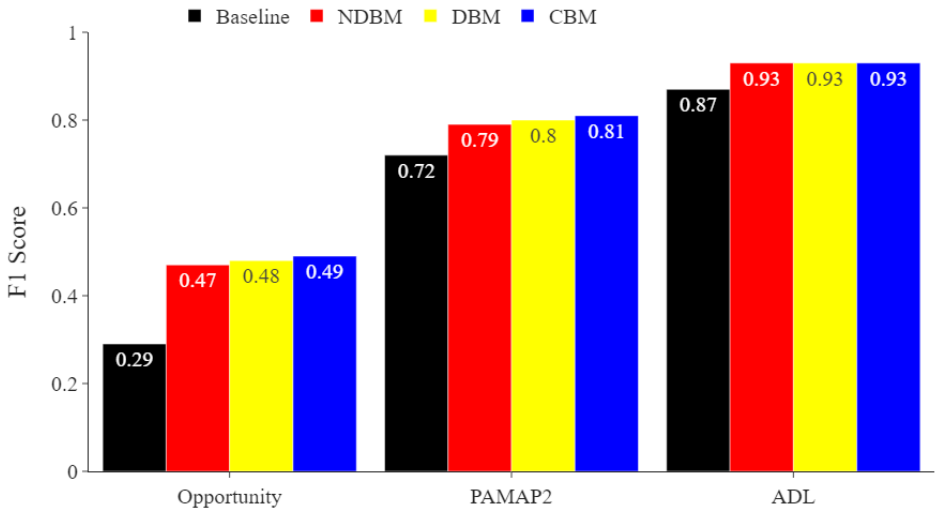

Figure 4.9: Comparing the mean F1 score of baselines (MLP), the proposed DBM, NDBM and CBM on multiple datasets. The reported mean of F1 scores were obtained from 30 repetitions

by 21%. In contrast, the lowest enhancement in the evaluation metrics was attained with the NDBM. For instance, the F1 score increased by 18%, the recall by 13% and the precision by 19%

Similarly, on the PAMAP2 dataset, Table 4.11 indicates the most considerable increase in the evaluation metrics was accomplished with the CBM , whereas the lowest was with NDBM. We saw an increase in performance of MLP on the F1 by 9%, and both the recall and precision by 8%. By comparison, the NDBM improved the performance of MLP on the F1 by 7%, and the recall and precision by 6%.

On the ADL dataset, the three proposed sampling methods showed a similarly improved performance of the MLP on the F1 score and the recall. However, precision with NDBM and CBM was marginally better. For example, with our proposed three sampling methods, the F1 score increased by 6% and the recall by 5%. The precision improved by 6% with NDBM and CBM but was enhanced by 5% with DBM.

Figure 4.10 offers a comparison for each proposed sampling method in terms of running times. The analysis was performed on a Fierce PC with 16 GB RAM, Intel Core i7-7700 processor with 3.60 GHz and using Ubuntu 16.04 LTS (64- bits). The DBM demonstrated the best performance in terms of running times for training compared with the other proposed sampling methods.

These results highlight that the proposed sampling methods are significantly capable of enhancing activity recognition performance. Therefore, the performance of the MLP improved as it was able to learn from an imbalanced activities dataset.

The next section compares how the three proposed sampling methods influence the

Table 4.11: Comparing the performance of MLP, the proposed DBM, NDBM and CBM on multiple datasets. The reported mean of F1 scores and (± standard deviation), recalls and precisions were obtained from 30 repetitions. The largest obtained scores are in bold font

Data	Method	F1 Score	Recall	Precision
ADL	Baseline	0.87 (± 0.05)	0.87	0.89
	DBM	0.93 (± 0.08)	0.92	0.94
	NDBM	0.93 (± 0.07)	0.92	0.95
	CBM	0.93 (± 0.09)	0.92	0.95
Opportunity	Baseline	0.29 (± 0.02)	0.34	0.30
	DBM	0.48 (± 0.05)	0.48	0.51
	NDBM	0.47 (± 0.07)	0.47	0.49
	CBM	**0.49** (± 0.04)	**0.49**	**0.51**
PAMAP2	Baseline	0.72 (± 0.08)	0.73	0.75
	DBM	0.80 (± 0.05)	0.80	0.82
	NDBM	0.79 (± 0.05)	0.79	0.81
	CBM	**0.81** (± 0.05)	**0.81**	**0.83**

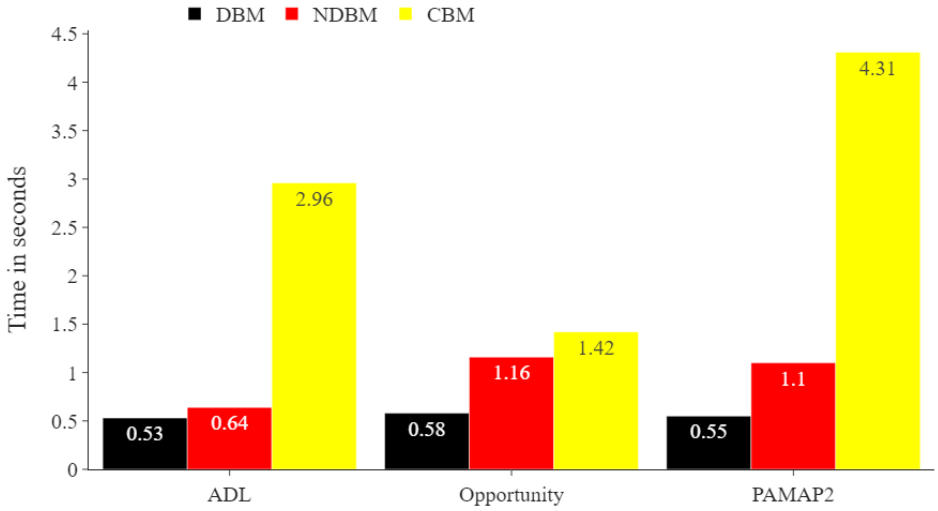

Figure 4.10: Comparing running times in seconds of the proposed DBM, NDBM and CBM for all training datasets. The number of samples in the training sets for the ADL, Opportunity, and PAMAP2 datasets were 11776, 1569 and 6450, respectively.

performance of the MLP classifier on activity-wise.

4.4.5 Comparing the Proposed Sampling Methods Influence on the Activities-Wise

This chapter proposed and applied three different sampling approaches as a way to overcome unbalanced samples in the training data. Consequently, the generalisation ability of the MLP

was enhanced to recognise human activities. We compared the recognition results using the F1 score of each activity obtained by the MLP classifier to analyse the sampling methods' impact. Specifically, we showed the mean F1 score of the most underrepresented activities based on 30 trials because we sought to highlight how useful the proposed sampling methods are. Similar to the the standard deviation of the mean F1 scores for MLP's baseline in tables 4.5, 4.6 and 4.7 on all datasets, DBM, NDBM and CBM's standard deviation values were zero, too. Also, note that the figures in (section 4.3.1) show the class distribution for each dataset.

For the Opportunity dataset, multiple activities were underrepresented, such as *Open_Fridge*, *Open_Drawer3*, and *Close_Drawer3*. Figure 4.11 indicates that the proposed DBM, NDBM and CBM improved the F1 score of the MLP in recognising underrepresented activities on the Opportunity dataset. Figure 4.11 also shows that without applying the sampling methods (baseline), the MLP classifier could not identify the *Open_Fridge* activity.

By applying the proposed sampling methods, the MLP's ability to recognise underrepresented activities improved. For example, the F1 of the MLP's ability to classify the *Open_Fridge* activity improved by more than 10% using any one of the three proposed sampling methods.

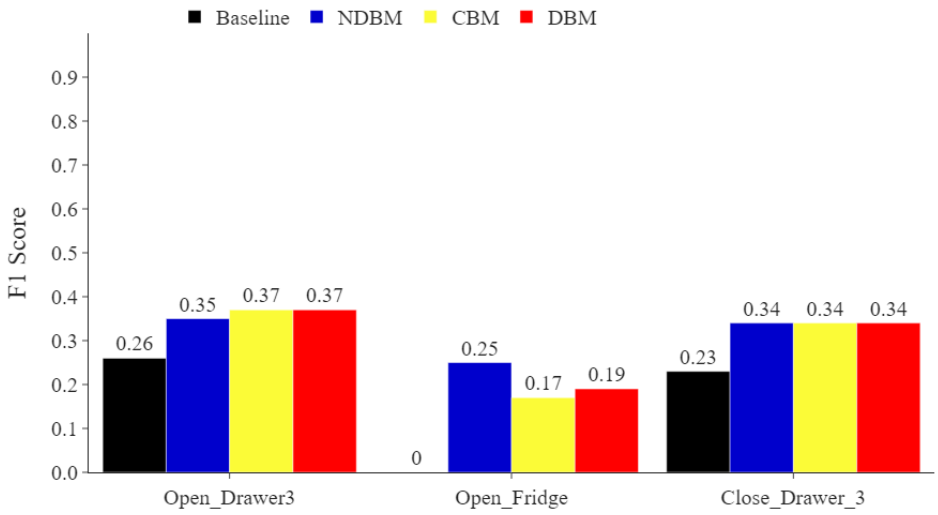

Figure 4.11: Comparing the impact of using the proposed DBM, NDBM and CBM on activity recognition performance, using MLP for the most underrepresented activities *Open_Fridge*, *Open_Drawer3*, and *Close_Drawer3* on the Opportunity dataset.The reported mean of F1 scores were obtained from 30 repetitions

On the ADL dataset, Figure 4.12 also suggests that by comparing applying the three proposed sampling methods the MLP classifier F1 score was improved by more than 10% and gained a significant advantage in identifying the underrepresented activities, including *Going Up/Downstairs (GUDS), Standing Up, Walking and Going Up/Downstairs (SWGUDS)*, and

Walking and Talking with Someone (WATWS).

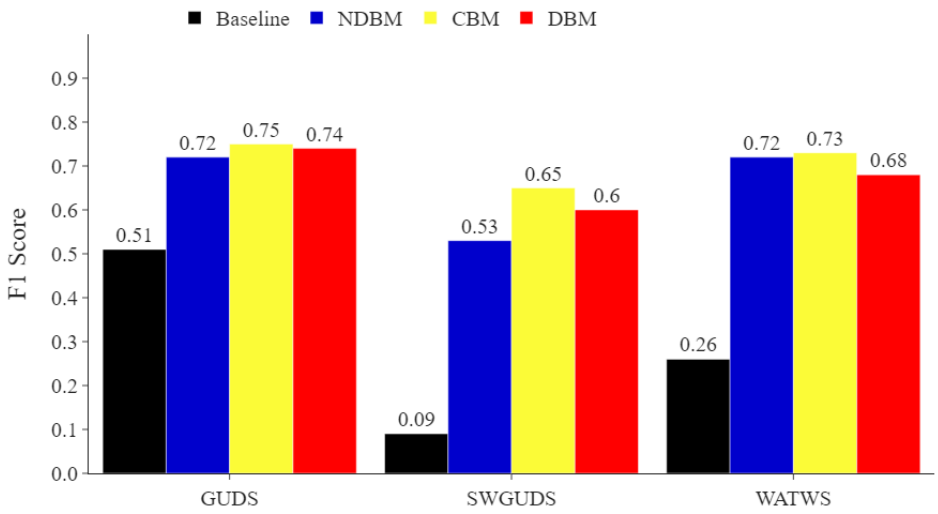

Figure 4.12: Comparing the impact of using the proposed DBM, NDBM and CBM on activity recognition performance, using MLP for the most underrepresented activities (*Going Up/Downstairs (GUDS), Standing Up, Walking and Going Up/Downstairs (SWGUDS)*, and *Walking and Talking with Someone (WATWS)*) on the ADL dataset. The reported mean of F1 scores were obtained from 30 repetitions

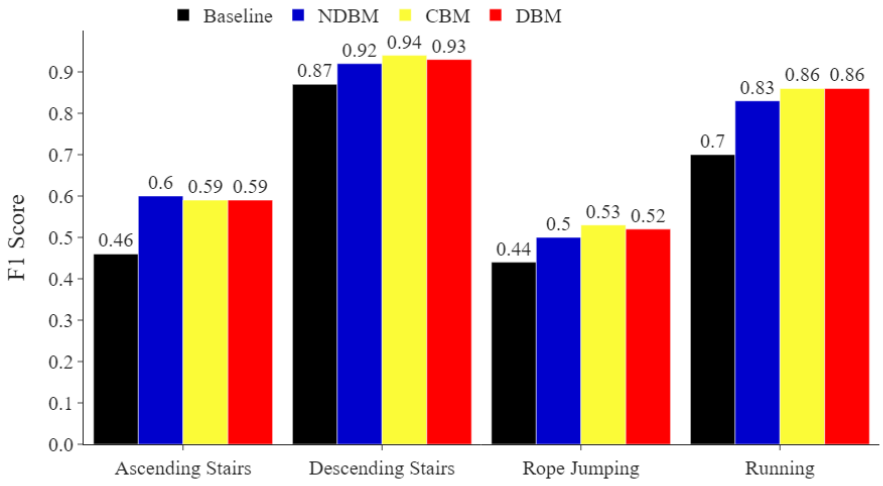

Figure 4.13: Comparing the impact of using the proposed DBM, NDBM and CBM on activity recognition performance, using MLP for the most underrepresented activities (*ascending stairs, descending stairs, rope jumping* and *running*, on the PAMAP2 dataset. The reported mean of F1 scores were obtained from 30 repetitions

Similarly, on the PAMAP2 dataset, Figure 4.13 implies that the MLP classifier was more capable of identifying the underrepresented activities, including *rope jumping, running,*

descending stairs and *ascending stairs*, when either one of the three proposed sampling methods was used. For example, the performance of the MLP improved on the F1 score by at least 6% when identifying the underrepresented *rope jumping*.

Consequently, we can conclude that the proposed DBM, NDBM and CBM has a strong potentiality to improve the MLP ability to generalise when trained with imbalanced human activity data. The result of our experiment is encouraging. Next, we will explore the statistical significance of the DBM, NDBM and CBM.

4.4.6 Statistical Analysis

A statistical analysis was performed to find out whether there were significant differences among the sampling methods based on their performance metric (the mean F1 scores) across the five classifiers.

Statistical significance tests, such as the Friedman test which is a non-parametric method help to identify the best sampling method [124]. The normality assumption was determined by using the Anderson-Darling normality test. This determines whether parametric statistical analysis, such as the ANOVA test, may be applied [126].

Table 4.12 shows the results of the Anderson-Darling normality test of data on all the datasets. Data here means the the mean F1 scores that were obtained from 30 trials of our experiments. The table suggests that the p-value is less than 0.05 (α = 0.05) for the ADL and Opportunity datasets. Consequently, the null hypothesis is rejected, which assumes that the data of these two datasets are normally distributed [126]. This indicated that the data of these datasets is not normally distributed. The ANOVA test then cannot be applied (details in section 4.3.4) [133].

However, on the PAMAP2 dataset, Table 4.12 indcates that the p-value is more than 0.05 (α = 0.05). The ANOVA test can then be applied, while the Friedman test can be used as an alternative to the ANOVA test on the Opportunity and ADL datasets to compare the sampling methods [130].

The Friedman test in Table 4.13 indicates that the p-value of the data is less than 0.05 (α = 0.05) for the ADL and Opportunity datasets. So, the null hypothesis is then violated. This means that there is a statistically significant difference across the sampling methods. In other words, one or more of the sampling methods can show different influences on these datasets.

Tables 4.14 and 4.15 displays the results of the summary of ranks drawn from the Friedman test in the ADL and Opportunity datasets. The process of ranking the data in the Freidman Test is explained in [130]. The data is sorted into blocks (columns), and six classifiers were used. The classifiers were KNN, SVM, LR, RF and MLP, while nine different sampling methods were utilised. The sampling methods were CBSO, NDBM, CBM, DBM, MSMOTE, ProWSyn, Random_SMOTE, SMOTE_Tomeklinks and SMOTE. Each col-

Table 4.12: Comparing the Anderson-Darling normality test. The p-value is less than 0.05 (α = 0.05) for ADL and Opportunity datasets. It is also confirmed that the data of ADL and Opportunity is not normally distributed compared to the PAMPA2 data

Data	Mean	Standard deviation	Sample size	P-value
ADL	0.8840	0.0399	45	0.0007
Opportunity	0.3773	0.0548	45	0.0000
PAMAP2	0.7272	0.0406	45	0.0680

Table 4.13: The Friedman test results indicate that the p-value is less than 0.05 (α = 0.05) for the ADL and Opportunity datasets. This means that one or more of the sampling methods is more effective than the others

Data	Degrees of freedom	Chi-square	P-value
ADL	8	21.8133	0.0053
Opportunity	8	24.2133	0.0021

umn was ranked separately, and the smallest score was assigned a number as a rank. Ranking was conducted across rows. For example, each classification method was ranked as 1 and 2 up to 9 for each sampling method. Table 4.14 and 4.15 summarize the total ranks obtained for each column. Further details on the process followed to rank the data in the Friedman test is shown in [130].

The outcome of the Friedman test in Tables 4.14 and 4.15 of the sum of ranks indicate the CBM method was better when it was compared with the other sampling approaches. This confirmed that the proposed CBM technique may be more effective than the other techniques.

Table 4.16 shows that the p-value was not less than 0.05 on the PAMPA2 dataset. Hence, the ANOVA test had detected no statistical evidence to reject the null hypothesis. In other words, all the sampling methods performed the same and none of them was detected to perform significantly different than the others on the PAMAP2 dataset.

Table 4.14: Information obtained from Friedman test about the sum of the ranks. The CBM is more effective than other methods as it has the highest rank on ADL dataset

Classifier	CBSO	NDBM	CBM	DBM	MSMOTE	Pro-WSyn	Random_SMOTE	SMOTE_TomekLinks	SMOTE
KNN	1	7	9	4	5	8	2	6	3
LR	1	8	3	9	2	5	6	7	4
MLP	3	8	9	7	1	5	2	4	6
RF	1	6	9	4	7	8	3	5	2
SVM	1	8	7	9	2	3	6	4	5
Sum of ranks	7	37	**37**	33	17	29	19	26	20

Table 4.15: Information obtained from Friedman test about the sum of the ranks. The CBM is more effective than other methods as it has the highest rank on Opportunity dataset

Classifier	CBSO	NDBM	CBM	DBM	MSMOTE	Pro-WSyn	Random_SMOTE	SMOTE_TomekLinks	SMOTE
KNN	5	6	9	7	1	4	8	3	2
LR	5	9	7	8	1	2	6	4	3
MLP	5	7	9	8	1	3	2	4	6
RF	4	5	8	3	1	9	7	6	2
SVM	2	7	8	9	1	4	3	5	6
Sum of ranks	21	34	**41**	35	5	22	26	22	19

Table 4.16: ANOVA results for PAMAP2 dataset

Data	Degrees of freedom	Sum of squares	Mean square	F value	P-value
PAMAP2	8	0.0067	0.0008	0.4602	0.8757

4.5 Discussion

We found that the proposed DBM, NDBM and CBM worked better than applying a single sampling algorithm including SMOTE, Random_SMOTE, SMOTE_Tomeklinks, MSMOTE, CBSO and ProWSyn to improve human activity recognition performance. The benefit of the proposed approach compared with the existing sampling algorithms is that it can generate more diverse samples by combining different sampling algorithms. The proposed sampling methods minimised the class imbalance in the training data by enlarging the training data with synthetic samples which improved the human activity recognition performance. In fact, in most cases, our proposed methods substantially improved the different adopted performance metrics in our research, including F1 score, recall, and precision compared to the six existing sampling algorithms. Also, recognition results were consistent across all experiments, which was demonstrated by the comparatively small standard deviation (e.g. in table 4.11).

The proposed CBM possesses a key characteristic that combines multiple clusters-based oversampling algorithms (CBSO and ProWSyn). They use different approaches before generating synthetic samples such as clustering and assign weights of minority classes (activities). Intraclass variability is one of the main characteristics of human activity data, where the same activity is performed differently by the same subject. Therefore, combining two oversampling algorithms with different distinctive clustering procedures made the CBM a promising approach to consider intraclass variability while oversampling the training data. The CBM became more robust and useful in producing diverse samples that increase the classifier's generalisation capability to recognise human activities. An obvious example can be seen in Table 4.10. The CBM led to significantly improve the MLP classifier's performance in all evaluation metrics compared to the CBSO and ProWSyn algorithms across all datasets. In particular, the F1 score always increased by 5% or more with the CBM across all datasets. Furthermore, Table 4.14 and 4.15 show the sum of ranks from the Friedman test on the ADL and Opportunity and indicates that the CBM is more effective comparing with the other sampling methods.

Likewise, the proposed DBM developed by combining two distance-based sampling approaches, (SMOTE Random_SMOTE algorithms), made the DBM more useful in improving the MLP classifier performance compared to SMOTE or Random_SMOTE algorithms across most of the used datasets. The proposed method takes advantage of combining two oversampling methods capable of handling small sample size. It is evident in Table 4.8 that the MLP classifier's performance significantly enhanced with the distance method on the Opportunity and PAMAP2 datasets. For instance, the F1 score was improved by more than 7%. We can speculate that this might be because the DBM overcame the small sample size issue as these datasets possess this problem (see Figure 4.2 and Figure 4.3).

In addition, we observed that the sensor position used to collect data impacts the sam-

pling methods' performance. On the ADL dataset, we used the only data available from the sensor placed on the participant's chest. We found that enlarging the training data samples by using either the three proposed sampling methods or the six sampling existing algorithms, in some cases presented similarly improved performance. For example, Table 4.8 indicates that the DBM and SMOTE algorithm exhibited equal performance on the MLP classifier's precision of 0.94%. However, both the F1 score and recall with the DBM were better than the SMOTE algorithm by 1%.

Even though the CBM was more effective than other sampling algorithms that used cluster in their process to generate synthetic samples such as CBSO and ProWSyn on the ADL dataset, the ProWsyn algorithm performance was not negligible. From Table 4.10, the CBM increased the performance in all metrics by 1% compared to the ProWsyn algorithm. For instance, the CBM enhanced the F1 score of the MLP from 0.87% to 0.93%. In contrast, the ProWsyn approach improved the F1 score to 0.92%. Therefore,it appears that a sensor position on the part of the body that relates to the performed activity is an important factor to consider when applying sampling methods. For instance, in the ADL dataset, the activities including *Walking and Talking with Someone (WATWS)* and *Standing Up, Walking and Going Up/Downstairs (SWGUDS)* might not be reliably recognised from a sensor placed at the chest position. Consequently, the sampling methods such as the proposed DBM, NDBM and CBM are likely to be more useful to other used datasets where the sensors were placed on hand or wrist. A significant consideration in future studies would be to consider which degree the sensor position on the body is attributed to the sampling methods' efficiency.

When comparing the proposed NDBM to SMOTE_TomekLinks and MSMOTE algorithms in Table 4.9, it is clear that SMOTE_TomekLinks and MSMOTE algorithms were useful when combined into one method, namely, the NDBM. It seems possible that the inadequate performance of the SMOTE_TomekLinks algorithm was due to the step where data samples is removed, which might lead to the loss of some useful information. In addition, when MSMOTE divided the samples of the minority class they were assigned into three types safe, border and noise samples. This might reduce the number of training data as some samples will be sampled as noise. Therefore, some potentially useful information for recognising some activities was lost. In contract, when SMOTE_TomekLinks and MSMOTE was combined into NDBM, they became effective because more synthetic samples combined from SMOTE_TomekLinks and MSMOTE techniques improved the MLP's performance. For instance, there was no significant indication that the SMOTE_TomekLinks and MSMOTE algorithm could enhance the MLP classifier's performances ability to recognise human activities on the PAMAP2 dataset, as the F1 score improved by 2% or below. However, when applying the NDBM, which combined the SMOTE_TomekLinks and MSMOTE algorithm, the F1 score, call, and precision of the MLP always enhanced by more than 4% or more on the PAMAP2

dataset.

In term of the number of classes (activities) on a training set in influencing the run time of the proposed methods, Figure 4.10 indicates that data with more classes led to longer run time. For example, the number of training samples on the Opportunity dataset was relatively small (1569), but the data had 17 classes, which increased the running time of the proposed sampling methods.

Ultimately, we found that the introduced DBM, NDBM and CBM were superior in using a single sampling algorithm such as SMOTE, Random_SMOTE, SMOTE_Tomeklinks, MSMOTE, CBSO and ProWSyn to enhance human activity recognition performance when confronted with imbalanced human activity data.

4.6 Summary

The majority of existing study for human activity recognition has used supervised learning methods. These supervised learning methods can show better performance when a large amount of labelled is available. Nevertheless, it is not always possible to obtain substantial amounts of labelled sensor data for several reasons. Collecting and labelling sensor data is not only cumbersome but also can be costly.Also, several noise causes can affect the quality of labelled sensor data, for example, sensor noise. Consequently, most of human activity labelled datasets suffer from class imbalance. In addition, other issues related to class imbalance might appear in imbalanced human activity data, such as small sample size, class overlapping, and within-class imbalance that might worsen a supervised model's performance. The most common challenge met when performing classification using a supervised model is the class imbalance problem. Class imbalance significantly negatively affects supervised model performances, as represented classes tend to influence the performances of a supervised model than the underrepresented classes. Sampling is a popular methodology that can be applied to overcome the problem of class imbalance. The sampling method can create a dataset with a relatively balanced class distribution so that a supervised model can better learn to differentiate between the majority and the minority classes and then improve a supervised model's generalisation ability.

Hence, we compared six different sampling methods on their internal procedure to produce synthetic data for human activity recognition. The distance-based approaches were Synthetic Minority Over-sampling Technique (SMOTE) as well as Random_SMOTE algorithm, where these approaches mostly rely on utilizing K nearest neighbours in the procedure of oversample data. The second approach is oversampling and undersampling sampling algorithms, including Smote with Tomek links (SMOTE_Tomeklinks) algorithm and Modified Synthetic Minority Over-Sampling Technique (MSMOTE). These approaches applied both oversampling and undersampling in an algorithm procedure to oversample the training data. The other approaches were cluster-based sampling algorithm, Cluster- Synthetic Oversampling (CBSO) algorithm and Proximity Weighted Synthetic Oversampling Technique (ProWSyn). The clustering approaches were integrated in the procedure of oversampling the training data.

Overall, in this chapter we proposed three sampling methods to overcome the class imbalance in the human activity dataset, and the proposed methods which were designed to consider the issues of small sample size, class overlapping, and within-class imbalance. We refer to the first method as distance-based method (DBM), which was developed on distance-based techniques to solve the small sample issue. The second method was called noise detection-based method (NDBM), which was created on oversampling, and undersampling techniques, namely SMOTE_Tomeklinks and MSMOTE, was able to handle overlapping issues within the training data as the SMOTE_Tomeklinks and MSMOTE algorithm included the filtering step.

The final technique, which named the cluster-based method (CBM) was built on using cluster-based methods CBSO and ProWSyn in order to deal with the within-class issues and thus, enhanced the human activity classifier's performance.

We found that the proposed DBM, NDBM and CBM were more suitable than implementing the six existing sampling methods, including SMOTE, Random_SMOTE, SMOTE_Tomeklinks, MSMOTE, CBSO and ProWSyn, because combining sampling methods, namely DBM, NDBM and CBM lead to an increase in the data variability and improves the MLP generalisation ability.

The next chapter will present our work to introduce the fourth sampling methods. We utilize the Wasserstein Generative Adversarial Networks (WGANs) to produce sensor data.

Chapter 5

Generative Adversarial Networks (WGANs) to Generate Synthetic Sensor Data

5.1 Introduction

In the previous chapter, we aimed to improve the performance of a learning algorithm when trained with imbalanced human activity data by comparing six different sampling approaches and we also proposed three sampling methods. It was important to indicate that these sampling methods could only produce synthetic sensor features as they are not capable to work on time series data.

We found that sampling methods that use data features, which are typically extracted by applying a sliding window over the raw data, usually work too. In cases where the input is raw sensor data sampling methods might not fully consider the temporal dependencies intersecting in HAR. The human activity data is a sequence of sensor reading arranged consecutively to capture the activity information in the form of a time series.

Another method to generate synthetic data is the Wasserstein Generative Adversarial Networks (WGAN), which is capable of generating data from the raw sensor data and does not require extracting features to produce synthetic human activity data.

This chapter explores whether it is feasible to generate sensor data by applying the WGAN method in order to generate data from the raw sensor data and does not need extracting feature to produce human activity data.

In addition, we compare the WGAN method to the DBM, NDBM and CBM that were introduced earlier in the thesis. The previous chapter showed that we proposed three sampling methods. In this chapter, we frequently refer to these methods as (the three proposed sampling methods).

Note that the WGAN methods are taken from previously published work in the proceedings of International Joint Conference on Neural Networks (IJCNN-2020).[1] The DBM, NDBM and CBM are taken from work in submission to the Multidisciplinary Digital Publishing Institute (MDPI) Sensors journal 2021.[2]

[1] F. Alharbi, L. Ouarbya, and J. A. Ward, *"Synthetic Sensor Data for Human Activity Recognition"*, Proc. Int. Jt. Conf. Neural Networks, 2020.

[2] F. Alharbi, L. Ouarbya, and J. A. Ward, *"Comparing Sampling Strategies for Tackling Imbalanced Data in Human Activity Recognition*", this work is in submission to Multidisciplinary Digital Publishing Institute (MDPI) Sensors journal 2021

5.2 Proposed Method

In the following sections, we introduce the pre-processing steps to develop the WGAN methods. We also present the experiment settings and evaluation setup. We then introduce the obtain result and discuss them.

5.2.1 Data Pre-processing

In this chapter we implemented two types of input data in order evaluate the efficiency of our proposed models: raw input (e.g., direct accelerometer or gyroscope derived readings) and feature data input (extracted handcrafted features from the raw data, such as the mean over a sliding window). The feature data input was needed for the three proposed sampling methods because these methods did not work directly with time series data, such as sensor data.

Figure 5.1 shows the pre-processing pipeline for each of the two types of data. For both, the first step was to low-pass filter the data using a 3rd-order Butterworth filter [137]. We then calculated the root-sum-squared magnitude ($\sqrt{x^2 + y^2 + z^2}$) for each 3-axis sensor to ensure the data was invariant to the shifting orientation of the smartphones [118]. The data was then segmented into non-overlapping windows. For the raw data, each window was a matrix of size: length of the window $\times$ the number of sensor channels.

Five features were calculated over each window: *mean, standard deviation, minimum, maximum, median*, and *range*. These features are computationally not only cheap but also proven to be effective for HAR [138]. Each windows was then a matrix of size: number of extracted features $\times$ number of sensor channels. Both raw and feature data was then scaled using min-max normalisation [139].

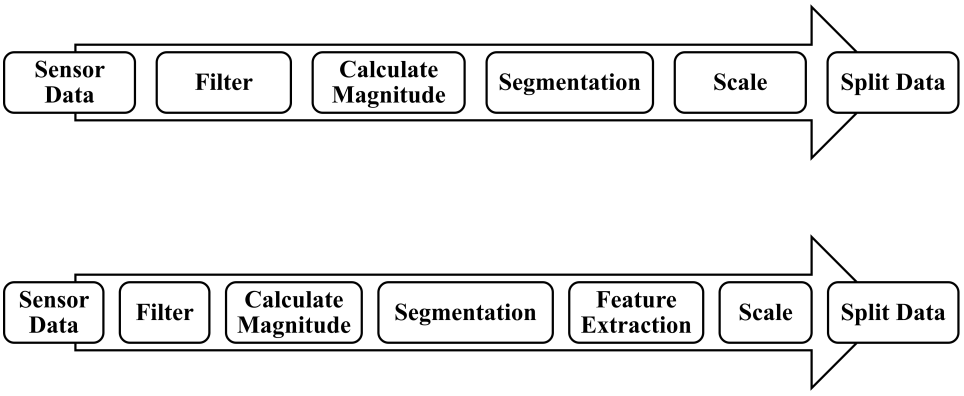

Figure 5.1: Pipelines for Raw (Top) Features (Bottom)

5.2.2 WGAN

To design a WGAN model, the type of activity affects the decision on which neural network architecture should be selected for the generator and discriminator [3] [28]. We performed extensive testing of several networks architectures and evaluations in order to select the appropriate network and the parameter values. The recommendation of HAR researchers also was considered in this thesis when identifying a suitable neural network for an activity. For example, Hammerla et al. [76] indicated that for periodic/repetitive activities that can exhibit periodicity, such as *running*, CNN is more appropriate. Filters of CNNs can capture the periodicity of an activity [140]. LSTM is suggested for static activities, such as being *still* or *sitting* [76]. The cells of the LSTM allow temporal dynamics to be captured within a processed activity data sequence [83].

A unified WGAN model may not be enough to learn several human activities distributions because human activities are heterogeneous [25]. Consequently, we constructed two different types of activity-specific WGAN model. The first created model was for relatively mouth to hand gestures (HMG) such as *smoking while in a group conversation*, as well as static activities lasting a relatively long time (e.g. *sitting*), and then created a second, more dynamic model for short-term activities such as *running*.

We fine-tuned a WGAN model on each activity class in order to obtain appropriate layers of the generator and the discriminator, the dimension of the noise vector, learning rate, and epochs. The hyperparameters for each model were determined over several trials and were validated using the validation set. Figure 5.2 presents the two models we used, the Model-1 has a generator based on one LSTM layer with 25 memory cells and uses a Tanh activation on its output. The generator's task was to generate data from the noise data with a similar structure to the real sensor data.

The discriminator had a single 1D-CNN layer using 10 filters with ReLU activation function and a dense layer with Tanh activation function. The output layer had a single neuron without an activation function. The discriminator's operation was to predict if its input was real or not based on its Wasserstein distance.

In contrast, the Model-2 had a generator which were constructed based on a 1D-CNN with 32 filters. The model used a dense layer with 32 units and the sigmoid activation function. We used dropout [141] with a rate of 50% and a dense layer with 8 units that used the sigmoid activation function. We then added a batch normalisation layer [77] and a dense layer with 4 neurons, which applied the sigmoid activation function. We again applied batch normalisation layer. The output layer of the generator was dense with the Tanh as the activation function.

The discriminator utilised 1D-CNN of 32 kernels with ReLU activation and a dense layer of 16 units with Tanh activation function. We also added a dense layer of one unit with the sigmoid activation function. The output layer was a further dense layer of one neuron but

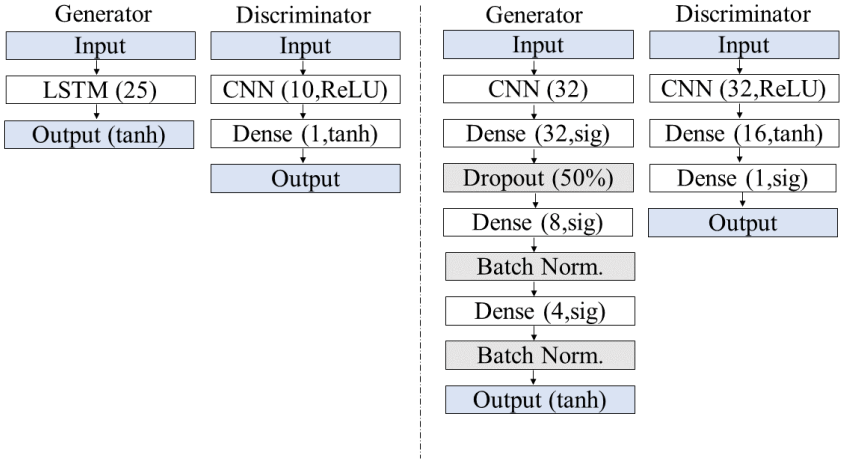

Figure 5.2: WGAN Moldel - 1 (Left) and WGAN Moldel - 2 (Right)

without an activation function.

5.2.3 Assessing Synthetic Sensor Data

To assess the synthetic sensor data, we utilised the common GAN-train and GAN-test techniques [142]. GAN-train entails training on synthetic sensor data but testing on real sensor data. High performance as a result of this training reveals that the GAN was capable of producing a realistic and diverse output and does accordingly not suffering from mode collapse [142]. By comparison, GAN-test was trained on real data and tested on synthesised data. This provided a complementary way to measure the quality of synthesised data [142].

Here, we evaluated two frequently applied classifiers, specifically 1D-CNN and LSTM. 1D-CNNs were created by stacking several processing units, including convolutional layers, pooling layers, and fully connected (dense) layers [143]. These stacked layers provide the 1D-CNNs with great ability to extract features automatically from raw sensor data. As a comparison, we also evaluated synthetic sensor data in a dynamic RNN-based model. We used LSTM, which could learn long-term dependencies by using a memory cell comprised of an input gate, output gate, and forget gate. LSTM is specifically designed to model temporal dynamics in sequences such as sensor data [144].

The categorical cross-entropy [77] was used as the loss function for training both 1D-CNN and LSTM classifiers. The training hyperparameters, comprising the number of epochs, learning rates, and optimiser functions, differed between datasets and classification tasks. When evaluating classes with a limited number of samples, for instance, the number of epochs had to be limited to avoid overfitting [77]. Several hand-tuning of these hyperparameters were consequently necessary.

5.2.4 1D-CNN Supervised Model

The CNN layout for n sensor streams could be seen in Figure 5.3. Each individual input sensor, such as accelerometer magnitude or extracted features from accelerator magnitude, was first passed to a single 1D-CNN layer [145]. The first layer used 9 filters with ReLU activation function. A dropout layer was then added with a rate of 50%. We also implemented a max-pooling layer (with a kernel of 2). The output of each subnet was then flattened, concatenated, and passed to a dense layer with 15 units with ReLU activation functions. The output SoftMax activation layer was finally used for classification [146].

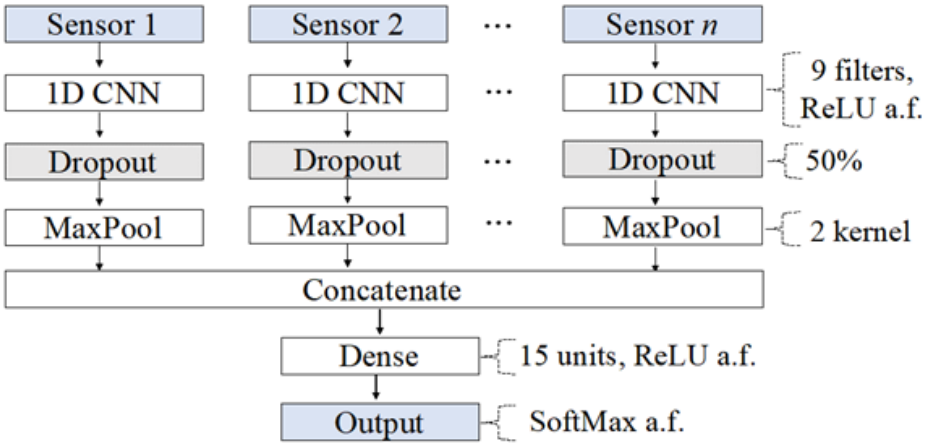

Figure 5.3: CNN model architecture

5.2.5 LSTM Supervised Model

The LSTM stacked layers in the second classification model displayed in Figure 5.4. Each sensor stream was then independently processed in order to capture longer temporary patterns. The LSTM layer uses 15 units and a Tanh activation function. We applied a dropout layer with a rate of 10%. Another layer of LSTM has 10 units and a Tanh activation function. The patterns from the individual pipelines were then concatenated together. We used a final dropout layer with a rate of 50%, and a dense layer with 8 neurons and a ReLU activation function. Finally, the output layer used a SoftMax activation function.

5.2.6 Oversampling Training Set with Synthetic Sensor Data

After we generated and evaluated the quality of the synthetic sensor data produced by WGAN, we used it to oversample the minority activity in the training set. We then evaluated the entire oversampled dataset using the two classifiers, 1D-CNN and LSTM. As a baseline comparison, we ran these classifiers using the original, imbalanced data.

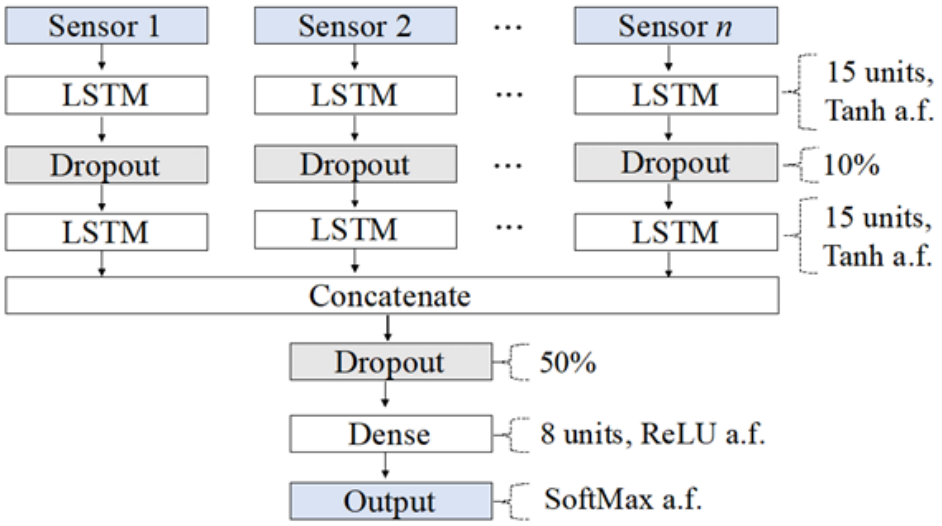

Figure 5.4: LSTM model architecture

5.2.7 Oversampling Training Set with Sampling Methods

We extracted the features mean, standard deviation, minimum, maximum, median, and range from the sensor data and we used the features sampling methods from previous chapter DBM, NDBM and CBM to oversample the least represented activity. Through feature extraction, we were able to compare the performance of these sampling methods to the WGAN method.

We made an evaluation of the three proposed sampling methods, and a baseline evaluation using features calculated from the original data. As before, we used 1D-CNN and LSTM classifiers.

5.2.8 Evaluation Method

The balanced F1 score (Macro F1 score) applied in order to treat classes equally, irrespective of how often a class appears (more details in section 2.5.4).

The dataset is divided into training, validation, and testing sets (70% for training, 15% for validation and 15% for testing), using the stratified split data method from scikit-learn[119]. This method balances the number of data samples of the classes in each split. The Python [119] and Keras [147] were used to implement the models.

The mean and standard deviation of macro F1 score, recalls, and precisions [33] were used to evaluate the performance of the classifiers, CNN and LSTM.

The trained WGAN networks were run for 30 times to generate synthetic data. We ran 30 trailers for each CNN model of all experiments [121] in sections 5.4.1, 5.4.2 and 5.4.3. The mean and standard deviation of the macro F1 scores, recalls and precisions [33] were then reported.

LSTM algorithm incurs high computational costs [148]. LSTM uses different compu-

tation gates [149], which results in more running time and the need for large computational resources [150]. However, as this thesis had limited computing resources and time, the LSTM was ran for 10 times [151]. The mean and standard deviation of the macro F1 scores, recall and precision according to 10 repetitions. It should be noted that the trained WGANs models were run 10 times to generate synthetic data.

5.2.9 Datasets

We used two datasets the Sussex-Huawei Locomotion (SHL) [16] and the Smoking Activities Dataset (Smoking) [17]. The reason we used these two datasets in this chapter is that WGAN required sufficient amount of training data to train and the two datasets comprise large quantities of data.

SHL records real-life activities of three subjects over three days, including eight different locomotion and transportation activities, including *walk, run, still, bike, car, bus, subway* and *train*. The subjects carried four smartphones at four locations (hand, hip, torso, and bag). We use data from 6 sensors including accelerometer, gyroscope, magnetometer, linear acceleration, orientation, and gravity, from both hand and hip. In addition, we used a subset of the dataset from the days where the same activities were performed by subjects one and three. These activities were *walk, run, still, bike* and *bus* [SHL]. Figure 5.5 (left) shows the proportion of each class in the dataset (with, e.g *run* making up 3% of samples).

The Smoking dataset was collected from 11 participants over 3 months. Each participant wore a smartwatch on the right wrist as well as a smartphone in the right pocket to capture data. These were embedded with accelerometers and gyroscopes, which were the sensors adopted in this chapter. The SHL dataset is sampled at 100 Hz.

The Smoking dataset was divided into three subsets according to the activities performed. We used only one of these (Subset 2) in this thesis since it included more activities and participants. Figure 5.5 (right) shows the imbalanced activity distributions for Smoking dataset. Subjects performed 8 activities, split into complex and simple. Complex activities contain *smoking while standing (SmokeST), smoking while sitting (SmokeSD), smoking while in a group conversation (SmokeGP), drinking while standing (DrinkST)* and *drinking while sitting (DrinkSD)*. The simple activities were *stand, sit* and *eat*. The Smoking dataset was sampled at 50 Hz.

We used data that collected from subjects who wore a smartwatch on the right wrist and a smartphone in the right pocket. These devices were embedded with accelerometers and accelerometers.

Both datasets were pre-processed by utilizing the pipeline explained above in section 5.2.1. The SHL data is low-pass filtered, using a 3rd-order Butterworth filter with a corner frequency 20Hz. This was then segmented using a 3 seconds window [94]. The SHL dataset

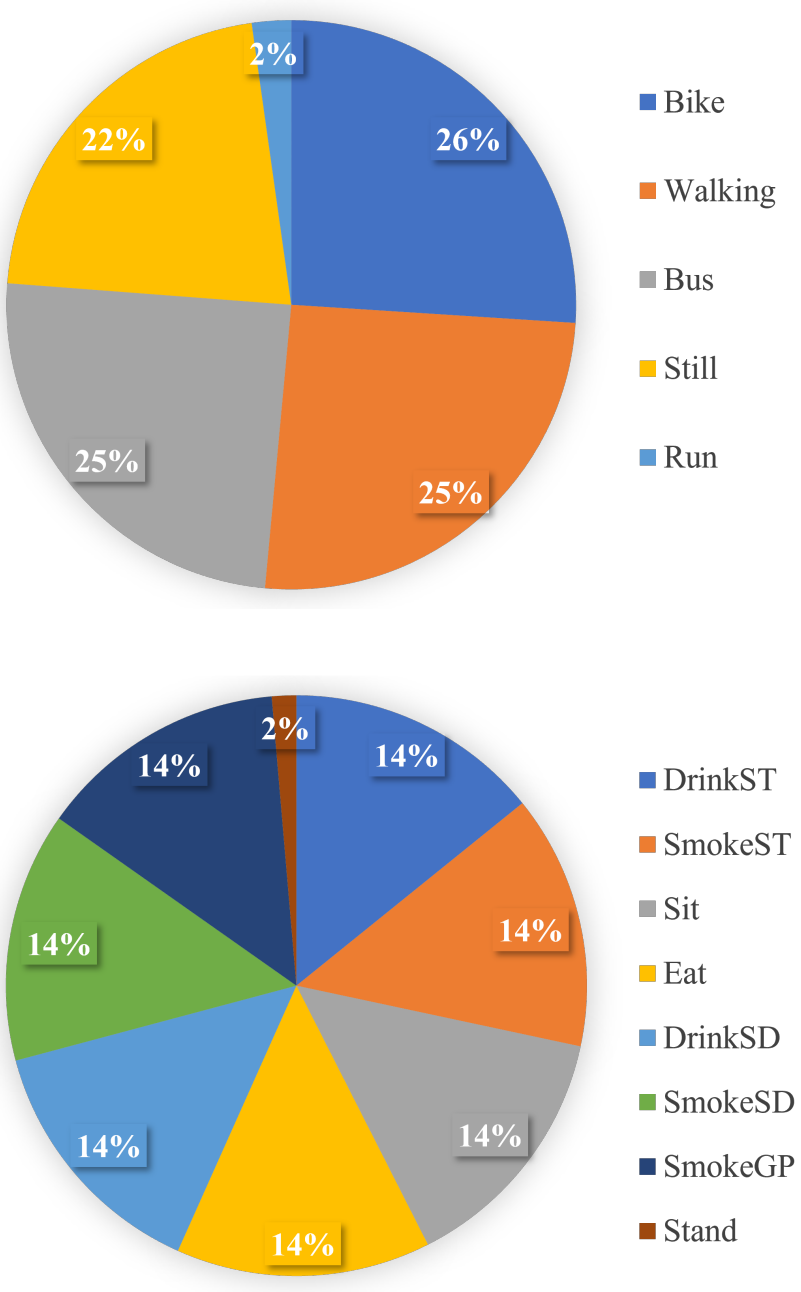

Figure 5.5: Distribution for SHL dataset (Top) and Smoking dataset (Bottom)

was sampled at 100Hz. This meant for each 1 second there would be 100 data points. So, for 3 seconds there would be 300 data points. For more details about sampling rate of a sensor see section 2.4.2. Furthermore, there were 12 sensor channels, hence the raw matrix size for each window was (300,12). We found that extracting feature from each window the shape of the matrix became (6,12). The total dataset size was 15280 windows ($\sim$ 13 hours).

The Smoking dataset was sampled at 50 Hz. We adjusted the low-pass filter with a corner frequency of 50Hz, with segment windows of 9 seconds. Here for each 1 second, there would be 50 data points. Hence, for 9 seconds there would be 450 data points. Given the 4

sensor channels, each raw data window was sized (450, 4). The feature-extracted matrix was then (6,4). The total dataset size for Smoking is 11776 windows ($\sim$ 30 hours).

Table 5.1: SHL dataset hyperparameters for WGAN Models

Activity	Noise Vector	Learning Rate	Epochs	WGAN Model
Bus	10	0.0005	1000	1
Run	5	0.03	1000	2
Still	10	0.0005	1000	1

Table 5.2: Smoking dataset hyperparameters for WGAN Models

Activity	Noise Vector	Learning Rate	Epochs	WGAN Model
SmokeGP	10	0.0005	1000	1
Stand	10	0.0005	1000	1

5.3 Evaluation Setup

We evaluated our approaches in different stages: first, we generated and evaluated the quality and diversity of the synthetic data that generated by WGAN. Then, we evaluated the raw synthetic data when used to oversample imbalanced datasets.

5.3.1 Evaluation of Synthetic Data

In order to assess the quality and diversity of our synthetic data we took two approaches: using generated data to train our classifiers, which we then evaluated using real test data (GAN-train), and by using real data to train, which we then evaluated on generated data (GAN-test). We thought it was imperative to generate the best quality data for each class, therefore, separate WGAN models were fine-tuned to match each minority class of interest. For the long-term static data in both datasets, such as *still* and *bus* (in SHL), and *smokeGP* and *stand* (in Smoking), the WGAN Model-1 worked best. The faster-changing data of *run* (SHL) was better characterised using Model-2. Table 5.1 and Table 5.2 exhibit the parameters used for each class model.

The SHL activity generator created 100 windows of synthetic sensor data for *still, bus*, and *run*. The Smoking activity generator generated 100 windows of synthetic data for two activities: *smokeGP* and *stand*.

Once generated, the data was evaluated using the two classifiers, 1D-CNN and LSTM, with the respective hyperparameters shown in Table 5.3.

Table 5.3: Hyperparameters for models assessing the quality of synthetic data for both datasets

Classifier	1D-CNN	LSTM
Optimiser	SGD	ADAM
Learning Rate	0.00001	0.0001
Epochs	15	15

Table 5.4: SHL dataset hyperparameters for classification

Classifier	1D - CNN		LSTM	
Input data	Raw Input	Feature Input	Raw Input	Feature Input
Optimiser	SGD	SGD	ADAM	ADAM
Learning Rate	0.001	0.001	0.001	0.0001
Epochs	20	25	35	50

Table 5.5: Smoking dataset hyperparameters for classification

Classifier	1D - CNN		LSTM	
Input data	Raw Input	Feature Input	Raw Input	Feature Input
Optimiser	SGD	SGD	ADAM	ADAM
Learning Rate	0. 0001	0.01	0.001	0.001
Epochs	20	50	50	50

5.3.2 Raw Data Oversampling Evaluation

In order to evaluate how our method be utilized in a real-world situation. The WGAN models were used to oversample each minority activity in the training set (*run* in SHL, *stand* in Smoking), and we used the new oversampled datasets to compare classifier performance. As a baseline, we evaluated the performances without oversampling.

5.3.3 Feature Data Oversampling Evaluation

In order to compare our proposed approaches, we extracted features from the raw sensor data and then compared two approaches. First, we evaluated a baseline using features from the imbalanced training set. Second, the DBM, NDBM and CBM were implemented to oversample the training set and to compare the performances of the sampling methods to the baseline.

5.3.4 Classifier Setup

Both the raw data and feature data oversampling evaluations were implemented using 1D-CNN and LSTM classifiers. Table 5.4 and Table 5.5 display the hyperparameters for the classification models on SHL and Smoking datasets, respectively.

Table 5.6: Classification performance of evaluation approaches GAN-Test and GAN-Train to assess the quality of synthetic data on the SHL dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 and 10 repetitions of 1D-CNN and LSTM respectively

Evaluation type	Classifier	Activity	F1 Score	Recall	Precision
GAN-Test	CNN	Bus	0.92 ($\pm$ 0.13)	1.00	0.88
		Run	0.78 ($\pm$ 0.40)	0.77	0.83
		Still	1.00 ($\pm$ 0.02)	1.00	0.99
	LSTM	Bus	0.93 ($\pm$ 0.14)	1.00	0.90
		Run	0.80 ($\pm$ 0.42)	0.80	0.80
		Still	1.00 ($\pm$ 0.00)	1.00	1.00
GAN-Train	CNN	Bus	0.89 ($\pm$ 0.06)	0.96	0.84
		Run	0.76 ($\pm$ 0.17)	0.82	0.77
		Still	0.82 ($\pm$ 0.21)	0.75	0.93
	LSTM	Bus	0.61 ($\pm$ 0.32)	0.78	0.6
		Run	0.39 ($\pm$ 0.26)	0.37	0.64
		Still	0.46 ($\pm$ 0.26)	0.45	0.72

5.4 Results

5.4.1 Evaluating the Synthetic Data

Table 5.6 and Table 5.7 illustrate the GAN-Test and GAN-Train classifier results applied to evaluate the diversity and quality of the new sensor samples. GAN-Train meant we trained on synthetic and tested on real data), whereas GAN-Test we trained on real and on synthetic data [142].

These tables indicated that the characteristics of the synthesised data strongly match real data. The tables also show samples were relatively diverse. However, Table 5.6 indicates exception of the *run* class when using 1D-CNN for GAN-Train. The F1 scores of the *run* was low compared with other activities. Similarity, Table 5.7 shows the F1 score of the *stand* class using LSTM for GAN-test was low.

5.4.2 Rebalancing the Training Set with Raw Data

Table 5.8 indicates that the WGAN improved 1D-CNN 's performance, but the LSTM did not significantly benefit from the WGAN. The table also shows the SHL dataset results for the 1D-CNN and LSTM using raw sensor data. The baseline F1score for 1D-CNN was 0.85%, which increased to 0.91% after oversampling the minority class *(run)* with 100 synthetic samples. We realized that the recall was improved by 6% . The precision also was increased by 3%. However, the LSTM results were not enhanced by oversampling.

Similarly, on the Smoking dataset results shown in Table 5.9, oversampling the minority

Table 5.7: Classification performance of evaluation approaches GAN-Test and GAN-Train to assess the quality of synthetic data on the Smoking dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 and 10 repetitions of 1D-CNN and LSTM respectively

Evaluation type	Classifier	Activity	F1 Score	Recall	Precision
GAN-Test	CNN	SmokeGP	0.71($\pm$ 0.09)	1.00	0.55
		Stand	0.15 ($\pm$ 0.32)	0.13	0.20
	LSTM	SmokeGP	1.00 ($\pm$ 0.00)	1.00	1.00
		Stand	1.00 ($\pm$ 0.00)	1.00	1.00
GAN-Train	CNN	SmokeGP	0.93 ($\pm$ 0.14)	0.91	0.99
		Stand	0.75 ($\pm$ 0.33)	0.88	0.77
	LSTM	SmokeGP	0.84 ($\pm$ 0.33)	0.83	0.93
		Stand	0.65 ($\pm$ 0.41)	0.78	0.82

Table 5.8: Comparing the performance of baselines (1D-CNN and LSTM) and the proposed WGAN on the SHL dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 and 10 repetitions of 1D-CNN and LSTM respectively. The largest obtained scores are in bold font

Classifier	Method	F1 Score	Recall	Precision
1D-CNN	baseline	0.85 ($\pm$ 0.05)	0.83	0.93
	WGAN	**0.91** ($\pm$ 0.05)	**0.89**	**0.96**
LSTM	baseline	0.90 ($\pm$ 0.09)	0.90	0.93
	WGAN	0.90 ($\pm$ 0.08)	0.88	0.92

Table 5.9: Comparing the performance of baselines (1D-CNN and LSTM) and the proposed WGAN on the Smoking dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 and 10 repetitions of 1D-CNN and LSTM respectively. The largest obtained scores are in bold font

Classifier	Method	F1 Score	Recall	Precision
1D-CNN	baseline	0.65 ($\pm$ 0.08)	0.66	0.67
	WGAN	**0.70** ($\pm$ 0.07)	**0.72**	**0.73**
LSTM	baseline	0.73 ($\pm$ 0.06)	0.74	0.75
	WGAN	0.70 ($\pm$ 0.04)	0.72	0.71

Table 5.10: Comparing the performance of baselines (1D-CNN and LSTM) and the proposed DBM, NDBM and CBM on the SHL dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 and 10 repetitions of 1D-CNN and LSTM respectively

Classifier	Method	F1 Score	Recall	Precision
CNN	Baseline	0.91 ($\pm$ 0.02)	0.88	0.96
	DBM	0.92 ($\pm$ 0.02)	0.90	0.96
	NDBM	0.92 ($\pm$ 0.02)	0.90	0.96
	CBM	0.91 ($\pm$ 0.01)	0.89	0.96
LSTM	Baseline	0.88 ($\pm$ 0.04)	0.86	0.95
	DBM	0.87 ($\pm$ 0.06)	0.85	0.93
	NDBM	0.87 ($\pm$ 0.05)	0.85	0.95
	CBM	0.86 ($\pm$ 0.06)	0.84	0.91

class *(stand)* by adding only 100 new samples exhibited an increase of 5% in overall F1 score, 6% in the recall, and a small enhancement on the precision by 4%. However, there was no significant improvement in the LSTM.

5.4.3 Rebalancing the Training Set with Feature Sampling Methods

The DBM, NRBM, and CBM appeared less efficient then WGAN in improving the performance of the 1D-CNN and LSTM.

On the SHL dataset, three proposed sampling methods showed similar results on the F1 score and the recall but did not improve the precision of the 1D-CNN by adding 50 new samples to the training set. For example, Table 5.10 indicates that WITH DBM and NRBM the F1 score and the recall never improved by more than 3%, and the precision was unchanged and by remaining at 0.96%. Likewise, the LSTM did not benefit from the sampling methods, and the performances on the recall and precision dropped by with the DBM and CBM.

On the Smoking dataset, Table 5.11 suggests that 1D-CNN performances also appeared to be positively unaffected by the three proposed sampling methods when they used to add 50 new samples to the training set. In addition, as Table 5.11 shows, no significant increase in the LSTM's performance was found when any one of the three proposed sampling methods was

Table 5.11: Comparing the performance of baselines (1D-CNN and LSTM) and the proposed DBM, NDBM and CBM on the Smoking dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 and 10 repetitions of 1D-CNN and LSTM respectively

Classifier	Method	F1 Score	Recall	Precision
CNN	Baseline	0.92 ($\pm$ 0.01)	0.92	0.93
	DBM	0.92 ($\pm$ 0.01)	0.91	0.92
	NDBM	0.92 ($\pm$ 0.01)	0.92	0.92
	CBM	0.92 ($\pm$ 0.01)	0.92	0.92
LSTM	Baseline	0.80 ($\pm$ 0.03)	0.79	0.82
	DBM	0.80 ($\pm$ 0.04)	0.79	0.82
	NDBM	0.80 ($\pm$ 0.04)	0.79	0.82
	CBM	0.80 ($\pm$ 0.03)	0.79	0.82

applied.

5.4.4 Comparing Class-Wise Recognition When the Training set was Oversampled

We compared the influence of the WGAN and on the F1 score of the 1D-CNN and LSTM. This to show how the WGAN methods increased the recognition of the least represented activity. We did not show the influence of the DBM, NRBM and CBM on class-wise. This is because table 5.12 and 5.13 show no significant improvements on 1D-CNN and LSTM when we applied the DBM, NRBM and CBM.

Table 5.12 shows the F1 score of the 1D-CNN improved to identify the *run* activity, which was the least represented activity on the SHL dataset. The F1 score and recall improved by more than 20% using WGAN. The precision also improved by 10%.

Table 5.12 illustrations that the WGAN was not able to show a significant improvement in the F1 score of the LSTM ability to recognise the *run* activity on the SHL dataset.

On the Smoking dataset, due to applying the WGAN there was a significant improvement in the all evaluation metrics of 1D-CNN in identifying the underrepresented activity *(Stand)*. Table 5.13 shows they were improved by more than 20%. In contrast, the F1 score of the LSTM did not increased to identify the *Stand activity* by implementing WGAN.

Table 5.12: Comparing the performance of baselines (1D-CNN and LSTM) and the proposed WGAN to identify the minority class (the *Run* activity) in the SHL datasett. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 and 10 repetitions of 1D-CNN and LSTM respectively. The largest obtained scores are in bold font

Classifier	Method	F1 Score	Recall	Precision
1D-CNN	baseline	0.35 ($\pm$ 0.25)	0.24	0.80
	WGAN	**0.64** ($\pm$ 0.22)	**0.51**	**0.90**
LSTM	baseline	0.60 ($\pm$ 0.40)	0.53	0.77
	WGAN	0.58 ($\pm$ 0.33)	0.50	0.73

Table 5.13: Comparing the performance of baselines (1D-CNN and LSTM) and the proposed WGAN to ideifyt the minority class (the *Stand* activity) in the Smoking dataset. The reported mean of F1 scores and ($\pm$ standard deviation), recalls and precisions were obtained from 30 and 10 repetitions of 1D-CNN and LSTM respectively. The largest obtained scores are in bold font

Classifier	Method	F1 Score	Recall	Precision
1D-CNN	baseline	0.32 ($\pm$ 0.42)	0.28	0.40
	WGAN	**0.75** ($\pm$ 0.35)	**0.71**	**0.83**
LSTM	baseline	0.78 ($\pm$ 0.29)	0.74	0.85
	WGAN	0.77 ($\pm$ 0.29)	0.84	0.73

5.5 Discussion

By using raw synthetic sensor data produced by WGAN, oversampled minority activities in imbalanced training data were promising in their ability to boost classifier performance compared to the DBM, NDBM and CBM. However, the choice of classifier plays a role in how well this may work.

The 1D-CNN evaluation revealed just how well our synthetic data oversampling method (WGAN) could work on raw data. When trained on the baseline case of imbalanced raw data, the 1D-CNN classifier tended to miss underrepresented classes (see the low baseline F1 score rates for *run* in (table 5.12) for the SHL dataset, and *stand* in (table 5.13) for the Smoking dataset. However, performances improved considerably when these classes were oversampled using WGAN, with the F1 score for *run* rising from 0.35% to 0.64%, and *stand* from 0.32% to 0.75%. In addition, recognition results of the 1D-CNN were consistent across all experiments, which was exhibited by the relatively small standard deviation (e.g. in tables 5.8 and 5.9).

Despite these promising results, the LSTM-based evaluation results were not as clear-cut as they were for 1D-CNN. It is unlikely that LSTM was positivity influenced by oversampling methods such as WGAN, DBM, NRBM, and CBM. This was because that the LSTM baseline results were already relatively high, such as on the SHL dataset, the F1 score of *run* activity in (table 5.12) was 0.60% for raw input. Likewise, in (table 5.13), the F1 score of *stand* activity was 0.78% for raw input on the Smoking dataset. Further study requires to be done to ascertain why our WGAN oversampling performance was more unsatisfactory when LSTM was used as the classifier instead of the 1D-CNN.

The proposed oversampling methods such as DBM, NRBM, and CBM seemed inefficient for relatively high dimension data. In this chapter we used data from multiple sensors and DBM, NRBM, and CBM did show any significant improvement of 1D-CNN and LSTM performance. Table 5.10 and 5.11 indicate the poor performance of the DBM, NRBM, and CBM. The reason might be due to the use of KNN by DBM, NRBM, and CBM. It is challenging for the KNN to show superior performance with high dimension data [152]. KNN uses euclidean distance which might not be appropriate metric for comparatively large data [153].

5.6 Summary

In this chapter, we introduced the use of a Wasserstein Generative Adversarial Network (WGAN) to generate sensor data for human activity recognition. We investigated WGAN on 5 different classes of human activity that were underrepresented across two publicly available datasets. We then evaluated the diversity and quality of the generated synthetic sensor data and found in most of the cases the F1 scores of over 55% when a 1D-CNN classifier is trained on synthetic and tested on real data, and of 50 % when it was trained on real data and tested on synthetic. We also oversampled imbalanced training sets using synthetic data and found overall F1 scores performance improvements of 5% using 1D-CNN classifiers on raw data. However, we found by comparing 1D-CNN-classified against features produced using the DBM, NRBM, and CBM no significant improvement. Also, similar evaluations using LSTM found no immediate advantage from our methods. In short, sampling methods are encouraging to improve the performance of CNN particularly the WGAN when train with imbalanced sensor data.

Chapter 6

Discussion and Conclusion

The last chapter discusses the main advantages and disadvantages of the sampling approaches proposed in this thesis. It addresses the limitations of the experiments and provides recommendations for future research in order to improve the proposed sampling methods.

6.1 Discussion

Prior studies such as [1] and [25] have highlighted the lack of works that address and investigate the impact of the class imbalance problem in human activity recognition. Our present thesis fills this gap by proposing four approaches based on both traditional machine learning and deep learning to reduce the class imbalance and substantially improve human activity recognition (HAR) performance.

There are six sampling methods based on traditional machine learning that we used in this thesis, Synthetic Minority Over-sampling Technique (SMOTE), Random_SMOTE algorithm, Smote with Tomek links (SMOTE_Tomeklinks), Modified Synthetic Minority Oversampling Technique (MSMOTE), Cluster-Based Synthetic Oversampling algorithm (CBSO), and Proximity Weighted Synthetic Oversampling Technique (ProWSyn).

Different issues might arise related to class imbalance, for instance, small sample size, class overlap and within-class imbalance. Consequently, we used these six sampling methods in order to develop three sampling techniques: DBM, NDBM and CBM. These methods are limited to generating features as they cannot operate on raw sensor data (time series data) [51].

We also considered Wasserstein Generative adversarial networks (WGANs) which are based on deep learning and can be used for raw sensor data generation due to convolutional and recurrent structures in the networks.

Our findings further support the argument of Chen et al [58]. They argued that the data sampling technique should be considered because of the different activities' distributions which can hinder the performance of human activity recognition models. Consequently, a classifier performance can be improved and attain more training data for activity recognition. Although Chen et al only used the SMOTE method in their work, their results were promising. In response to such findings, we compared and showed the potential of applying six different sampling methods. In addition, we also proposed four different approaches (the DBM, NRBM, CBM and WGAN method) to deal with challenges that introduced by the class imbalance. We also showed these methods enhanced the performance of human activity recognition. Our results provide more comprehensive methods for the class imbalance issue in human activities than the aforementioned studies.

We introduce the DBM, which is a combination of two sampling algorithms, SMOTE and Random_SMOTE, in order to deal with a small size sample by producing synthetic data. The most important benefit of this proposed method is that it does not involve any computation complexity when generating synthetic data. These sampling methods also did not perform any data filtering, such as noise detection, or apply cluster processes to the training data, making them less complicated algorithms in terms of the computation process requirement. Figure 4.10 indicates that the DBM was the fastest compared to the NDBM and CBM. We suggest the DBM in instances where the training data is suffering from small sample size

problem. However, due to the lack of such processes, the quality of the produced synthetic data may not be optimal because this method may also generate noise samples from the training data.

One way that we recommend improving the proposed DBM is to use a technique to assess the similarities between the synthetic samples and the training data samples (i.e. the original data). Then, one can use only the most similar synthetic samples to the original sample in order to oversample the training data. For example, one can use the SMOTE and Random_SMOTE approaches to generate synthetic samples from the original training samples and use an efficient similarity metric such as Euclidean distance to compare the generated synthetic data to the original training samples. Then, one utilise only the most similar synthetic samples and disregard the least similar. Our reason for this is that it might ensure that high-quality synthetic samples are used to oversample the training data.

The second proposed sampling approach, the NDBM, combines the SMOTE-Tomek and MSMOTE sampling algorithms. The main benefit of the SMOTE-Tomek and MSMOTE apply data cleaning techniques that can deal with issues such as the class overlapping that may be introduced from windowing methods for sensor data segmentation (see section 2.4.4). The key advantage of this proposed sampling method is that it is capable of tackling both class overlap and noise. This method includes applying data filtering techniques intend to identify and remove mislabelled as well as noisy data, and to clean possible overlapping between classes (as detailed in section 4.4.3.2). One limitation of the second proposed method is that SMOTE-Tomek oversamples the minority samples using SMOTE prior to the cleaning procedure (see section 3.2.3), which might lead to oversampling the noise in the minority data. This may increase and retain noisy samples after the cleaning procedure. Thus, Tomek Links might not identify the noisy samples as noise to remove. This may be due to their neighbourhood being changed. As a result, the new synthetic sample might include noisy samples or poor quality synthetic data. A further study may consider replacing the SMOTE-Tomek sampling technique with another sampling method that can perform data cleaning and then oversampling the minority class. This is recommended to determine if this can improve the produced synthetic data and lead to an improved performance of the introduced NDBM.

We also proposed a sampling approach named the cluster-based method (CBM), which combines the CBSO and ProWSyn sampling algorithms. This approach's main advantage is that it can handle the issue known as intraclass variability that it can lead to within-class imbalance. The intraclass variability describes a case when the an activity is performed in a distinct way by the same person. The CBSO and ProWSyn sampling methods apply a clustering procedure to identify minority class within-class imbalance before performing oversampling to the training data. The CBM can consider the structure of minority class samples due to it using clustering approaches (see detailed in section 3.2.4), which is of great benefit when compared

to the DBM and NDBM. Table 4.11 shows that the CBM achieved slightly superior results (for example, the F1 scores) in most cases when compared to the DBM and NDBM due to the clustering property. We suggest applying the CBM when a human activity training data suffers from a small sample size as it showed encouraging promise in improving the performance of the MLP classifier on the opportunity and PAMPA2 datasets (see Table 4.11). However, this fundamental property of the CBM may also lead to increase the computational cost. In Figure 4.10, the CBM demonstrates the longest running times compared with the DBM and NDBM.

When comparing our findings to those of studies where complex human activity models were used to achieve great accuracy, we show that the DBM, NDBM and CBM are promising in achieving high accuracy with less implementation complexity. Regarding the opportunity data, Ordóñez et al. [83] introduced a classification model that combined the CNN and LSTM layers as well as compared it with the CNN classifier (baseline) to recognise human activity by using data from multiple sensors for instance, accelerometer and gyroscope. They demonstrated that the F1 score of the CNN was 0.78%. The proposed combined model improved the recognition of human activity by 9% (the F1 of the combined approach was 0.87%). By comparison, we evaluated our sampling methods using data from a single sensor, particularly the accelerometer, to recognise human activity, and showed that the proposed methods improved performance by more than 10% (see Table 4.11).

Furthermore, on the ADL dataset where the data is collected using a single sensor (accelerometer), Erdas et al. [93] compared the performance of deep learning classifiers such as CNN and another classifier that was based on combining CNN and LSTM layers to recognise human activities. They showed that both classifiers exhibited similar performances. Both CNN and the combined CNN and LSTM layers achieved an accuracy score of 0.91%. In contrast, we assessed DBM, NDBM and CBM in distinguishing human activity, and demonstrated that our proposed sampling methods enhanced performance of the MLP and obtained an F1 score of 93% (see Table 4.11) using a less complex model.

We suggest that future researchers apply and compare the proposed DBM, NDBM and CBM when facing the challenge of emergent or unanticipated activities to produce more samples of such activities. Chen et al. [25] argued that to train as well as evaluate a supervised learning algorithm to recognize human activity requires a large quantity of annotated data samples. Chen et al. also indicated that there are not only emergent activities but also unanticipated activities such as an accidental falling down is extraordinarily challenging to acquire. Furthermore, the authors mentioned this may lead to a class imbalance issue. The DBM, NDBM and CBM can enable the researcher to overcome this challenge and consequently, use a supervised learning algorithm in their research.

In term of the Wasserstein Generative adversarial networks (WGANs), the proposed

method can enhance the performance of the CNN to learn from imbalanced human activity datasets. When the WGAN approach was used to oversample the training data and improve learning from imbalanced human activity datasets, the convolutional neural network classifier's performance increased by more than 5% (see section 5.4.2).

The main issue with the WGANs that they require large quantity of training data. They also do not perform well when dataset suffer from the small sample size issue. Performance is worse when multiple activities are underrepresented. For example, there are multiple classes (activities) on the Opportunity and PAMAP2 datasets with only few samples, as shown in Figure 4.2 and Figure 4.3. We could only apply the WGAN approach to the smoking activity and SHL datasets because these datasets had substantial amounts of sensor data. The datasets were collected for long periods, three months or more, to enable researchers to conduct studies with a sufficient amount of labelled sensor data. However, in the real world, it might not be feasible to obtain large quantities of labelled sensor data as the process of labelling data is costly as well as time consuming. Consequently, this thesis further supports the argument in [26] on the possibility of using data augmentation techniques such as generative adversarial networks for human activity in the real-world remains challenging due to the quantity of the wearable human activity training data.

In term of the number of used sensors in order to train the proposed methods, the DBM, NDBM and CBM improved the performance of the MLP in learning from different imbalanced human activity datasets using data from a single sensor (accelerometer). The performance increased by more than 5% (see Table 4.11). The three proposed sampling methods did not require a large amount of labelled sensor samples to work when compared with the WGAN. In order to train the WGAN to create synthetic sensor data, we used data from multiple sensor modalities (e.g. accelerometer, gyroscope, and magnetometer). However, using data from multiple sensors seem not always promising approach. We compared the performance of the WGAN and the other proposed sampling methods to generate sensor data using data from multiple sensor modalities (see section 5.4.4), and found that their performances, when used to oversample the training data, were not encouraging. The WGAN could not significantly impact the performance of the LSTM, as shown in Table 5.8 for the SHL dataset and in Table 5.9 for the smoking dataset. Likewise, the DBM, NDBM and CBM were also not helpful in enhancing the performance of the LSTM when we used the data from multiple sensor modalities (see Table 5.10 for the SHL dataset and 5.11 for the Smoking dataset).

We would recommend using the proposed DBM, NDBM and CBM, rather than data augmentation methods that are based on deep learning for two reasons.

The benefits of using the DBM, NDBM and CBM are firstly that they require less training data compared with the WGAN method. In order to train the WGAN methods training data from multiple sensors is needed whereas with DBM, NDBM and CBM data from single

sensor is required. Therefore, we agree with Lago et al. [33] that when a human activity model rely on training data from single sensor is more applicable for real world scenario as it not always feasible to obtain human activity from multiple sensors.

Secondly, we proposed two different types of activity-specific WGAN models. The first WGAN model was created for hand-to mouth (HMG) activities such as *Smoking while in a group conversation* and static activities lasting a relatively long time (e.g. *sitting*). The second WGAN model was formed for more dynamic, short- term activities such as*running*. Therefore, more effort in time was required to explore the suitable WGAN architecture, such as using 1D-CNN or LSTM and optimising hyperparameters related to each network architecture, as they were significantly influencing performances. In contrast, the other proposed sampling methods are not limited to specific activities. In this thesis we showed that the DBM, NDBM and CBM can be used for datasets that include different type of activities such as ambulation, or daily activities (e.g. *running, walking, cleaning a table* or *open a drawer*).

The following section will discuss the limitations of the proposed sampling approaches and provide several directions for future works to enhance these introduced sampling methods.

6.2 Limitations and Future Work

One of the limitations of the proposed DBM, NDBM and CBM is the features that we used. We chose to adopt time-domain features because they are efficient and quick to compute. Consequently, this work could be extended with more focus on applying different features. For example, the results can be validated using a different type of common extracted feature such as frequency-domain features and then comparing them with our obtained results. Also, in some cases, the DBM, NDBM and CBM's performances are similar, as Table 4.11 illustrates for the ADL dataset. So, future studies may opt to investigate how different features can influence which of the DBM, NDBM and CBM works best. This also might help researchers in selecting the most appropriate of the DBM, NDBM and CBM.

Manually designing hand-crafted features in HAR applications usually requires domain knowledge and generally a heuristic process [154]. In contrast, the main benefit of the deep learning method is that it can automatically learn and find patterns from raw sensor data [25]. They also does not rely on human expertise and further effort in feature extraction [155]. Therefore, future studies could explore the use of a deep learning method, for example, the convolutional neural network (CNN) to extract features. The CNN performs convolutional operations to extract features automatically (as shown in section 2.5.3.2) and one does not need to spend time and effort extracting and selecting appropriate features manually. Ordóñez et al. [83] stated that CNN compared to handcrafted feature extraction has the potential for extracting robust features when a limited raw sensor data. This can make the features extracted by the CNN more robust compared with experimental feature design that requires domain knowledge in order to identify features manually.

We suggest for future research that the DBM, NDBM, CBM might be applied to over-sample the training data that its features are extracted using CNN. Accordingly, we can determine how the proposed method benefits from incorporating the CNN approach as a feature extraction technique compared with the handcrafted feature extraction. This idea is strongly recommended because automatically extracted CNN features can be of better quality than hand-crafted features [140]. As we mentioned earlier that the features driven from CNN more robust when compared with manually extracted features. This can be a potentially useful method in improving the human activity model's ability to learn from imbalanced human activity data..

Another limitation is that we used all the six sampling methods and the MLP's classifier in their default settings. Exploring and determining how different parameter settings can influence the sampling methods and the MLP's performance was not included in this thesis. Our scope was to demonstrate the capability of using sampling methods to increase the MLP's generalisation ability for activity recognition. Future investigations could find out the influence of several parameter settings on the sampling methods and aim to optimise the MLP parame-

ters. The reason is to to provide researchers with recommendations to enhance the sampling methods and the MLP's performance.

In addition, Table 4.16 shows the ANOVA test indicted that all sampling methods performed the same on the PAMPA2 dataset.By exploring different parameters of sampling methods this might help to enhance the performance of the sampling methods. Then, one might be able to find which sampling methods is more statistically significant on the PAMPA2 dataset.

The DBM, NDBM, CBM were evaluated using three different human activity datasets and we showed in chapter 4 that these methods are capable of improving MLP in its learning from imbalanced human activity datasets. However, in the future, more work should be done with different types of datasets in order to generalise these three sampling methods to other domains. For example, one can evaluate these methods using image data or medical data.

The NDBM demonstrated acceptable results (see Table 4.9). However, further work to improve it is required. The overlapping issue is likely to occur in human activity data when applying sliding windows while pre-processing. Therefore, different sizes of the segmentation window should be applied and compared in order to explore how this affects the viability of the method. It would also be useful to determine how to measure class overlap before applying the NDBM, so that one may explore how it performs differently with varying degrees of class overlapping.

The within-class imbalance, which is a longstanding challenge for human activity recognition, may occur because of intraclass variability. The proposed CBM is therefore useful because it includes a clustering step when producing synthetic samples. This thesis shows that CBM is promising as it increased the MLP's generalisation ability when learning from an imbalanced human activity dataset. However, there is abundant room for a future study to determine the influence of the intraclass variability on the CBM's performance. For example, one can use data that belong to the same individual's activity data to train and test a single classification model. Then, one might incorporate CBM with the classification model to evaluate how it impacts the classifier's performance. The intraclass variability degree may be lowered all training and testing data comes from a single individual when compared with using multiple individuals' data for training and testing a recognition model.

In order to determine the most useful sensor position for applying sampling method. Also, how the sensor position on the body impact the sampling techniques performances. It is recommended for future investigation.

The two datasets used for evaluating the WGAN method are fairly diverse and cover a relatively wide variety of human activities. However, further study needs to be carried out to investigate the use of this method on a wider variety of classes and datasets.

We did not perform statistical significance test for WGAN method because we used different input data to each sampling method. For WGAN the input was raw data whereas

for the other methods input was features data. A further research could apply another stable GAN variant method such as WGAN-PG [47]. This is to conduct fair statistical significance of WGAN and WGAN-PG as sampling methods. We also suggest using another classifier with the 1D-CNN and LSTM such as MLP. This is to statistically evaluate the significance of how the WGAN, and WGAN-PG improve the performance on more supervised methods namely 1D-CNN, LSTM and MLP [4]. In addition, the proposed WGAN, DBM, NDBM and CBM demonstrated inadequate performances on the LSTM. For instance, Table 5.8 and Table 5.9 show the F1 score of the LSTM did not benefit of these sampling methods on both datasets. Consequently, we were not encouraged to implement statistical significance test.

In the future, we intend to explore the computational complexity of applying WGAN for human activity oversampling. Finally, as there are currently no widely recognized approaches or frameworks to assess synthetic sensor data, the work in this thesis makes some promising steps, upon which we will investigate further in future work.

6.3 Conclusion

To summarise, this thesis aims to address the problem of class imbalance in human activity recognition (HAR) and investigates several sampling solutions to overcome this challenge. Supervised learning methods are commonly used for HAR, and they often require labelled training data. One of the main characteristics of sensor data is that its quality is occasionally insufficient because for example, an individual not wearing a sensor or a malfunctioning sensor. These issues can often make the human activity data imbalanced. The combination of class imbalance and inadequate data quality can result in the ineffective recognition performance of a learning algorithm. The thesis concluded that the sampling solutions based on traditional machine learning and deep learning could generate synthetic sensor data from imbalanced human activity training data. In addition, using synthetic samples to oversample the training data improved the generalisation capacity of learning algorithms particularly MLP and CNN.

Previous research has rarely investigated the class imbalance in HAR or proposed solutions to overcome the issue [25], [4] and [26]. In particular, there is limited literature that investigate using sampling solutions to tackle the class imbalance in HAR. This thesis fills the gap of tackling the class imbalance problem in HAR. It also illustrates the significance of sampling methods to reduce class imbalance by proposing four sampling approaches.

First, we used six existing sampling methods based on traditional/shallow machine learning to propose three new hybrid approaches to deal with issue of the class imbalance in HAR. The main intrinsic property of these sampling methods is that they do not capture time-series data structure or operate on raw sensor data. However, they are capable of operating on extract features. We combined the Synthetic Minority Oversampling Technique (SMOTE) and the Random_SMOTE to create the first hybrid approach, called the distance-based method (DBM). The second hybrid approach was built using Tomek links (SMOTE_Tomeklinks) and the Modified Synthetic Minority Oversampling Technique (MSMOTE) and was called the noise detection-based method (NDBM). For the third hybrid approach, we combined the Cluster-Based Synthetic Oversampling (CBSO) algorithm and the Proximity Weighted Synthetic Oversampling Technique (ProWSyn), which we dubbed the cluster-based method (CBM).

We showed the usefulness of the Proposed DBM, NDBM and CBM to improve the generalisation ability of MLP to recognise human activity. We compared the proposed approaches with the six existing sampling techniques and the original training dataset without sampling. We assessed these methods on three public datasets: Opportunity, Physical Activity Monitoring (PAMAP2), and Activity Recognition from a Single Chest-Mounted Accelerometer (ADL).

Second, a fourth new sampling technique based on the Wasserstein Generative Ad-

versarial Network (WGAN) was introduced. Compared to the DBM, NDBM and CBM, the WGAN approach is based on deep learning, can operate on raw sensor data because it comprises convolutional and recurrent structures. We demonstrated that the proposed WGAN approach could generate raw sensor data and overcome the limitations of the shallow machine learning based sampling methods, which only work on extract features. We used two public datasets, the Sussex-Huawei Locomotion (SHL) and the Smoking Activities (Smoking) datasets, to assess the proposed WGAN technique for raw sensor data generation and demonstrated it improved CNN performance learning from imbalanced human activity data.

The small sample size problem often arises in human activity data because some activities are performed more than others in real life. Learning algorithms' performance might drop because there is not enough data to generalise unseen samples. Based on this thesis' conclusions, we recommend that sampling methods based on traditional machine learning are used when imbalanced human activity data has a small sample size. Thus, the proposed DBM, NDBM and CBM do not need a large quantity of training data to create synthetic data. Moreover, they require less time to optimise the hyperparameters than the WGAN approach. The WGAN method is more beneficial for extensive imbalanced human activity training data as deep learning methods are more useful when training with a large quantity of data. The WGAN method also needs more human effort in term of time to optimise the WGAN construction's hyperparameters because they can substantially affect performance.

REFERENCES

[1] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, 2014.

[2] D. J. Cook and N. C. Krishnan, *Activity learning: discovering, recognizing and predicting human behavior from sensor data*. John Wiley & Sons, 2015.

[3] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, 2019.

[4] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, vol. 105, pp. 233–261, August 2018.

[5] W. S. Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, "Human activity recognition using inertial sensors in a smartphone: An overview," *Sensors (Switzerland)*, vol. 19, no. 14, pp. 14–16, 2019.

[6] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[7] L. Wang, S. Mekki, H. Gjoreski, S. Valentin, M. Ciliberto, and D. Roggen, "Benchmarking the shl recognition challenge with classical and deep-learning pipelines," in *UbiComp/ISWC 2018 - Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2018 ACM International Symposium on Wearable Computers*, 2018.

[8] Q. Tang, D. J. Vidrine, E. Crowder, and S. S. Intille, "Automated detection of puffing and smoking with wrist accelerometers," in *Proc. - PERVASIVEHEALTH 2014 8th Int*. Conf. Pervasive Comput. Technol. Healthc, 2014, pp. 80–87.

[9] M. E. Mlinac and M. C. Feng, "Assessment of activities of daily living, self-care, and independence," *Arch. Clin. Neuropsychol.*, vol. 31, no. 6, pp. 506–516, 2016.

[10] R. C. al., "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, 2013.

[11] A. Reiss and D. Stricker, *Creating and benchmarking a new dataset for physical activity monitoring*. ACM Int. Conf. Proceeding Ser, 2012.

[12] P. Casale, O. Pujol, and P. Radeva, "Personalization and user verification in wearable systems using biometric walking patterns," *in Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 563–580, 2012.

[13] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, *Using mobile phones to determine transportation modes*. ACM Trans. Sens. Networks, 2010.

[14] L. Wang, H. Gjoreskia, K. Murao, T. Okita, and D. Roggen, "Summary of the sussex-huawei locomotion-transportation recognition challenge," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, ser. UbiComp '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1521 – 1530.

[15] B. Friedrich, C. LÃ¼bbe, and A. Hein, "Analyzing the importance of sensors for mode of transportation classification," *Sensors*, vol. 21, no. 1, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/1/176

[16] H. G. al., "The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices," *IEEE Access*, vol. 6, pp. 42 592–42 604, 2018.

[17] M. Shoaib, H. Scholten, P. J. M. Havinga, and O. D. Incel, "A hierarchical lazy smoking detection algorithm using smartwatch sensors," in *2016 IEEE 18th International Conference on e-Health Networking*. Healthcom 2016: Applications and Services, 2016.

[18] S. Agac, M. Shoaib, and O. Durmaz Incel, "Smoking recognition with smartwatch sensors in different postures and impact of user's height," *Journal of Ambient Intelligence and Smart Environments*, vol. 12, pp. 239–261, 2020.

[19] N. S. al., "Puffmarker: A multi-sensor approach for pinpointing the timing of first lapse in smoking cessation," in *UbiComp 2015 - Proc. 2015 ACM Int.* vol. 2015: Jt. Conf. Pervasive Ubiquitous Comput, 2015, pp. 999–1010.

[20] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE pervasive Comput.*, vol. 9, p. 1, 2010.

[21] G. Wang, Q. Li, L. Wang, W. Wang, M. Wu, and T. Liu, "Impact of sliding window length in indoor human motion modes and pose pattern recognition based on smartphone sensors," *Sensors (Switzerland)*, vol. 18, p. 6, 2018.

[22] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Activity recognition with evolving data streams: A review," *ACM Comput. Surv.*, vol. 51, p. 4, 2018.

[23] M. Janidarmian, A. R. Fekr, K. Radecka, and Z. Zilic, "A comprehensive analysis on wearable acceleration sensors in human activity recognition," *Sensors (Switzerland)*, vol. 17, p. 3, 2017.

[24] M. Zhang and A. A. Sawchuk, "A feature selection-based framework for human activity recognition using wearable multimodal sensors," *BODYNETS*, vol. 2011, pp. 92–98, 2012.

[25] 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 Comput. Surv.*, vol. 54, no. 4, May 2021.

[26] H. Kwon, C. Tong, H. Haresamudram, Y. Gao, G. D. Abowd, N. D. Lane, and T. Plötz, "Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 3, Sep. 2020.

[27] H. H. al., "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng*, vol. 21, no. 9, pp. 1263–1284, 2009.

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

[29] T. R. Hoens and N. V. Chawla, *Imbalanced datasets: From sampling to classifiers*. in Imbalanced Learning: Foundations, Algorithms, and Applications, 2013.

[30] H. F. Nweke, Y. W. Teh, G. Mujtaba, U. R. Alo, and M. A. Al-garadi, "Multi-sensor fusion based on multiple classifier systems for human activity identification," *Human-centric Comput. Inf*, vol. 9, p. 1, 2019.

[31] Y. Vaizman, N. Weibel, and G. Lanckriet, "Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification," in *Proc. ACM Interactive*, W. U. T. Mobile, Ed. no. 4: vol. 1, 2018, pp. 1–22.

[32] S. R. Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—a survey," in *Interdiscip. Rev. Data Min. Knowl. Discov., vol. 8, no. 4*, 2018, pp. 1–11.

[33] P. Lago, M. Matsuki, and S. Inoue, "Achieving single-sensor complex activity recognition from multi-sensor training data," *ArXiv*, vol. abs/2002.11284, 2020.

[34] B. Nguyen, Y. Coelho, T. Bastos, and S. Krishnan, "Trends in human activity recognition with focus on machine learning and power requirements," *Machine Learning with Applications*, vol. 5, p. 100072, 2021.

[35] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Comput. Surv.*, vol. 52, p. 4, 2019.

[36] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[37] R. Yao, G. Lin, Q. Shi, and D. C. Ranasinghe, "Efficient dense labelling of human activity sequences from wearables using fully convolutional networks," *Pattern Recognit.*, vol. 78, pp. 252–266, 2018.

[38] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*, 1st ed. Wiley-IEEE Press, 2013.

[39] W. Sousa Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, "Human activity recognition using inertial sensors in a smartphone: An overview," *Sensors*, vol. 19, no. 14, 2019.

[40] A. Akbari and R. Jafari, "Transferring activity recognition models for new wearable sensors with deep generative domain adaptation," in *IPSN 2019 - Proc. 2019 Inf.* Process. Sens. Networks, 2019, pp. 85–96.

[41] J. A. Ward, P. Lukowicz, G. Troster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Trans. Pattern Anal*, vol. 28, no. 10, pp. 1553–1567, 2006.

[42] H. F. Nweke, Y. W. Teh, G. Mujtaba, and M. A. Al-garadi, "Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions," *Inf. Fusion*, vol. 46, pp. 147–170, June 2019.

[43] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[44] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 214–223.

[45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[46] S. Zhao, J. Song, and S. Ermon, "Towards deeper understanding of variational autoencoding models," 2017.

[47] Z. Wang, Q. She, and T. Ward, "Generative adversarial networks: A survey and taxonomy," *ArXiv*, vol. abs/1906.01529, 2019.

[48] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process*, vol. 35, no. 1, pp. 53–65, 2018.

[49] V. Sampath, I. Maurtua, J. J. Aguilar Martín, and A. Gutierrez, "A survey on generative adversarial networks for imbalance problems in computer vision tasks," *Journal of Big Data*, vol. 8, no. 1, p. 27, 2021. [Online]. Available: https://doi.org/10.1186/s40537-021-00414-0

[50] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 5769 – 5779.

[51] S. Ger and D. Klabjan, "Autoencoders and generative adversarial networks for imbalanced sequence classification," *ArXiv*, 2020.

[52] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res*, vol. 16, pp. 321–357, 2002.

[53] Y. Dong and X. Wang, "A new over-sampling approach: Random-smote for learning from imbalanced data sets," *Lect. Notes Comput. Sci*, vol. 7091, pp. 343–352, 2011.

[54] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.

[55] S. Hu, Y. Liang, L. Ma, and Y. He, *MSMOTE: Improving classification performance when training data is imbalanced*. WCSE 2009: in 2nd International Workshop on Computer Science and Engineering, 2009.

[56] S. Barua, M. M. Islam, and K. Murase, "A novel synthetic minority oversampling technique for imbalanced data set learning," *Lect. Notes Comput. Sci*, vol. 7063, pp. 735–744, 2011.

[57] S. Barua, M. Islam, and K. Murase, "Prowsyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning," *Advances in Knowledge Discovery and Data Mining*, pp. 317–328, 2013.

[58] Y. Chen and C. Shen, "Performance analysis of smartphone-sensor behavior for human activity recognition," *IEEE Access*, vol. 5, pp. 3095–3110, 2017.

[59] J. W. Lockhart, T. Pulickal, and G. M. Weiss, "Applications of mobile activity recognition," in *UbiComp'12 - Proc. 2012 ACM Conf. Ubiquitous Comput.* no. September 2012, 2012, pp. 1054–1058.

[60] G. Santos, P. Endo, K. Monteiro, E. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," *Sensors*, vol. 19, no. 7, p. 1644, 2019.

[61] R. Jafari, W. Li, R. Bajcsy, S. Glaser, and S. Sastry, *Physical activity monitoring for assisted living at home*. in IFMBE Proceedings, 2007.

[62] S. L. Lau, I. K"onig, K. David, B. Parandian, C. Carius-D"ussel, and M. Schultz, "Supporting patient monitoring using activity recognition with a smartphone," in *Proceedings of the 2010 7th International Symposium on Wireless Communication Systems*. ISWCS'10, 2010.

[63] M. B. al., *Wearable assistant for Parkinsons disease patients with the freezing of gait symptom*. IEEE Trans. Inf. Technol. Biomed, 2010.

[64] J.-L. Reyes-Ortiz, L. Oneto, A. SamÃ , X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[65] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Complex human activity recognition using smartphone and wrist-worn motion sensors," *Sensors (Switzerland)*, vol. 16, no. 4, pp. 1–24, 2016.

[66] D. R. al., *Walk-through the OPPORTUNITY dataset for activity recognition in sensor rich environments*. Int. Conf. Pervasive Comput, 2010.

[67] M. Ciliberto, F. J. O. Morales, H. Gjoreski, D. Roggen, S. Mekki, and S. Valentin, "Poster abstract: High reliability android application for multidevice multimodal mobile data acquisition and annotation," in *SenSys 2017 - Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems*, 2017.

[68] Y. E. Ustev, C. Ersoy, and O. D. Incel, "User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal," in *UbiComp 2013 Adjunct - Adjunct Publication of the 2013 ACM Conference on Ubiquitous Computing*, 2013.

[69] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014.

[70] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, 2011.

[71] O. Banos, J. M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors (Switzerland)*, vol. 14, no. 4, pp. 6474–6499, 2014.

[72] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors (Switzerland)*, vol. 15, no. 12, pp. 31 314–31 338, 2015.

[73] J. Zhu, R. San-Segundo, and J. M. Pardo, "Feature extraction for robust physical activity recognition," *Human-centric Comput. Inf*, vol. 7, no. 1, pp. 1–16, 2017.

[74] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors - a review of classification techniques," *Physiol. Meas*, vol. 30, p. 4, 2009.

[75] S. A. Muhammad, B. N. Klein, K. Van Laerhoven, and K. David, "A feature set evaluation for activity recognition with body-worn inertial sensors," *Communications in Computer and Information Science*, pp. 101 – 109, 2012.

[76] N. Y. Hammerla, S. Halloran, and T. Pl"otz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *IJCAI Int*, vol. 2016, pp. 1533–1540, Jan 2016.

[77] F. Chollet, *Deep learning with Python*. Manning, 2018.

[78] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[79] A. D. Ignatov and V. V. Strijov, "Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer," in *Multimed*. vol. 75, no. 12: Tools Appl, 2016, pp. 7257–7270.

[80] Z. Feng, L. Mo, and M. Li, "A random forest-based ensemble method for activity recognition," in *Proc. Annu*. vol. 2015-Novem: Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS, 2015, pp. 5074–5077.

[81] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," *Lect. Notes Comput. Sci*, vol. 7657, pp. 216–223, 2012.

[82] H. Gjoreski, J. Bizjak, M. Gjoreski, and M. Gams, "Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer," in *Proceedings of the IJCAI 2016 Workshop on Deep Learning for Artificial Intelligence*. NY, USA vol. 10, no. Ml, p. 970: New York, 2016.

[83] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016.

[84] M. Z. al., "Convolutional neural networks for human activity recognition using mobile sensors," in *Proceedings of the 2014 6th International Conference on Mobile Computing*. MobiCASE 2014 no. January: Applications and Services, 2015, pp. 197–205.

[85] H. C. S. al., "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," in *IEEE Trans*. vol. 35, no. 5: Med. Imaging, 2016, pp. 1285–1298.

[86] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1d 2d cnn lstm networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, 2019.

[87] A. Murad and J. Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors (Switzerland)*, vol. 17, p. 11, 2017.

[88] G. AurelÌlien, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly, 2019.

[89] J. V. Jeyakumar, S. S. Sandha, E. S. Lee, N. Tausik, Z. Xia, and M. Srivastava, "Deep convolutional bidirectional lstm based transportation mode recognition," in *UbiComp/ISWC 2018 - Adjun*. Proc. 2018 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. Proc. 2018 ACM Int. Symp. Wearable Comput, 2018, pp. 1606–1615.

[90] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[91] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process*, vol. 45, no. 4, pp. 427–437, 2009.

[92] Ç.Berke Erdaş, I. Atasoy, K. Açıcı, and H. Oğul, "Integrating features for accelerometer-based activity recognition," *Procedia Computer Science*, vol. 98, pp. 522–527, 2016.

[93] Ç. B. Erdaş and S. Güney, "Human activity recognition by using different deep learning approaches for wearable sensors," *Neural Processing Letters*, vol. 53, no. 3, pp. 1795–1809, Jun 2021. [Online]. Available: https://doi.org/10.1007/s11063-021-10448-3

[94] A. D. Antar, M. Ahmed, M. S. Ishrak, and M. A. R. Ahad, "A comparative approach to classification of locomotion and transportation modes using smartphone sensor data," in *UbiComp/ISWC 2018 - Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2018 ACM International Symposium on Wearable Computers*, 2018, pp. 1497–1502.

[95] H. He and Y. Ma, *Imbalanced learning foundations, algorithms, and applications*. IEEE Press, Wiley, 2013.

[96] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special issue on learning from imbalanced data sets," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 1–6, Jun. 2004.

[97] M. M. Rahman and D. N. Davis, "Addressing the class imbalance problem in medical datasets," *Int. J. Mach*, vol. 3, no. 2, pp. 224–228, 2013.

[98] J. A. Saez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Information Sciences*, vol. 291, pp. 184–203, 2015.

[99] N. Japkowicz, "Concept-learning in the presence of between-class and within-class imbalances," *in Advances in Artificial Intelligence pp*, pp. 67–77, 2001.

[100] Y. Geng and X. Luo, "Cost-sensitive convolution based neural networks for imbalanced time-series classification," *arXiv Prepr. arXiv*, vol. 1801, p. 04396, 2018.

[101] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020.

[102] R. Blagus and L. Lusa, "Smote for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 14, 2013.

[103] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and smote," *Inf. Sci*, vol. 465, pp. 1–20, 2018.

[104] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial Intelligence Review*, vol. 22, pp. 177–210, 2004.

[105] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[106] A. A. Shanab, T. M. Khoshgoftaar, and R. Wald, "Impact of noise and data sampling on stability of feature selection," in *2011 10th International Conference on Machine Learning and Applications and Workshops*, vol. 1, 2011, pp. 172–177.

[107] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *In Proceedings of the Fourteenth International Conference on Machine Learning*, 1997, pp. 179–186.

[108] G. Kovács, "smote-variants: a python implementation of 85 minority oversampling techniques," *Neurocomputing*, vol. 366, pp. 352–354, 2019, (IF-2019=4.07).

[109] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Autom. Sin*, vol. 4, no. 4, pp. 588–598, 2017.

[110] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial networks and their variants work: An overview," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–46, 2019.

[111] Q. N. al., "Leveraging wearable sensors for human daily activity recognition with stacked denoising autoencoders," *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–22, 2020.

[112] G. V. al., "Overly optimistic prediction results on imbalanced data: a case study of flaws and benefits when applying over-sampling," in *Artif.* vol. 111, no. September 2020: Intell. Med, 2021.

[113] "Stable variable selection of class-imbalanced data with precision-recall criterion," *Chemometrics and Intelligent Laboratory Systems*, vol. 171, pp. 241–250, 2017.

[114] M. Alzantot, S. Chakraborty, and M. Srivastava, "Sensegen: A deep learning architecture for synthetic sensor data generation," vol. 2017, pp. 188–193, 2017.

[115] A. Saeed, T. Ozcelebi, and J. Lukkien, "Synthesizing and reconstructing missing sensory modalities in behavioral context recognition," *Sensors*, vol. 18, no. 9, 2018.

[116] A. Fernández, S. García, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res*, vol. 61, pp. 863–905, 2018.

[117] D. R. al., "Collecting complex activity datasets in highly rich networked sensor environments," in *INSSth International Conference on Networked Sensing Systems pp.* 233–240, 2010, pp. 2010–7.

[118] M. Shoaib, S. Bosch, H. Scholten, P. J. M. Havinga, and O. D. Incel, "Towards detection of bad habits by fusing smartphone and smartwatch sensors," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops*, P. W. 2015, Ed., 2015, pp. 591–596.

[119] F. Pedregosa and G. Varoquaux, "Scikit-learn: Machine learning in python," *vol.*, vol. 12, 2011.

[120] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: https://www.R-project.org/

[121] Y. Guan and T. Plotz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1 – 28, Jun 2017.

[122] R. Ghorbani and R. Ghousi, "Comparing different resampling methods in predicting students' performance using machine learning techniques," *IEEE Access*, vol. 8, pp. 67 899–67 911, 2020.

[123] A. Shahi, J. D. Deng, and B. J. Woodford, "A streaming ensemble classifier with multi-class imbalance learning for activity recognition," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3983–3990.

[124] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1 – 30, Dec. 2006.

[125] B. Yazici and S. Yolacan, "A comparison of various tests of normality," *Journal of Statistical Computation and Simulation*, vol. 77, no. 2, pp. 175–183, 2007.

[126] L. Jantschi and S. D. Bolboaca, "Computation of probability associated with anderson-darling statistic," *Mathematics*, vol. 6, no. 6, 2018.

[127] A. Agrawal, H. L. Viktor, and E. Paquet, "Scut: Multi-class imbalanced data classification using smote and cluster-based undersampling," in *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, vol. 01, 2015, pp. 226–234.

[128] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[129] M. Mukherjee and M. Khushi, "Smote-enc: A novel smote-based method to generate synthetic data for nominal and continuous features," *Applied System Innovation*, vol. 4, no. 1, 2021.

[130] A. Kaur and I. Kaur, "An empirical evaluation of classification algorithms for fault prediction in open source projects," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 1, pp. 2–17, 2018.

[131] N. Verbiest, E. Ramentol, C. Cornelis, and F. Herrera, "Improving smote with fuzzy rough prototype selection to detect noise in imbalanced classification data," in *Advances in Artificial Intelligence – IBERAMIA 2012*, J. Pavón, N. D. Duque-Méndez, and R. Fuentes-Fernández, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 169–178.

[132] G. W. Corder and D. I. Foreman, *Nonparametric Statistics for Non-Statisticians*, sep 2011.

[133] R. A. Fisher, *Statistical methods and scientific inference*. Oxford, England: Hafner Publishing Co., 1956.

[134] D. J. Dittman, T. M. Khoshgoftaar, and A. Napolitano, "Selecting the appropriate data sampling approach for imbalanced and high-dimensional bioinformatics datasets," in *2014 IEEE International Conference on Bioinformatics and Bioengineering*, 2014, pp. 304–310.

[135] A. Ghazikhani, H. S. Yazdi, and R. Monsefi, "Class imbalance handling using wrapper-based random oversampling," in *20th Iranian Conference on Electrical Engineering (ICEE2012)*, 2012, pp. 611–616.

[136] B. Remeseiro, M. Penas, A. Mosquera, J. Novo, M. G. Penedo, and E. Yebra-Pimentel, "Statistical Comparison of Classifiers Applied to the Interferential Tear Film Lipid Layer Automatic Classification," *Computational and Mathematical Methods in Medicine*, vol. 2012, p. 207315, 2012. [Online]. Available: https://doi.org/10.1155/2012/207315

[137] A. D. Antar, M. Ahmed, M. S. Ishrak, and M. A. R. Ahad, "A comparative approach to classification of locomotion and transportation modes using smartphone sensor data," in *UbiComp/ISWC 2018 - Adjun*. Proc. 2018 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. Proc. 2018 ACM Int. Symp. Wearable Comput, 2018, pp. 1497–1502.

[138] M. Arif, M. Bilal, A. Kattan, and S. I. Ahamed, "Better physical activity classification using smartphone acceleration sensor," *Journal of Medical Systems*, vol. 38, no. 9, 2014.

[139] F. Li, K. Shirahama, M. A. Nisar, L. K"oping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors (Switzerland)*, vol. 18, p. 2, 2018.

[140] F. C. al., "Comparing cnn and human crafted features for human activity recognition," in *Proc. - 2019 IEEE SmartWorld*, U. Intell, Ed. Comput. Adv. Trust. Comput. Scalable Comput. Commun. Internet People Smart City Innov. SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019, 2019, pp. 960–967.

[141] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. J. Mach. Learn. Res, 2014.

[142] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my gan?" *in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1120, pp. 218–234, 2018.

[143] S. W. Heads and L. Leuzinger, "On the placement of the cretaceous orthopteran brauckmannia groeningae from brazil, withnotes on the relationships of schizodactylidae (orthoptera, ensifera)," *in ZooKeys*, vol. 77, pp. 17–30, 2011.

[144] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, "Lstm networks for mobile human activity recognition," *Int. Conf. Artif. Intell. Technol. Appl., no*, pp. 50–53, 2016.

[145] F. M. Rueda, R. Grzeszick, G. Fink, and S. F. and, "and m. ten hompel, "convolutional neural networks for human activity recognition using body-worn sensors," informatics, vol. 5, no. 2, p. 26," vol. 2018.

[146] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 351–360.

[147] F. Chollet, "Keras," *GitHub Repository*, 2015. [Online]. Available: https://github.com/fchollet/keras

[148] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2014.

[149] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[150] J. Jo, J. Kung, and Y. Lee, "Approximate lstm computing for energy-efficient speech recognition," *Electronics*, vol. 9, no. 12, 2020.

[151] P. Le and W. Zuidema, "Compositional distributional semantics with long short term memory," 2015.

[152] "Is the k-nn classifier in high dimensions affected by the curse of dimensionality?" *Computers Mathematics with Applications*, vol. 65, no. 10, pp. 1427–1437, 2013, grasping Complexity.

[153] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *SpringerPlus*, vol. 5, no. 1, p. 1304, 2016.

[154] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, ser. IJCAI'11. AAAI Press, 2011, pp. 1729 – 1734.

[155] Y. Mohammad, K. Matsumoto, and K. Hoashi, "Deep feature learning and selection for activity recognition," in *Proceedings of the ACM Symposium on Applied Computing*, 2018, pp. 930–939.

Appendix A

A Convolutional Neural Network for Smoking Activity Recognition

Fayez Alharbi
Department of Computing
Goldsmiths, University of London
London, United Kingdom
E-mail: falha011@gold.ac.uk

Katayoun Farrahi
Electronics and Computer Science
University of Southampton
Southampton, United Kingdom
E-mail: k.farrahi@soton.ac.uk

***Abstract*—Smoking is linked to more than two million preventable deaths yearly. The widespread use of sensors embedded in everyday devices provides novel means for research on smoking. Smartphones and smartwatches can monitor smoking behavior, which could lead to the development of new methods for smoking reduction or cessation. However, smoking often co-occurs with other activities, such as drinking and eating, which makes the recognition of concurrent and overlapping smoking activities from wearable sensors challenging. In this paper, we proposed for the first time to use deep learning for the automatic detection of smoking activities. A Convolutional Neural Network (CNN) architecture was proposed, and this improved on previously reported performance results. We investigated the impact of various data preprocessing approaches that influence the CNN classification results with statistical features and raw sensor data. We also considered the individual performance of the smartwatch vs. the smartphone and the gyroscope vs. accelerometer sensors for smoking activity recognition. Considering a dataset of concurrent activities such as drinking, eating, smoking while sitting, standing, walking, and partaking in a group conversation, our CNN approach obtained an F1-score of 92-96% in person-independent evaluation.**

Keywords: Smoking activity recognition; health monitoring; human activity recognition; ubiquitous computing; deep learning; Convolutional Neural Networks

I. INTRODUCTION

As reported by the World Health Organization (WHO), smoking has been classified as one of the leading causes of premature death [1]. Smoking also causes numerous types of neoplastic diseases [2]. Moreover, the popularity of tobacco products such as cigarettes has increased dramatically in the last decade, which has led to the rising use of tobacco products among the youth [3]. These facts highlight that research on smoking may play a significant role in decreasing diseases and deaths caused by smoking. Wearable devices provide advanced health behavior tools for smoking recognition, which may be constructive in smoking addiction treatment and research.

Smoking detection has been investigated using varying technologies such as computer vision and text messages [4], [5]. Smartphones and more recently, the widespread use of smartwatches and other wearable sensing devices in certain populations [6] are providing large amounts of ubiquitously collected data as well as tools and apps for researchers in the human activity recognition (HAR) domain, particularly in relation to smoking activities [7], [8].

Prior studies have investigated approaches based on machine learning (ML) to automatically classify activities related to smoking [9]. All of the previous works used handcrafted features extracted from sensor data and considered single- or two-layer popular classifiers such as Random Forest (RF) [10] and Support Vector Machine (SVM) [11]. In this paper, we consider the successful use of deep learning, paying particular attention to convolutional neural networks (CNNs), for the sensor data. We explore various architectures based on CNNs and find an architecture with raw input data obtained directly from the accelerometer and gyroscope sensors to outperform all previous work. The contributions of this paper are as follows:

- To the best of our knowledge, this is the first study to apply deep learning, particularly CNNs, for smoking activity recognition. We outperform previous work with our CNN approach applied to the raw sensor data on several smoking activity classifications, specifically considering concurrent activities such as smoking while sitting, standing or walking, and while partaking in a group conversation.
- We consider the individual impact of the smartwatch vs. the smartphone for smoking classification and find the smartwatch to be the most predictive. However, the smartphone does almost always increase the predictive power of the CNN model.
- We consider the gyroscope vs. accelerometer sensors individually for smoking activity classification and find they both perform effectively.

This paper is organized as follows. Section II presents the related work. Then in section III, we describe our method, including the dataset, the dataset preprocessing, model evaluation, the raw input and features input, an explanation of the CNN architecture and network design. Section IV then discusses our results, incorporating, the number and size of the convolutional filters, the batch size, inputs, and smartwatch vs. smartphone and accelerometer vs. gyroscope. Finally, section V presents the conclusion and recommendations for future work.

II. RELATED WORK

One of the earliest works on smoking activity recognition was a feasibility study using wrist-worn accelerometer data

978-1-5386-4294-8/18/$31.00 ©2018 IEEE

[12]. In this study, Scholl et al. collected data from four participants who wore an accelerometer sensor device on their wrist for one week. They used a Gaussian mixture classifier for smoking activity detection. The authors reported the performance of their model with a recall of 70% and a precision of 51% for smoking recognition. Despite the low smoking detection performance, they presented fascinating insights into the recognition of smoking activity by adopting wrist-worn sensor data. Nevertheless, this study was only conducted in terms of smoking while standing and lacked other confounding activities such as smoking while sitting or drinking.

Tang et al. [13] built a two-layer smoking detection model, which used two accelerometers to sense data from both wrists. The dataset was comprised of six participants who performed several smoking-related activities such as smoking while talking in a group, smoking while sitting, standing, walking or eating, and smoking while using a phone. They also extracted some time-domain features from the accelerometer data. They reported an F1 score of 79% for smoking detection using RF and SVM classifiers. The F1 score incorporated both recall and precision, which specified the total number of correct identified samples. The F1 score was low due to the similarity between smoking and the other activities such as drinking. However, our approach aims to use accelerometer and gyroscope data, which will add more sensor information to the classifier. We also only consider one wrist, which will address a more realistic problem formulation for smoking detection, given that smartwatches are now so widespread.

Qin et al. [14] performed a study where the data was comprised of multiple types of sensors, such as an accelerometer, Wi-Fi, and GPS; the data was collected from three participants over a one-month period. The data was labeled in terms of segments of intervals of smoking and non-smoking. They used a multivariate hidden Markov model to classify periods in order to identify whether a participant was smoking or not smoking. In contrast, we consider more varied and fine-grained activities related to smoking rather than a binary classification problem.

Shoaib et al. [15] presented some results on both two-layer and single-layer classifiers, including Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT), for smoking detection. They extracted time-domain features to classify the sensor data. The F1 score fell between 83-94% when using two-layer classifiers to detect smoking activities. However, they mentioned certain challenges in recognizing concurrent activities such as smoking and drinking activities. In this paper, we use the data collected by Shoaib et al. [15]. However, we report improved performance results and present further insights into the smoking activity recognition problem. We use an approach based on deep learning, specifically CNNs, which allows for the use of raw input data and removes the need for two-layer classifiers and feature engineering.

III. METHOD

A. Dataset

The dataset which used in this study was collected from eleven participants and was originally presented by Shoaib et al.[15]. Each participant wore a smartwatch and a smartphone, which represents a realistic way to capture data owing to the ubiquitous nature of these devices and their widespread use. Sensors on both wrists or legs would not readily scale and are therefore not of interest. Most smartwatches and smartphones are embedded with accelerometers and gyroscopes, which are the sensors used in this study. The participants performed ten activities: smoking while standing (SmkSTD); smoking while sitting (SmkSIT); smoking while partaking in a group conversation (SmkG); smoking while walking (SmkW); drinking while standing (DrinkSTD); drinking while sitting (DrinkSIT); standing (STD); sitting (SIT); walking (WALK) and eating (Eat). Each activity was performed various times by each participant on multiple days for three months. Details of the data can be seen in [15]. In this paper, we use the acronym of each activity when we describe or mention them. We excluded some physical activities from the data, such as walking, sitting, and standing, because they were very simple to classify using the CNN.

This study objective is to optimize a deep learning approach for smoking activity recognition. It also aims to determine how well it performs compared to previous "state-of-the-art" approaches. We particularly focus on concurrent activities such as smoking and drinking. We also investigate how much of the predictive performance comes from the watch vs. the phone as well as the accelerometer vs. the gyroscope.

B. Evaluation

We evaluate the models using the F1 score. The CNN is also coded in Python using the Keras framework with a Tensorflow backend [16].

In this paper, we consider person-independent evaluation. We split the whole dataset into training, validation and testing sets using the stratified split data method from Scikit-learn [17]. The allocation of the split data is 70% for training, 15% for validation and 15% for testing. This method balances the number of instances of each class in each split. Since none of the participants had performed all the activities, we divided the participants into three groups for evaluation based on the activities that they had performed, as is shown in TABLE I. The main difference is that the SmkG activity is present in group 2 and SmkW is in group 3. Next, we will present the two approaches considered for modeling the input data.

TABLE I. SMOKING ACTIVITY GROUPS FOR EXPERIMENTS

Groups	Participants	Performed Activity
group 1	1-11	SmkSTD, SmkSIT, DrinkSTD, DrinSIT, Eat
group 2	1-8	SmkG, SmkSTD, SmkSIT, DrinkSTD, DrinkSIT, Eat
group 3	1-3	SmkW, SmkG, SmkSTD, SmkSIT, DrinkSTD, DrinkSIT, Eat

C. Raw Input

First, we implemented a non-overlapping sliding window to segment the data. We tried multiple window sizes of 10, 20 and 30 and chose the window size of 30 because it increases the model performance. We called this method "raw input". Each segment is formulated as a tensor and input to the CNN model. The first dimension of the tensor consists of the number of observations in a window, which is 30 seconds in dimension. The second dimension is 4, which corresponds to

the four streams of sensor data. We had two accelerometer sensors (smartwatch and smartphone) in addition to two gyroscope sensors, each consisting of three channels, X, Y, and Z. Finally, our input tensor had the shape (30,4,3).

D. Feature Input

In this method, we considered an approach based on feature extraction from the accelerometer and gyroscope data. We extracted four time-domain features and called this method "features input". We applied a non-overlapping sliding window to segment the data as in previous work [18]. To do this, we selected a window of 30 seconds, because the network showed a better performance in terms of smoke detection. Then, we extracted four time-domain statistical features from each window segment: maximum, minimum, skewness and kurtosis. In this approach, the input tensor shape is (30,16,3). Here, 16 is the total number of extracted features of each of the three channels, X, Y, and Z, of the accelerometer and gyroscope sensors.

E. Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a variant of the neural network [19]. A CNN is composed of three types of layers: a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer is the primary element of a CNN. This layer attempts to discover patterns in the data by using sliding filters or kernels across the data. A dot product is computed at each step, and the captured values produce the outputs of the convolutional layer, which are called the feature map. The pooling layer often follows a convolutional layer and down-samples the previous layers' feature maps. To create its features map, the pooling layers take an average or a maximum number of small rectangular blocks of the data. The last layer in the CNN is a fully connected layer. This layer is used at the end of the network after several convolutional and pooling layers have performed feature extraction. The output of this layer is flattened into a one-dimensional vector and used for classification. This layer usually uses a nonlinear activation function or a SoftMax activation function, which we use to output probabilities for class predictions.

F. Network Design

We now present the CNN model and a discussion of the parameters, which were optimized for this architecture. The number of convolutional layers was crucial to improving the F1 score. A CNN of more than two layers was not efficient for the raw input task, as the network began to overfit the data.

The network design that we used for the raw input (Fig.1) began with two layers of convolution with 120 and 128 feature maps. We then added a batch normalization layer after each convolution layer. Next, a global average pooling layer was incorporated [20]. After that, the dropout layer with a rate of 65% was employed. We also added a dense layer of 128 neurons with ReLU activation functions, which improved the network performance before the output layer. ReLU activation function was selected as it enhanced the model performance. The number of neurons on the output layer was based on the group classes, which was 5 for group 1, 6 for group 2 and 7 for group 3, with a SoftMax activation function to compute the probability distribution for each class.

For the features input, the optimal number of the convolutional layer was three layers. The network design started with three convolutional layers (as presented in Fig.1) with 120, 128 and 256 feature maps, and the activation function for each layer was a ReLU. When ReLU used, the model accomplished enhanced results. We then defined a global average pooling layer that increased the classifier performance. Next, the layer dropout was included to reduce overfitting. It was also set to randomly exclude 65% of the neurons. Next, a dense layer with 128 neurons was added with the ReLU activation function and then a batch normalization layer, which enhanced the network performance. Lastly, the output layer had a number of neurons according to the group classes, 5, 6, or 7, with a SoftMax activation function.

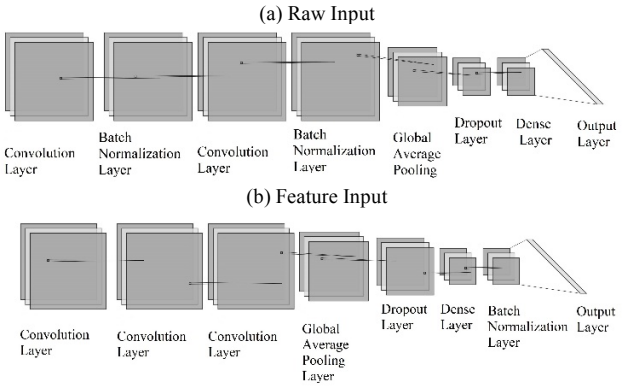

Figure 1. NETWORK ARCHITECTURE

The network was trained to minimize cross-entropy utilizing Adam gradient descent optimization with a logarithmic loss function (categorical cross-entropy). The optimal learning rate in terms of features input was 0.00001, while for raw input it was 0.0001.

IV. RESULT

We investigated how CNN parameters influence the classification results such as network design, number and size of convolutional filters, and batch size. We also analyzed the impact of classification results by using the smartwatch vs. smartphone data, as well as explored using accelerometer vs. gyroscope data for various smoking-related activities.

A. Number and Size of The Convolutional Filters

The F1 score was remarkably increased when the number of convolutional filters in each convolutional layer was raised to 120 and 128 for raw input and 120, 128 and 256 for features input respectively. Utilizing a smaller number (e.g. 32 or 64) of convolutional filters provided insufficient F1 scores. The size of the convolutional filter parameter was sensitive to the network. The filter size was 2 for all convolutional layers in terms of both raw input and features input. The F1 score dropped if this filter size was increased to more than 2.

B. Batch Size

With regard to batch size, the network was apparently sensitive to this parameter both in the case of raw input and features input. We noticed that training batch sizes of 8 and 16 yielded better performance in comparison to 32, 64 and 128. The best F1 score was obtained with a batch size of 8 for raw input for all groups. Based on our experiments, small batch sizes such as 8 showed that the network was not prone to becoming confused between similar activities, such as smoking and drinking, mainly in the case of raw input.

C. Inputs

This section compares the results of raw input vs. features input. The CNN model shows notable results in terms of discriminating between smoking and drinking activities, which are regularly confused due to the similar hand-to-mouth gestures. We considered a batch size of 8. As shown in TABLE II, the CNN model with raw input achieved the best F1 score for all groups. The table presents the results for person-independent classification. In group 1, in terms of smoking activity the CNN achieved an average F1 score of 94% using the raw input. In group 2, the F1 score was 92%, while it was 96% for group 3.

TABLE II. THE CNN AVERAGE F1 SCORE FOR PERSON-INDEPENDENT CLASSIFICATION

Activity	Raw Input	Features Input
smoke	0.94	0.83
drink	0.95	0.82
eat	0.97	0.93

(a) group 1

Activity	Raw Input	Features Input
smoke	0.92	0.80
drink	0.93	0.85
eat	0.97	0.82

(b) group 2

Activity	Raw Input	Features Input
smoke	0.96	0.88
drink	0.91	0.71
eat	0.92	0.74

(c) group 3

In TABLE III, we further compare the raw input vs. features input cases for the concurrent activities. In the case of the features input, we can observe that similar activities were often confused, particularly the drinking and smoking activities. In group 2, there was one case where the model failed to recognize the difference between drinking while sitting and smoking while partaking in a group conversation. The F1 score was 86% for both activities. Another case was in group 3 where the batch was 8; here, the F1 score was 95 % for both smoking in group conversation and smoking while walking. Note that the raw input always outperformed the features input case [21].

TABLE III. THE AVERAGE F1 SCORE FOR PERSON-INDEPENDENT CLASSIFICATION OF SMOKING-RELATED ACTIVITIES. WE CAN COMPARE THE RAW INPUT VS. FEATURES INPUT CASES AND FIND THE CNN WITH THE RAW INPUT ALWAYS OUTPERFORMS THE FEATURES INPUT.

Activity	Raw Input	Features Input
DrinkSTD	0.97	0.86
DrinkSIT	0.93	0.79
Eat	0.97	0.93
SmkSTD	0.96	0.88
SmkSIT	0.92	0.78

(a) group 1

Activity	Raw Input	Features Input
DrinkSTD	0.93	0.84
DrinkSIT	0.95	0.86
Eat	0.97	0.82
SmkG	0.92	0.86
SmkSTD	0.91	0.74
SmkSIT	0.94	0.81

(b) group 2

Activity	Raw Input	Features Input
DrinkSTD	0.90	0.74
DrinkSIT	0.92	0.69
Eat	0.92	0.74
SmkW	0.100	0.95
SmkG	0.97	0.95
SmkSTD	0.98	0.87
SmkSIT	0.89	0.77

(c) group 3

D. Smartwatch vs. Smartphone

In this experiment, we explore how independently informative each of the sensors are for the smoking activity recognition task. Intuitively speaking, we would presume that the smartwatch will outperform the smartphone, as smoking involves a hand gesture and the watch directly senses the hand's activity. However, we would like to know how much additional information the smartphone adds to the classification problem, and how effective the stand-alone smartphone data is for the predictive task.

We ran the CNN for the classification task considering the smartwatch data and smartphone data independently. As is shown in TABLE IV, for group 1, we obtained an average F1 score of 86% for smoking activity using raw input and 80% when utilizing features input. For group 2, the average F1 score in terms of raw input was 92%, and 72% in terms of features input. For group 3, the F1 score was a high 95% using the smartwatch's raw input and 87% in terms of features input. However, as was expected, in terms of smoking activity, the model performance dropped when we utilized the smartphone data. For group 1, the average F1 score for raw input was 64% and 62% for the features input. In group 2, the performance was much lower, with 51% for raw input and 64% for the features input. Meanwhile, the F1 score for group 3 dropped to 81% using raw input and 73% utilizing features input. We can observe that the raw input seems to outperform the features input, even when considering the sensors individually. For all

the activities, when comparing Table IV to Table II, we can see that the smartphone data always added to the classification performance, albeit sometimes very minimally. We can conclude that the smoking activity recognition task would not be feasible using smartphone data alone, as while performance would be better than random, it would be below 83%. Furthermore, using smartwatch sensor data alone should be sufficient for predicting smoking activities with very high accuracy.

E. Accelerometer vs. Gyroscope

In this task, we investigate how useful each of the sensors is for the smoking activity identification. We studied the performance of the CNN model utilizing accelerometer data and gyroscope data independently as input data alongside our methods. We combined the accelerometer data from the smartwatch and smartphone and did the same with the gyroscope data. We then used each of them individually as input to the model. The results showed that using the raw input of accelerometer data and gyroscope data with CNN always provided better F1 scores, as is shown in TABLE V. In group 1, the CNN's F1 score was 93% for smoking activity using the accelerometer data. Using the gyroscope data for group 2, the highest F1 score was 92% for smoke activity, as it was for group 3 (94%). We can observe that using the raw input with CNN generally yielded a promising performance. This is due to the properties of CNN that means it discovers patterns within the data, regardless of whether the data was from the accelerometer or the gyroscope.

TABLE IV. SMARTWATCH VS. SMARTPHONE PERFORMANCE RESULTS CONSIDERING THE F1 SCORE FOR PERSON-INDEPENDENT CLASSIFICATION

Activity	Smartwatch Data		Smartphone Data	
	Raw Input	Features Input	Raw Input	Features Input
smoke	0.86	0.80	0.64	0.62
drink	0.73	0.72	0.67	0.60
eat	0.96	0.92	0.65	0.65

(a) group 1

Activity	Smartwatch Data		Smartphone Data	
	Raw Input	Features Input	Raw Input	Features Input
smoke	0.92	0.72	0.51	0.64
drink	0.90	0.76	0.66	0.65
eat	0.95	0.89	0.72	0.65

(b) group 2

Activity	Smartwatch Data		Smartphone Data	
	Raw Input	Features Input	Raw Input	Features Input
smoke	0.95	0.87	0.81	0.73
drink	0.90	0.69	0.82	0.59
eat	0.92	0.85	0.73	0.40

(c) group 3

TABLE V. THE ACCELEROMETER VS. GYROSCOPE SENSOR DATA PERFORMANCE RESULTS CONSIDERING THE F1 SCORE FOR PERSON-INDEPENDENT CLASSIFICATION. THE RAW INPUT OUTPERFORMS THE FEATURES INPUT CASE AND BOTH SENSORS ARE HIGHLY PREDICTIVE OF MOST SMOKING RELATED ACTIVITIES.

Activity	Accelerometer Data		Gyroscope Data	
	Raw Input	Features Input	Raw Input	Features Input
smoke	0.93	0.83	0.88	0.75
drink	0.91	0.83	0.79	0.67
eat	0.96	0.85	0.97	0.86

(a) group 1

Activity	Accelerometer Data		Gyroscope Data	
	Raw Input	Features Input	Raw Input	Features Input
smoke	0.89	0.75	0.92	0.66
drink	0.91	0.80	0.89	0.64
eat	0.96	0.74	0.93	0.86

(b) group 2

Activity	Accelerometer Data		Gyroscope Data	
	Raw Input	Features Input	Raw Input	Features Input
smoke	0.90	0.83	0.94	0.81
drink	0.93	0.75	0.83	0.68
eat	0.92	0.75	0.95	0.77

(c) group 3

V. CONCLUSION

To the best of our knowledge, this study is the first work to use CNN in smoking detection using wearable sensors. The CNN smoking detection model improves the recognition of smoking and the concurrent activities. The model significantly surpasses the performance of prior work and maintains competitive FI score results of 92-96% for smoking detection. Our results show that a CNN architecture with raw input achieves high classification performance and can classify complex activities such as drinking while smoking and drinking.

A potential future direction for this work could involve investigating the use of CNNs for other complex human activity recognition problems and considering the task of transfer learning for smoking activity recognition.

REFERENCES

[1] World Health Organization, "WHO report on the global tobacco epidemic: Raising taxes on tobacco," *World Heal. Organ.*, pp. 52–53, 2015.

[2] U.S. Department of Health and Human Services, *How Tobacco Smoke Causes Disease: The Biology and Behavioral Basis for Smoking-Attributable Disease*. 2010.

[3] R. A. Arrazola, I. B. Ahluwalia, E. Pun, I. Garcia de Quevedo, S. Babb, and B. S. Armour, "Current Tobacco Smoking and Desire to Quit Smoking Among Students Aged 13–15 Years — Global Youth Tobacco Survey, 61 Countries, 2012–2015," *MMWR. Morb. Mortal. Wkly. Rep.*, vol. 66, no. 20, pp. 533–537, 2017.

[4] P. Wu, J. W. Hsieh, J. C. Cheng, S. C. Cheng, and S. Y. Tseng, "Human smoking event detection using visual interaction clues," *Proc. - Int. Conf. Pattern Recognit.*, no. May, pp. 4344–4347, 2010.

[5] J. L. Obermayer, W. T. Riley, O. Asif, and J. Jean-Mary, "College smoking-cessation using cell phone text messaging.," *J Am. Coll. Health*, vol. 53, no. 2, pp. 71–8, 2004.

[6] H. Li and M. Trocan, "Deep learning of smartphone sensor data for personal health assistance," *Microelectronics Journal*, 2018.

[7] C. A. Cole, J. F. Thrasher, S. M. Strayer, and H. Valafar, "Resolving ambiguities in accelerometer data due to location of sensor on wrist in

application to detection of smoking gesture," in *2017 IEEE EMBS International Conference on Biomedical and Health Informatics, BHI 2017*, 2017, pp. 489–492.

[8] P. M. Scholl, N. Kücükyildiz, and K. Van Laerhoven, "When do you light a fire?: capturing tobacco use with situated, wearable sensors," in *UbiComp Adjunct*, 2013, pp. 1295–1304.

[9] J. P. Varkey, D. Pompili, and T. a. Walls, "Human motion recognition using a wireless sensor-based wearable system," *Pers. Ubiquitous Comput.*, vol. 16, pp. 897–910, 2011.

[10] Nazir Saleheen *et al.*, "puffMarker: A Multi-Sensor Approach for Pinpointing the Timing of First Lapse in Smoking Cessation.," *Proc. ... ACM Int. Conf. Ubiquitous Comput. . UbiComp*, vol. 2015, pp. 999–1010, 2015.

[11] P. Lopez-Meyer, S. Tiffany, and E. Sazonov, "Identification of cigarette smoke inhalations from wearable sensor data using a Support Vector Machine classifier.," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2012, pp. 4050–3, 2012.

[12] P. M. Scholl and K. van Laerhoven, "A Feasibility Study of Wrist-Worn Accelerometer Based Detection of Smoking Habits," in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2012, pp. 886–891.

[13] Q. Tang, D. Vidrine, E. Crowder, and S. Intille, "Automated Detection of Puffing and Smoking with Wrist Accelerometers," in *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, 2014.

[14] Y. Qin, W. Qian, N. Shojaati, and N. Osgood, "Identifying smoking from smartphone sensor data and multivariate hidden Markov models," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10354 LNCS, pp. 230–235.

[15] M. Shoaib, H. Scholten, P. J. M. Havinga, and O. D. Incel, "A hierarchical lazy smoking detection algorithm using smartwatch sensors," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services, Healthcom 2016*, 2016.

[16] F. Chollet, "Keras," *GitHub Repos.*, 2015.

[17] F. Pedregosa and G. Varoquaux, *Scikit-learn: Machine learning in Python*, vol. 12. 2011.

[18] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Complex Human Activity Recognition Using Smartphone and Wrist-Worn Motion Sensors," *Sensors*, vol. 16, no. 4, p. 426, 2016.

[19] A. Goodfellow, Ian, Bengio, Yoshua, Courville, "Deep Learning," *MIT Press*, 2016.

[20] M. Lin, Q. Chen, and S. Yan, "Network In Network," *arXiv Prepr.*, p. 10, 2013.

[21] L. Zhang, X. Wu, and D. Luo, "Recognizing Human Activities from Raw Accelerometer Data Using Deep Neural Networks," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 865–870.

Appendix B

We showed the F1 score of the baseline classifiers including the Support vector machine (SVM), Random forest (RF) Logistic regression (LogReg), and K-nearest neighbours (KNN) in order to compare the influence of the sampling methods in improving their F1 score. The sampling methods were the proposed distance-based (DBM), noise detection-based method (NDBM) and cluster-based method (CBM) . In addition, the six existing methods which was including, Synthetic Minority Oversampling Technique (SMOTE) [52], Random_SMOTE algorithm [53], SMOTE with Tomek links (SMOTE_Tomeklinks) [54], Modified Synthetic Minority Over-sampling Technique (MSMOTE) [55], Cluster-Based Synthetic Oversampling algorithm (CBSO) algorithm [56], and Proximity Weighted Synthetic Oversampling Technique (ProWSyn). The below figures compared the F1scores of the SVM, RF, LogReg and KNN on the Opportunity, PAMAP2 and ADL datasets. For more details about the dataset (see section 4.3.1)

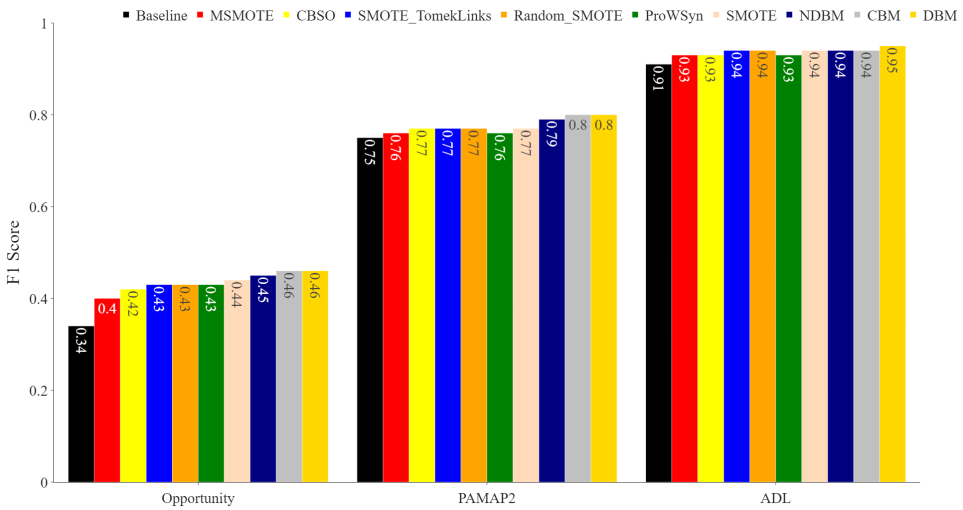

Figure B.1: The mean F1 score of baseline (SVM), the proposed methods, and the six existing sampling methods on the Opportunity, PAMAP2 and ADL datasets. The reported mean of F1 scores were obtained from 30 repetitions

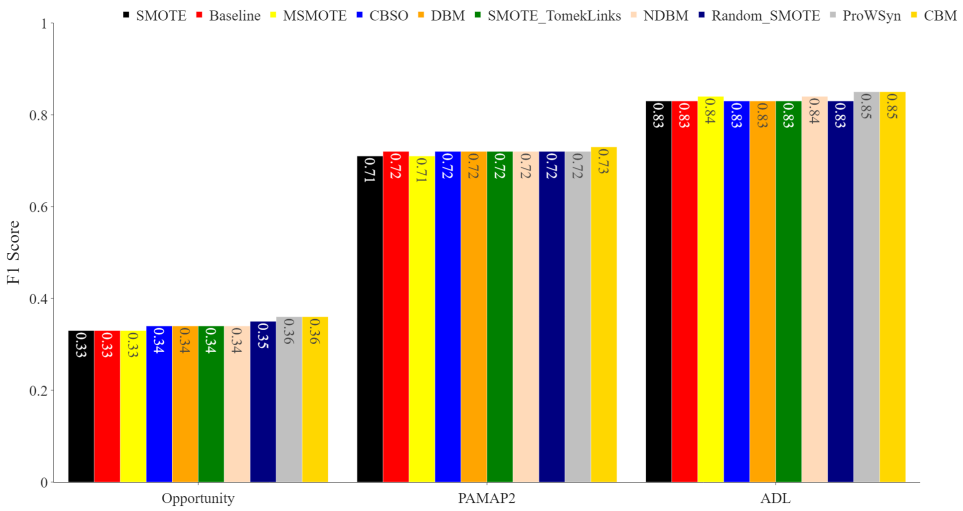

Figure B.2: The mean F1 score of baseline (RF), the proposed methods, and the six existing sampling methods on the Opportunity, PAMAP2 and ADL datasets. The reported mean of F1 scores were obtained from 30 repetitions

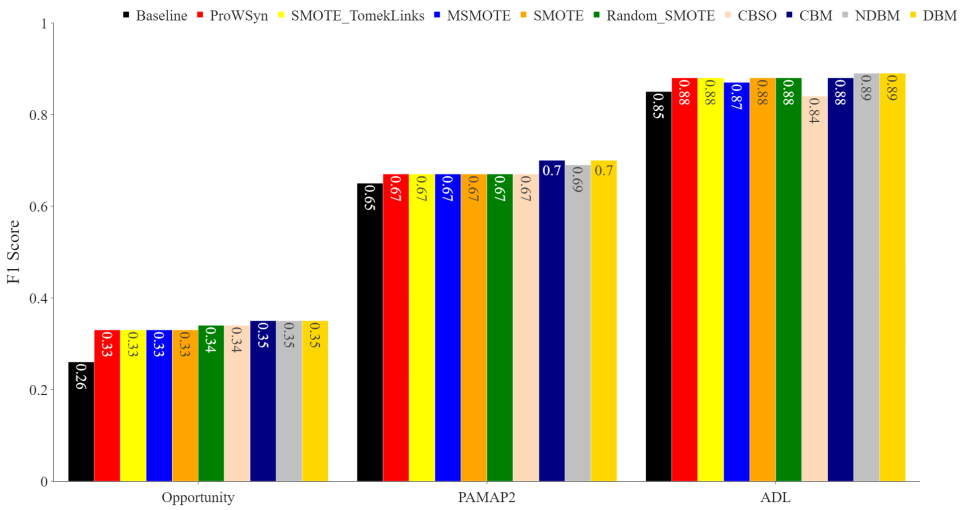

Figure B.3: The mean F1 score of baseline (LogReg), the proposed methods, and the six existing sampling methods on the Opportunity, PAMAP2 and ADL datasets. The reported mean of F1 scores were obtained from 30 repetitions

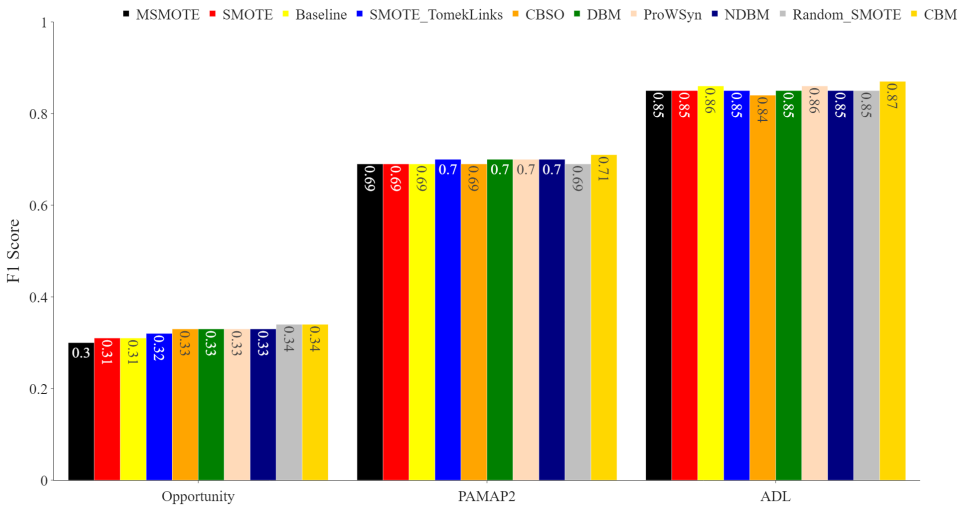

Figure B.4: The mean F1 score of baseline (KNN), the proposed method, and the six existing sampling methods on the Opportunity, PAMAP2 and ADL datasets. The reported mean of F1 scores were obtained from 30 repetitions