NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

RICARDO JOSÉ BATISTA OLIVEIRA

Licenciado em Ciências da Engenharia

DISSERTATION REPORT

HUMAN POSTURE CLASSIFICATION THROUGH PASSIVE RF SENSING

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
Janeiro, 2023

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

DISSERTATION REPORT

HUMAN POSTURE CLASSIFICATION THROUGH PASSIVE RF SENSING

RICARDO JOSÉ BATISTA OLIVEIRA

Licenciado em Ciências da Engenharia

Adviser: Rodolfo Oliveira
Associate Professor w/ Habilitation, NOVA University Lisbon

Examination Committee

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
Janeiro, 2023

Dissertation Report
Human Posture Classification through PAssive RF sensing

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.10.10) [**novathesis-manual**].

Acknowledgements

Firstly, I would like to express my heartfelt gratitude to my advisor, Professor Rodolfo Oliveira, for all his help, the knowledge he has shared with me, and his remarkable patience. I truly appreciate his constant availability to assist me throughout this journey. I am genuinely thankful for his guidance and support, which have been invaluable to my growth.

I would also like to thank PhD student Caio Eugene for his valuable assistance with the Variational Autoencoder. His willingness to share his expertise and address my questions was essential to the success of this work.

I am immensely grateful to my family, especially my parents and my sister, for their constant encouragement and belief in me. Their love and support have been a steady source of motivation.

Additionally, I want to express my appreciation to my girlfriend for her unwavering support and understanding during this journey. Your encouragement played a crucial role in my progress.

Lastly, I would like to acknowledge all my friends, both at the university and beyond, for their companionship and support throughout this journey. Your friendship has meant so much to me.

"Success is a combination of hard work,
perseverance, and talent."

Abstract

Human activity recognition through passive Radio Frequency (RF) sensing systems has gained attention in recent research due to its promising applications across various fields. These passive RF-sensing systems are often characterized by their cost-effectiveness, high portability, reliability, and non-intrusive nature.

In this thesis, we present a passive RF-sensing system that utilizes two Software Defined Radio (SDR) devices operating within the Very High Frequency (VHF) range, specifically at a frequency of 104.3 MHz. To examine and analyze human postures, we generated two experimental datasets. The first dataset encompasses the entire frequency range of an ??transmitter, spanning 0–100 kHz, while the second dataset focuses solely on the stereo pilot frequency of 19 kHz. These datasets were utilized to automatically identify three categories: standing, sitting, and a "none" class, which indicates the absence of a person.

A Variational Autoencoder (VAE) was employed to extract features from both datasets, alongside the use of statistical features. These features were subsequently classified using a neural network, achieving the highest accuracy of 92% in the offline scenario for the statistical features derived from the complete frequency band dataset, followed by 67% for the VAE features from the same dataset. For the dataset extracted for the stereo pilot, the maximum accuracy achieved was 46%. It was concluded that the information available solely from the stereo pilot is insufficient for effective classification in this context.

Additionally, a prototype was developed for real-time human posture classification using the complete frequency band dataset, yielding promising results.

The results obtained validate the selected features and underscore the effectiveness of the classification techniques applied in this domain, highlighting the significant potential of passive RF-sensing systems for human posture recognition, which encourages further research in this area.

Keywords: Machine Learning, Variational Autoencoder, Stereo Pilot, Human Activity Sensing, Passive RF Sensing, Automatic Recognition

Resumo

O reconhecimento de atividades humanas através de sistemas de detecção passiva de RF tem ganho atenção em pesquisas recentes devido às suas promissoras aplicações em várias áreas. Estes sistemas de detecção passiva de RF são frequentemente caracterizados pela sua relação custo-eficácia, alta portabilidade, fiabilidade e natureza não intrusiva.

Nesta tese, apresentamos um sistema de detecção passiva de RF que utiliza dois dispositivos de SDR operando na faixa de VHF, especificamente a uma frequência de 104,3 MHz. Foram gerados dois conjuntos de dados experimentais para examinar e analisar as posturas humanas, com o objetivo de identificar automaticamente três categorias: em pé, sentado e uma classe "none", que indica a ausência de uma pessoa.

Nesta tese, apresentamos um sistema de detecção passiva de RF que utiliza dois dispositivos de SDR operando na faixa de VHF, especificamente a uma frequência de 104,3 MHz. Foram gerados dois conjuntos de dados experimentais para examinar e analisar as posturas humanas, com o objetivo de identificar automaticamente três categorias: em pé, sentado e uma classe "none", que indica a ausência de uma pessoa. O primeiro conjunto de dados abrange toda a faixa de frequência de um transmissor Frequency Modulation (FM), de 0–100 kHz, enquanto o segundo foca exclusivamente na frequência do stereo pilot, de 19 kHz.

Foi utilizado um VAE para extrair características de ambos os conjuntos de dados, juntamente com a utilização de características estatísticas. Estas características foram subsequentemente classificadas utilizando uma rede neuronal, alcançando a maior precisão de 92% no cenário offline para as características estatísticas derivadas do conjunto de dados da banda de frequência completa, seguida por 67% para as características do VAE do mesmo conjunto de dados. Para o conjunto de dados extraído para o stereo pilot, a precisão máxima alcançada foi de 46%. Constatou-se que a informação disponível apenas a partir do stereo pilot é insuficiente para uma classificação eficaz neste contexto.

Além disso, foi desenvolvido um protótipo para a classificação de posturas humanas em tempo real, utilizando o conjunto de dados da banda de frequência completa, com resultados promissores.

Os resultados obtidos validam as características selecionadas e sublinham a eficácia das técnicas de classificação aplicadas neste domínio, destacando o potencial significativo dos sistemas de detecção passiva de RF para o reconhecimento de posturas humanas, o que incentiva futuras investigações nesta área.

Palavras-chave: Machine Learning, Variational Autoencoder, Stereo Pilot, Deteção de Atividades Humanas, Deteção Passiva de RF, Reconhecimento Automático

Contents

Annexes

List of Figures

List of Tables

Acronyms

ADC analog-to-digital converter *(p. 32)*
AE Autoencoders *(p. 16)*
ANN Artificial Neural Network *(pp. 15, 16)*
ANOVA Analysis of variance *(pp. 14, 21)*
AoA Angle of Arrival *(pp. 3, 10, 20)*
AP Access Point *(p. 21)*

CFAR Constant False Alarm Rate *(p. 8)*
CNN Convolutional Neural Network *(pp. xi, 3, 15, 16, 20, 21)*
CSI Channel State Information *(pp. 3, 9, 10, 20, 21)*

dbm decibels miliwatts *(pp. 8, 9)*
DFAR Device Free radio-based activity recognition *(p. 20)*
DFTs Discrete Fourier Transforms *(p. 12)*
DL Deep Learning *(pp. 15, 21)*

FFT Fast Fourier Transform *(pp. 3, 11, 36, 50–52)*
FM Frequency Modulation *(pp. vi, 1, 2, 6, 20, 22, 29–33, 61, 65, 73, 76)*
FPGA Field-Programmable Gate Array *(pp. 23, 27)*

GRC GNU Radio Companion *(p. 28)*
GUI Graphical User Interface *(p. 28)*

HMM Hidden Markov Model *(pp. 3, 17, 20)*

I/Q In-phase and Quadrature *(p. 32)*
IoT Internet of Things *(p. 1)*
IQR Interquartile Range *(p. 20)*

KL Kullback-Leibler *(p. 37)*
KNN K-Nearest Neighbors *(pp. 3, 17, 18, 20, 21)*

LNA Low-Noise Amplifier *(p. 23)*
LOS Line-of-Sight *(p. 20)*
LSTM Long-Short Term Memory *(p. 20)*

MAD Median Absolute Deviation *(p. 20)*
MIMO Multiple-Input Multiple-Outpu *(pp. 9, 23)*
ML Machine Learning *(pp. 14, 16, 18)*
MSE Mean Squared Error *(p. 37)*

NIC Network Interface Controller *(p. 21)*
NLOS Non-Line-of-Sight *(p. 20)*

OFDM Orthogonal Frequency-Division Multiplexing *(pp. 8, 9)*
OSVM One-Class Support Vector Machine *(p. 20)*

PAR Photosynthetically Active Radiation *(p. 29)*
PIR Passive Infrared *(p. 21)*
PSD Power Spectral Density *(pp. 12, 13)*
PSE Power Spectral Entropy *(p. 13)*

Radar Radio Detection and Ranging *(p. 1)*
RBDS Radio Broadcast Data System *(p. 31)*
ReLu Rectified Linear Unit *(pp. 37, 55, 56)*
RF Radio Frequency *(pp. v, vi, 1, 3, 10, 20, 23, 26, 29)*
RFID Radio-Frequency Identification *(p. 21)*
RSSI Received Signal Strength Indicator *(pp. 3, 8, 9, 20, 21)*

SAR Synthetic Aperture Radar *(p. 21)*
SDR Software Defined Radio *(pp. v, vi, 21, 23, 25, 28)*
STD Standard Deviation *(p. 20)*
SVM Support Vector Machine *(pp. 3, 18, 21)*

TDOA Time Difference Of Arrival *(p. 10)*

VAE Variational Autoencoder *(pp. v, vi, 2, 3, 36, 37, 42, 46, 49, 56, 57, 66–72, 76, 77, 82, 88, 101, 127, 133, 139)*
VHF Very High Frequency *(pp. v, vi, 29)*

1 Introduction

1.1 Motivation

As the Internet of Things (IoT) continues to gain prominence, systems capable of monitoring people's posture, movement, and location have appeared in many IoT applications. These applications are increasing since applying this type of system can significantly improve society's lifestyle. These systems can be applied in several contexts, the most common being smart homes and health scenarios.

Usually, this type of monitoring system uses devices that the people themselves have to carry with them or cameras. This type of device contains very advanced hardware, which is expensive and requires maintenance, apart from privacy issues of the people who carry it and other people that may be captured.

In recent years, the number of studies on RF for human activity monitoring has increased because this type of system does not require devices to be carried, making it cheaper, much more flexible, and reliable.

Radio Detection and Ranging (Radar) technology has become critical in these contexts. There are two types of radar: active and passive. The active ones have their transmitters, and the passive ones use signals of interest already available in the area (illuminators). Passive radars have some advantages over active radars: for example, there is no need to buy bandwidth to operate the transmitter, which is becoming increasingly complicated, and their hardware is much simpler and, therefore, cheaper.

Notably, this study focuses on utilizing FM radio bands as illuminators due to their continuous signal transmission characteristic. This ensures a constant source of information, reducing the potential loss of critical data compared to other signals that may exhibit intermittent transmissions.

Currently, numerous applications of passive Radar exist, such as fall detection [1], human motion analysis [2], and presence detection [3].

1.2 Goals

This thesis aims to explore several key areas within the domain of passive sensing and human posture classification. The first objective is to investigate the feasibility of utilizing different frequency bands for passive sensing applications. This study will focus on two main scenarios: one considering the assumption of a limited band, specifically a 19 kHz pilot, and the other exploring the potential of utilizing the entire FM band dedicated to a single broadcast station.

The use of FM radio in this study allows us to examine the advantages of having a continuous signal over time, which can provide consistent information for sensing applications. By exploring these two approaches, we aim to assess their viability for robust sensing applications

The second objective is to analyze how various features perform within the context of posture classification. Two distinct types of features will be explored in this study: statistical features and features derived from a VAE. By investigating these feature sets, we seek to determine their effectiveness in improving classification accuracy and robustness, particularly in passive sensing environments.

Finally, the third objective is to characterize the classification of human posture under varying conditions. This characterization will take into account the different frequency bands mentioned above, as well as the feature sets and classification methods employed. Through this analysis, the study aims to identify the optimal combination of frequency bands, features, and classification techniques to enhance the accuracy and efficiency of human posture classification.

1.3 Contributions

This research makes several significant contributions to the field of passive sensing and human posture classification. First, we gathered two distinct datasets: one corresponding to a limited band scenario, specifically the 19kHz pilot, and another considering the full FM band utilized by an FM broadcast station. These datasets enabled a comparative analysis of the sensing potential of both bands.

Our results demonstrate that the features derived from the limited band dataset are of lower quality for classification purposes when compared to those obtained from the complete FM band. The complete FM band provides a more robust solution, as it enables sensing across the entire band, thereby minimizing information loss compared to the case of the stereo pilot.

We also propose a novel methodology for feature extraction based on a VAE. The quality of the features generated by this method is compared to traditional statistical features. This comparison offers valuable insights into the strengths and limitations of both approaches for passive sensing applications.

Lastly, we characterize the performance of different classification schemes, particularly using neural networks. For the full FM band scenario, the best classification accuracy achieved with features derived from the VAE was 67%. In contrast, when using statistical features, the accuracy reached 92%, demonstrating the superior performance of traditional statistical features. For the limited 19kHz band scenario, classification was performed exclusively with VAE features, yielding a maximum accuracy of 47%.

1.4 Thesis Structure

This document is organized to provide a thorough examination of the research conducted on passive sensing and human posture classification. The structure begins with Chapter 1, which serves as the introduction. This chapter outlines the motivation behind the study and defines the research goals, setting the context and objectives for the entire research.

Following Chapter 1, Chapter 2 reviews the literature relevant to passive radar. It starts with an overview of passive radar, including its historical background, sources of illuminators, and principles of operation. This chapter also covers various input features such as Received Signal Strength Indicator (RSSI), Channel State Information (CSI), and Angle of Arrival (AoA), and explores different classification features and methods. Among the methods discussed are CNNs, Autoencoders, Hidden Markov Model (HMM), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). Additionally, Chapter 2 addresses the state-of-the-art in passive RF sensing.

Chapter 3 describes the experimental setup used in the research and the methodology for acquiring data. It includes detailed information on data characterization, dataset extraction, and downsampling processes, which are crucial for preparing the data for subsequent analysis.

In Chapter 4, various types of features are examined. This chapter includes an analysis of features derived from a VAE for both the full band and the 19 kHz band, as well as statistical features, phase features, and the Fast Fourier Transform (FFT) features.

Chapter 5 presents the results from different classification methods. It discusses the application of VAE and statistical features in classification tasks, introduces performance metrics, and provides a detailed analysis of the results and their implications. This chapter also includes information about a real-time prototype developed as part of the research.

Finally, Chapter 6 summarizes the key findings of the research, offers final considerations, and suggests potential directions for future work. This concluding chapter ties together the research findings and provides recommendations for further investigation.

2 Related Work

2.1 Passive Radar

There are two types of radar systems: active and passive.

Active radars, known as conventional monostatic radars, use their dedicated transmitter and a single antenna for transmitting and receiving. The form of the signal, which is usually pulsed, is optimized solely for the radar function. These systems emit pulses of radio waves and analyze the returned signals to determine the range, directions, and speed of targets.

Passive radar systems have emerged as a promising alternative to conventional radars and are the focus of this master thesis. These systems do not require a dedicated transmitter for sensing and instead rely on illuminators as their transmission source, which are electromagnetic waves transmitted by other sources not optimized for radar purposes. By sensing disturbances in the electromagnetic field caused by a target's reflection, passive radars can detect and track objects with comparable accuracy to conventional radars. One of the major advantages of passive radar is its cost-effectiveness, as it does not require licensing and is thus more accessible to organizations or individuals who cannot afford or obtain conventional radar. The passive radar is also undetectable as long as the receive antenna is inconspicuous. However, one limitation of passive radar is that the transmitting source is usually beyond the system's control and may not be optimized for radar purposes, thus affecting its performance [4][5].

Given the uncertainty regarding the specific illuminators that will be used, it is necessary to measure both the direct and reflected signals on the target. The passive radar system must then preprocess the signals, perform cross-correlation, and track the target. These principles will be explained in more detail in Subsection 2.1.3. Fig. 2.1 provides a typical example of target detection with passive radar, which involves a reference antenna and a surveillance antenna. The direct signal is received by the reference antenna, while the surveillance antenna receives the reflected signals from the target.

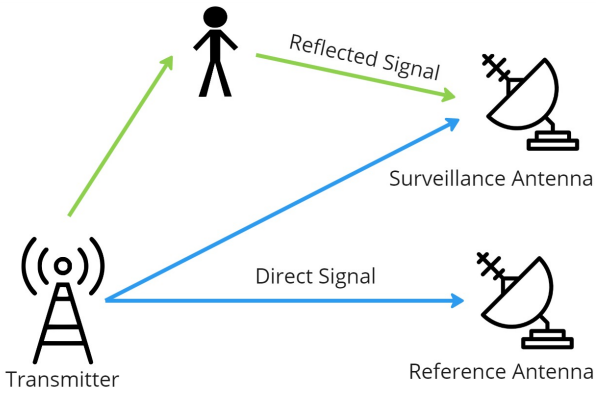

Figure 2.1: Passive Radar Arquitecture.

2.1.1 History

Upon recognizing radar as a means of detecting air or naval targets at extended ranges, the military identified a significant issue with radar's localization method. This method was highly vulnerable, as it only functioned when energy was transmitted. Consequently, determining the location of radars was relatively easy, as typical radar signals could be received and triangulated, resulting in their neutralization.

During the early development of radar technology, the localization method was highly susceptible to jamming and destruction, as it could only work when energy was transmitted. This led to a demand for more effective and less revealing radar systems. The first radar experiments in the United States in 1922 were led by A. Taylor and L.C. Young, which were bistatic and involved the use of a transmitter and receiver at separate locations [4].

In 1924, Appleton and Barnett documented the first use of a broadcast transmitter for radar purposes [4]. They aimed to measure the height of the Heaviside layer and used a broadcast radio transmitter located at Bournemouth, operating at 770 kHz, and a receiver located at Oxford. Another significant event in radar history was the Daventry experiment in 1935, in which a British Broadcasting Corporation broadcast transmitter, operating at a frequency of approximately 6 MHz, was used for aircraft target recognition at a range of 13 km.

After demonstrating the results to a senior civil servant, the British Air Ministry supported a development program that resulted in the British Chain Home air defence radar system just before the outbreak of World War II. The bistatic radar system Klein Heidelberg was the first example of an operational passive radar, a German radar device from WWII that utilized British Chain Home radars as a travelling illumination source. The first Klein Heidelberg was put into service in 1943 and proved valuable in German air defence when conventional early warning radar was affected by jamming and other countermeasures.

Although bistatic radar was helpful in some specific applications, the extra intricacy of bistatic activity did not offer any significant benefit or capacity after the end of WWII, and interest in it subsequently decreased [4].

Between 1980 and 2000, passive radar technology underwent a transformative phase, primarily fueled by decreasing computing costs. This period witnessed significant advancements in coverage and reliability through innovative approaches using various types of transmissions [5].

The pivotal role played by advancements in signal processing algorithms cannot be overstated, as they enhanced passive radar performance and expanded its applications beyond military uses. These improvements led to using passive radars in gap-filling roles within conventional radar coverage and air traffic control for cost-effective situational awareness.

This transformative era laid a solid foundation for the continued evolution and integration of passive radar systems into modern defence and surveillance architectures.

2.1.2 Sources of Illuminators

Selecting a suitable illuminator for a passive radar system is crucial for its effectiveness. This external electromagnetic energy source allows the radar to operate covertly, leveraging existing signals and reducing electronic emissions. The choice impacts the system's stealth, coverage, and adaptability to different environments. A well-matched illuminator enhances signal processing, contributing to better target detection and identification. The most critical parameters to be considered when choosing an illuminator are power density, the nature of the waveform, and the coverage. The main source of iluminatores examples are [4][5]:

- Analog Television
- FM Radio Signals
- GPS Satellites
- Cellular Phone Base Stations
- Digital Video and Audio Broadcasting

2.1.3 Principle of Operation

In a traditional radar system, precise information about the pulse transmission time and the transmitted waveform is readily available. This knowledge facilitates straightforward calculations of object range and enables the use of a matched filter for optimal signal-to-noise ratio in the receiver. In contrast, a passive radar lacks direct access to such information. Consequently, it relies on a specialized receiver channel, called the "reference channel," to continuously monitor each utilized transmitter. The passive radar dynamically samples the transmitted waveform, requiring the implementation of specific processing steps. The functioning of a passive radar system relies on the cross-correlation between

the direct signal transmitted from an illuminator and the reflections it receives from a specific target.

At this juncture, each step of the passive radar process will be enumerated and subsequently elucidated with precision:

- Reception of the direct and surveillance signal from the transmitter
- Digital Beamforming
- Transmitter-specific signal conditioning
- Adaptive filtering
- Cross-correlation of the reference channel with the surveillance channels
- Target Detection
- Target Tracking

Reception of the signal: A passive radar system is designed to detect small target reflections amid strong and continuous interference. In contrast to conventional radar, which listens for echoes during signal gaps, the passive radar receiver requires low noise, high dynamic range, and strong linearity. Despite these requirements, received echoes are usually well below the noise floor, with the system primarily limited by external noise sources, including the transmitted signal and other distant in-band transmitters. Passive radar systems utilize digital receivers, producing a digitized, sampled signal.

Digital Beamforming: Determining signal arrival direction constitutes a pivotal procedure achievable through conventional radar beamforming techniques. This process involves amalgamating antenna array elements, inducing constructive interference at specific angles and concurrently yielding destructive interference at others. Such signal processing methodology serves the dual purpose of discerning desired signals while mitigating the impact of substantial interference from frequency overlap within the operational bandwidths of transmitters and receivers.

Transmitter-specific signal conditioning: Certain transmitter types require specific signal conditioning procedures before cross-correlation processing. These procedures encompass high-fidelity analogue bandpass filtering, channel equalization to enhance the reference signal's quality, eliminating undesired structures within digital signals to optimize the radar ambiguity function, and, in some cases, comprehensive reconstruction of the reference signal from the received digital signal.

Adaptive filtering: A consequential subsequent phase involves attenuating the direct signal within the surveillance region, as the Doppler sidelobes associated with it could obscure faint target reflections during the cross-correlation stage. Consequently,

adaptive filters are commonly employed in the surveillance channel(s) to eliminate the direct signal.

Cross-correlation: The central processing step in passive radar involves cross-correlation, acting as a matched filter and providing bistatic range and Doppler shift estimates for each target echo. Due to the noise-like nature of most broadcast signals, correlation primarily occurs within individual signals, posing a challenge for moving targets with Doppler shifts. To address this, cross-correlation processing employs a bank of matched filters, each tailored to a specific target Doppler shift. Efficient implementations based on the discrete Fourier transform, especially for Orthogonal Frequency-Division Multiplexing (OFDM) waveforms, are common, yielding a signal processing gain typically quantified by the time-bandwidth product often reaching 50 dB. Extended integration times are limited by target motion, causing smearing in range and Doppler during integration [6].

Target Detection: Targets are identified on the cross-correlation surface by applying an adaptive threshold, categorizing all returns surpassing this threshold as targeted. A common approach employs a the Constant False Alarm Rate (CFAR) algorithm.

Target Tracking: Additionally, tracks of identified targets are generated over time in the range-Doppler domain. Typically, a standard Kalman filter is employed for this purpose. This processing step effectively rejects the majority of false alarms. Finally, the line tracks originating from each transmitter are associated and consolidated to ascertain the object's location and speed.

2.2 Input Features

Input features are the measurable and distinct characteristics or variables provided as input to a system, model, or algorithm for analysis or processing. In data science, machine learning, and statistical modeling, input features are the attributes or dimensions of the data that contribute to understanding or predicting a particular phenomenon. In this context, a feature is a unique quantifiable aspect of an event being analyzed. It is a numeric representation of the raw data derived from a specific process.

2.2.1 RSSI

RSSI is a metric in wireless communication that quantifies the strength of the signal received by a device from a transmitter. Expressed in decibels miliwatts (dbm), RSSI provides insight into the power level of the received signal. Typically measured in voltage per length received by a reference antenna, it helps estimate the distance between the transmitter and receiver. In the context of low-power systems, units like dBµV/m are common. RSSI is crucial in assessing signal quality, with values near -60 dbm indicating a solid signal and near -100 dbm suggesting a weaker signal. This metric is widely

used in technologies like IEEE 802.11 to monitor and optimize wireless communication performance.

RSSI can be expressed through

$$RSSI = P_t - P_L(d), \tag{2.1}$$

where P_t represents the transmitted signal power, $P_L(d)$ denotes the reduction in power density of the signal (commonly referred to as path loss) when the separation between a transmitter and receiver antenna is d, and both quantities are expressed in dbm. Under the assumption of the LogNormal Shadowing Model for the loss model, the path loss can be articulated as

$$P_L(d) = P_L(d_0) - 10n\ log(\frac{d}{d_0}) + X_\sigma, \tag{2.2}$$

where $P_L(d_0)$ represents the path loss at a reference distance d_0, n stands for attenuation factor, reflecting the characteristics of the signal propagation environment. Additionally, X_σ is a Gaussian random variable representing both multipath and shadow fading effects, represented as $X_\sigma \sim (0, \sigma^2)$ [7].

2.2.2 CSI

CSI delivers an intricate depiction of wireless channel characteristics, offering detailed insights into signal propagation complexities. It unveils attributes such as fading, power decay, and scattering, directly influencing the amplitude and phase of the wireless signal. In contrast, RSSI is deemed coarse-grained compared to CSI. RSSI, relying on packets, furnishes a mean value for received signals. In contrast, CSI, grounded in channel estimation, permits fine-grained measurements for each OFDM subcarrier [8].

In the context of a Multiple-Input Multiple-Outpu (MIMO)-OFDM WiFi system with multiple transmit and receive antennas, CSI manifests as a three-dimensional matrix containing complex values. These values represent the amplitude attenuation and phase shift of multipath WiFi channels. Over time, performing CSI measurements enables the capture of how wireless signals traverse through targets across temporal, spatial, and frequency domains. This capability makes CSI valuable for a broad range of wireless sensing applications.

CSI is estimated at the receiver through a process known as Channel Estimation. Typically, the estimated CSI is quantized and fed back to the transmitter. It is important to note that the transmitter's CSI and the receiver's CSI may differ due to the quantization process.

The adaptability of transmissions to instantaneous channel conditions is crucial for achieving reliable communications with high data rates. Therefore, CSI becomes a valuable tool in this regard.

There are two commonly utilized types of CSI, namely Instantaneous CSI and Statistical CSI. Instantaneous CSI provides knowledge of the current channel conditions, allowing the adaptation of transmitted signals in real time. This helps in reducing bit error rates

and improving the overall performance of the system. On the other hand, Statistical CSI provides knowledge of the statistical characterization of the channel, including information such as average channel gain, spatial correlation, fading distribution type, and line-of-sight components. Statistical CSI is more appropriate in situations with rapidly changing channel conditions (fast fading), while Instantaneous CSI is suitable for slow-fading systems where channel conditions remain roughly constant.

Practical scenarios often involve CSI lying between these two types. Therefore, choosing an appropriate CSI estimation method becomes critical. Accurate CSI estimation can lead to better system performance, efficient utilization of resources, and enhanced user experience.

2.2.3 AoA

AoA is a crucial parameter in wireless communication representing the angle from which a RF signal is received. This directional information is essential for antenna arrays and is determined through three passive direction-finding techniques: amplitude comparison, Time Difference Of Arrival (TDOA), and phase interferometry [9].

The amplitude comparison technique involves two directional antennas with distinctive gain ratios that define angles within the visual area. The AoA is then derived from gain patterns obtained through these ratios. On the other hand, the TDOA technique calculates AoA and transmitter range by measuring signal arrival time differences, and its accuracy is contingent on the distance between antennas. Phase interferometry, another technique, measures signal phase differences at each antenna in the array, providing a unique approach to AoA determination.

AoA finds application in wireless sensor networks, radar systems, and communication networks, contributing to precise localization and efficient resource allocation. Despite challenges like multipath propagation and environmental factors, ongoing research focuses on advanced signal processing and innovative antenna configurations to refine AoA estimation.

2.3 Classification Features

In machine learning, effective feature selection is crucial when dealing with large datasets containing multiple variables and features. The aim is to develop a model with a reduced number of parameters, which helps in achieving accurate classifications, mitigating overfitting, shortening training duration, and preventing the curse of dimensionality. Features can be categorized based on their characteristics and acquisition method and further classified based on their domain. In this specific case, features can be classified into two main domains: time-domain and frequency-domain, which will be discussed in detail in the following subsections.

2.3.1 Time-Domain Features

Time domain features are extrapolated from time series data, comprising a sequence of data points amassed over a specific timeframe. The signals intercepted by the radar exist within the temporal realm before undergoing FFT. Several of these features will now be introduced.

Raw Moments: Raw moments are statistical measures that provide insights into the shape and tendency of a probability distribution or a set of data points. These moments are calculated by raising the values of the data points to a certain power and then averaging them. The first four raw moments are Mean, Variance or Standard Deviation, Skewness, and Kurtosis value, respectively. The expression for the i-th raw moment is given by

$$\mu_i' = \frac{1}{N}\sum_{n=1}^{N} x_n^i, \tag{2.3}$$

where x_n denotes the signal amplitude value for each sample from the dataset, while N represents the size of the fixed window of samples.

Mean: is a central tendency measure calculated by summing up all values in a dataset and dividing by the total count, which can be expressed as

$$\mu_1' = \frac{1}{N}\sum_{n=1}^{N} x_n. \tag{2.4}$$

Variance: is a statistical metric that gauges the degree of dispersion or spread of a dataset. It quantifies how far individual data points deviate from the mean, emphasizing the variability within the data. It can be computed as

$$\text{var} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_1')^2. \tag{2.5}$$

Standard Deviation (STD): quantifies the amount of variability or dispersion in a dataset. It provides a measure of how spread out the values are around the mean.It is determined by

$$\text{std} = \sigma = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_1')^2}. \tag{2.6}$$

Skewness: quantifies the deviation of a signal's probability distribution, depicted by a random variable, from the normal distribution. It serves as a measure of asymmetry in the probability distribution of the signal.

$$\text{skew} = \frac{\mu_3}{\sigma_3} = \frac{\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_1')^3}{\sigma^3}. \tag{2.7}$$

Kurtosis: characterizes how values cluster in the tails or the peak of a probability distribution in a signal represented by a random variable. It provides insights into the distribution's shape, highlighting whether it has heavy or light tails compared to a normal distribution , and is represented by

$$\text{kurt} = \frac{\mu_4}{\sigma_4} = \frac{\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_1')^4}{\sigma^4}. \tag{2.8}$$

Central Moments: is a normalized version of a raw moment in statistics. Unlike raw moments, central moments are adjusted based on the mean, making them more robust to changes in scale. This adjustment enhances their stability and reliability, making central moments valuable for analyzing probability distributions and capturing underlying characteristics with increased sensitivity. It is defined as

$$\mu_i = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_1')^i. \tag{2.9}$$

Median Absolute Deviation (MAD): is a statistical measure that calculates the median distance between each sample x_i and a central point in a dataset x. This robust metric provides valuable information about the dispersion or variability of the data, offering resilience to the influence of outliers. MAD is especially useful in situations where traditional measures of variability might be sensitive to extreme values, as it relies on the median for a more robust assessment of spread. It can be represented as

$$\text{MAD} = \text{median}(|x_i - m(x)|), \tag{2.10}$$

where $m(x)$ represents the central point.

Root mean square (RMS): also known as the quadratic mean, serves as a measure for characterizing the effective or average power of a time-varying signal. It can be represent as follows

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}x_n^2}. \tag{2.11}$$

Peak to Peak value (PP): is value of a signal represents the amplitude variation by measuring the difference between its highest and lowest points over a specific time interval. It may be defined as

$$\text{PP} = |max(x(i)) - min(x(i))|. \tag{2.12}$$

2.3.2 Frequency-Domain Features

Typically, features retrieved in the frequency domain are identified using the signal's Power Spectral Density (PSD) estimate. A sequence of Discrete Fourier Transforms (DFTs) are applied to the time series signal in order to transform it into the frequency domain. Some of these features will then be made available.

Spectral Moments: refer to the statistical measures used to analyze the distribution of frequencies in a signal. In this context, M represents the length of the frequency vector. The i-th spectral moment is computed by

$$M_i = \sum_{m=0}^{M} f_m^i |X(f_m)|^2. \tag{2.13}$$

The moments provide valuable insights into the characteristics and properties of the frequency content within a given signal, aiding in the interpretation and understanding of the spectral information.

Total Power (TTP): is the sum of squared magnitudes of all frequency components in a signal, representing its overall energy. This metric is vital in signal processing, offering a comprehensive measure of signal intensity across its entire frequency spectrum. Now, please provide the formula for further clarification. It can be represented as

$$\text{TTP} = M_0 = \sum_{m=0}^{M} |X(f_m)|^2. \tag{2.14}$$

Mean Frequency (MNF): is calculated as the weighted average of the frequencies in a signal, reflecting the central tendency of its spectral content. This metric is commonly employed in signal analysis to characterize the typical frequency at which the signal's energy is concentrated. It can be computed as

$$\text{MNF} = \frac{M_1}{\sum_{m=0}^{M} |X(f_m)|^2} = \frac{\sum_{m=0}^{M} f_m |X(f_m)|^2}{\sum_{m=0}^{M} |X(f_m)|^2}. \tag{2.15}$$

Power Spectrum Deformation: serves as a concise measure of spectrum asymmetry, revealing crucial details about its deformation characteristics. In the realm of signal processing, this metric plays a pivotal role, aiding in the identification of irregularities within the frequency distribution. It is defined as

$$\omega = \frac{\sqrt{\frac{M_2}{M_0}}}{\frac{M_1}{M_0}}. \tag{2.16}$$

Power Spectral Entropy (PSE): quantifies the frequency-domain uncertainty in a signal, drawing from the concept of Information Entropy (IE). Entropy, characterized by a lack of order or predictability, gauges the disorder within the information intended for transmission or reception. To compute Power Spectral Entropy (PSE), the PSD must be initially determined as

$$S(f_i) = \frac{1}{M} |X(f_i)|^2. \tag{2.17}$$

PSE is defined as

$$H = -\sum_{i=1}^{M} p_i log(p_i). \tag{2.18}$$

2.3.3 Feature Selection

When dealing with complex real-world scenarios, it is crucial to identify significant data features, particularly in high-dimensional datasets [10]. This process, commonly referred to as feature selection, can be executed through various methodologies, each with different criteria for assessing feature relevance and seamless integration with broader Machine Learning (ML) workflows. A thorough examination of these approaches is essential in improving model efficiency and interpretability. This section will focus on filter techniques and highlight their importance and application in the feature selection landscape.

Filter Methods: serve as feature selection techniques that independently evaluate and rank features, creating subsets with the most relevant ones. Examples include Pearson's Correlation Coefficient and mutual information criteria like Shannon's Entropy [11]. Another method, Analysis of variance (ANOVA), assesses statistical differences in means among groups. These methods are simple to implement and computationally efficient since they do not involve training the ML model. This makes filter methods advantageous for large datasets or resource-constrained scenarios, improving overall model performance by simplifying models and enhancing interpretability.

Wrapper Methods: involve an approach to feature selection that requires an exhaustive search for a subset of features that can optimize the classifier's performance. Despite being computationally intense, these methods provide an advantage over filter methods as they excel in retaining features that demonstrate value when combined, resulting in higher accuracy compared to filter methods [12]. Additionally, the performance of Wrapper Methods is evaluated using accurate modeling algorithms, which provides a more realistic assessment of their efficiency.

Examples of wrapper methods include genetic algorithms, sequential feature selection algorithms, and recursive feature elimination. These techniques operate by iteratively evaluating a machine learning model's performance with subsets of features, optimizing the subset based on the highest classifier performance. This structured approach to wrapper methods ensures a comprehensive understanding of their functionality, benefits, and practical applications.

Embedded Methods: operating within the model training process using ensemble learning, striking a balance between computational expense and classification accuracy. Unlike filter methods, they consider the model in the feature selection process, contributing to heightened accuracy. While not as powerful as wrapper methods, embedded methods offer a cost-effective alternative due to their computational efficiency [13]. Additionally, this approach captures the interplay between feature relevance and the model, providing a nuanced understanding of feature importance within the learning process.

2.4 Classification Methods

2.4.1 Convolutional Neural Networks (CNN)

In this sub-section, the focus is placed specifically on CNNs due to their proven effectiveness in tasks related to feature extraction and pattern recognition, which are central to human posture classification. While there are various types of neural networks, CNNs are particularly well-suited for the nature of the data and the goals of this study CNNs, commonly known as ConvNets, are a specialized type of Artificial Neural Network (ANN) that have been specifically designed to perform tasks such as natural language processing, image segmentation, recognition, and classification [14][15]. These sophisticated Deep Learning (DL) algorithms intricately process input data by assigning biases and weights to different components, enabling a nuanced understanding and classification of each aspect. CNNs operate on the principle of hierarchical feature learning, which differs from hand-engineered feature methods. Instead of relying on manual feature extraction, the network autonomously learns hierarchical data representations through its layers. This feature extraction capability makes CNNs particularly adept at discerning complex patterns and structures within the input, a trait highly beneficial in various domains. The fundamental architecture of a CNN comprises three main building blocks: convolution, pooling, and fully connected layers [16][17]. The convolutional and pooling layers focus on extracting and consolidating meaningful features, while the fully connected layer refines these features for the final output. The synergy of these blocks enables CNNs to capture intricate hierarchical patterns within the data. Compared with traditional classification algorithms, CNNs present a notable advantage in reduced pre-processing requirements. Unlike methods that demand manually crafted filters for feature extraction, CNNs autonomously learn relevant features from the raw data. Nevertheless, it is essential to acknowledge that this autonomy comes at the cost of increased computational complexity. The learning process involves estimating numerous parameters, making CNNs more computationally intensive [15]. A typical representation of a CNN is depicted in Figure 2.2 and will be now described.

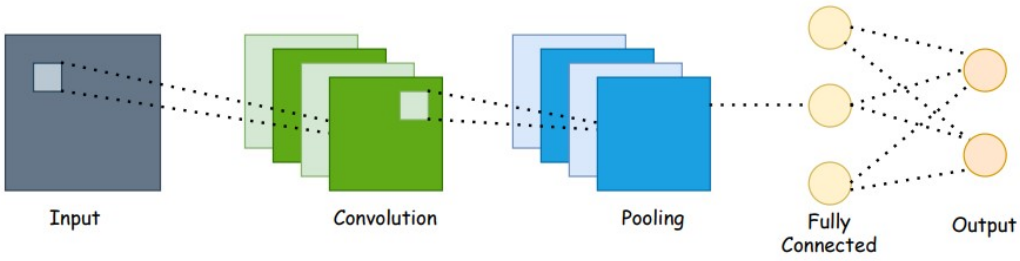

Figure 2.2: Schematic diagram of a CNN architecture (adapted from [16]).

Convolution Building Block: extracts unique features from input data. It uses a mathematical operation called convolution, where a filter or kernel slides across the input

to capture spatial hierarchies and patterns. As it slides, the filter multiplies its weights with the corresponding input values, generating a dot product that represents the presence of certain features or patterns in that particular region of the input. These dot products create feature maps highlighting the spatial features and hierarchies present in different input regions. The stride and padding determine the movement of the filter across the input, while non-linear activation functions introduce non-linearity and adjust the weights of the filters during the training process. The Convolution Building Block is foundational for capturing local patterns and hierarchies in the input data, making CNNs ideal for tasks like image recognition, where patterns often appear in specific regions of an image.

Pooling Building Block: collaborates to down-sample feature maps with the Convolution Building Block. It employs pooling operations on small regions of the input data, such as max pooling or average pooling. By reducing the spatial dimensions, pooling provides computational efficiency and introduces translation invariance. The stride determines the step size, which impacts down-sampling speed. While pooling reduces dimensionality and robustness, it may sacrifice fine-grained spatial information. Common techniques include max pooling and average pooling, with adaptive pooling adjusting window size dynamically. The Pooling Building Block is integral for optimizing CNN efficiency and robust feature extraction.

Fully Connected Building Block: maps extracted features to the final output. Neurons in this block are connected to all neurons in the preceding layer, facilitating comprehensive information processing. During training, learnable parameters, including weights and biases, are adjusted through backpropagation. This block enables the network to capture complex relationships and predict based on the learned features. While powerful, fully connected layers can be computationally expensive and prone to overfitting. Their inclusion completes the CNN architecture, bridging localized feature extraction and the final output layer.

2.4.2 Autoencoders

Autoencoders (AE) are a type of ANN that is well-suited for unsupervised learning within the domain of ML. This approach is particularly effective in handling unlabelled datasets. The fundamental principle underlying an autoencoder is the compression of input data into a code, from which the output is subsequently generated to function as a reconstruction of said code. The iterative refinement of this output is achieved by comparing it with the input data by applying a specific cost function, thus facilitating the attainment of an optimal outcome. The training procedure relies on the principles of backpropagation.

A typical autoencoder comprises three layers: the input layer, commonly called the encoder; an intermediate hidden layer housing the code; and the output layer, designated

as the decoder in Figure 2.3. The encoder component plays a pivotal role in generating the code by compressing the input data, while the decoder endeavors to reconstruct the input data by leveraging the acquired code, producing the final output.

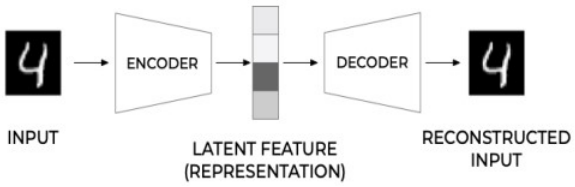

Figure 2.3: Schematic of Autoencoder Arquitecture (copied from[18]).

It is noteworthy that the generated output, while containing fewer features, retains the crucial aspects of the input data. This characteristic renders autoencoders particularly valuable for dimensionality reduction and feature learning tasks. Both the encoder and decoder components function as fully connected feedforward neural networks, contributing to the efficacy of the autoencoder in learning meaningful representations from unlabelled data [19].

Furthermore, the significance of autoencoders in various domains, including image processing, data compression, and anomaly detection, cannot be overstated. This is due to their ability to capture essential features and patterns within the input data, thus enabling efficient and effective processing and analysis.

2.4.3 Hidden Markov Model (HMM)

HMMs are machine learning models designed for modeling sequential data. They operate on the assumption that a system follows an underlying Markov process with unobservable (hidden) states influencing observable outcomes. Each hidden state generates an observable outcome in an HMM, forming a sequence of observations over time. The transitions between hidden states are governed by probabilities, allowing HMMs to capture the temporal dependencies inherent in sequential data.

HMMs find applications in diverse fields such as speech recognition, natural language processing, and bioinformatics [20]. Their ability to model dynamic systems makes them well-suited for scenarios where understanding the underlying states and transitions in sequential data is crucial. Training an HMM involves estimating parameters from observed data, enabling the model to learn the probabilities associated with state transitions and emissions. Despite their simplicity, HMMs offer a robust framework for capturing complex dependencies in time-series data.

2.4.4 K-Nearest Neighbors (KNN)

In classification scenarios, KNN utilizes the votes of its K nearest neighbors to assign a class to a new data point. The choice of K impacts the algorithm's accuracy, with larger K values enhancing accuracy but potentially blurring class boundaries [21].

However, due to memory requirements and complexity, KNN's implementation can be computationally expensive for larger datasets. The challenge lies in determining an optimal K value, influencing the algorithm's efficiency during prediction [22].

As a supervised technique, KNN's training involves pairs of feature vectors and corresponding class labels. Unlabelled feature vectors receive labels based on the majority vote from the K nearest neighbors during classification. The algorithm's structured sequence includes training, initializing K, computing distances, selecting K smallest distances, obtaining corresponding class labels, and finally, classifying the new data. This succinct overview highlights the simplicity and adaptability of KNN in various ML applications.

2.4.5 Support Vector Machine (SVM)

SVMs are supervised learning systems versatile in regression and classification tasks. Utilizing linear functions in a high-dimensional feature space through optimization theory learning [23], SVMs aim to find an N-dimensional hyperplane (N being the number of features) for clear data classification.

In SVM's linear classification technique, a hyperplane distinguishes classes based on the number of inputs, a weight vector, and a bias factor [24]. The objective is to position the hyperplane far from support vectors, optimizing accuracy.

Despite its accuracy and robustness, especially for numerous classes, SVM can have high computational complexity during training [25]. To address non-linear separability in some datasets, Kernel functions are introduced. These functions elevate data to a higher-dimensional space, aiding in finding a linear separation and defining an optimal hyperplane. Kernel functions enhance SVM's adaptability across diverse datasets.

2.5 Introduction to Performance Metrics

This section presents and analyzes the performance metrics used to evaluate the classifiers. Understanding these metrics is crucial for interpreting the effectiveness of the models in accurately classifying the data. The key metrics considered are accuracy, precision, recall, and F1-score, along with the confusion matrix, which provides a comprehensive overview of the classification performance.

Accuracy

Accuracy measures the model's ability to classify the instances correctly. It is the ratio of correctly predicted cases to the total cases. Although accuracy is a straightforward and commonly used metric, it may not always be the best indicator of model performance, especially in cases where the data is imbalanced. Accuracy is defined as follows

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}.$$

Precision

Precision, also known as the positive predictive value, measures the proportion of true positive predictions among all positive predictions. It indicates the accuracy of the model when it predicts the positive class. High precision means that the model has a low false positive rate. The precision metric is defined as follows

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}.$$

Recall

Recall, or sensitivity, measures the proportion of true positive predictions among all actual positive instances. It indicates how well the model can identify all relevant instances. High recall means that the model has a low false negative rate. Recall is defined as

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}.$$

F1-Score

The F1-score is the harmonic mean of precision and recall, balancing the two. It is a useful metric when there is an uneven class distribution or when one is interested in the trade-off between precision and recall. The F1-score is defined as

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Confusion Matrix

The confusion matrix is a table that provides a detailed breakdown of the model's predictions compared to the actual outcomes. It shows the number of true positive, false positive, true negative, and false negative predictions, giving insight into the errors the model makes. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. This allows for a comprehensive evaluation of the classification performance across all classes.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

By analyzing these metrics and the confusion matrix, we can gain a deeper understanding of the strengths and weaknesses of the neural network model in classifying data, allowing us to make informed decisions about model improvements and adjustments.

2.6 State-of-the-art in Passive RF Sensing

The research presented in [26] proposed a system capable of estimating and tracking human poses in motion. The system utilized a 3D skeleton as a foundation for pose estimation. The process involved capturing the signal from a Wi-Fi router reflected on the human body and received on two non-linear spaced antennas. Subsequently, 2D AoA values were exploited to identify the human body. A deep learning model based on CNN was then employed to extract dynamics such as limbs and torso. Finally, the Long-Short Term Memory (LSTM) model was used to model the temporal dynamics of the human pose, such as the trajectory of human limbs and torso. The system was thoroughly tested in various scenarios and demonstrated an overall localization error of only 4.5 cm.

A real-time system for human body motion sensing, with a particular emphasis on body localization and fall detection, was proposed in [1]. The system continuously monitors and processes RF signals emitted by radio devices operating in the 2.4GHz ISM band. Indoor diffraction and multipath phenomena that affect the RF signal are used for body localization. The system employs the HMM to identify different human postures and detect safety-relevant events by tracking the RSSI footprints. This approach achieved an impressive 97% accuracy in fall detection.

In [2], a human body motion sensing system was proposed to detect activities such as crawling, lying, standing, walking, or even out of the scene in real time. The system was an active and passive non-adoc Device Free radio-based activity recognition (DFAR) system. The passive system used RSSI from an ambient FM radio signal at 82.5MHz. The system used a pool of features, and out of them, three features provided higher accuracies: variance, the third momentum, and the minimum values over a window of max values. With these features, two classification methods were applied, namely KNN and classification three. The system obtained an f1-score of 0.809 in terms of accuracy.

In [27], a multitarget human motion detection system utilizing commercial Wi-Fi infrastructure was proposed. The study investigated the effect of intense human movements on CSI in different scenarios, including Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS), to enhance system performance. Features were extracted from the CSI amplitude and phase difference, Standard Deviation (STD), Median Absolute Deviation (MAD), Interquartile Range (IQR), and Signal Entropy. These features were utilized to train a One-Class Support Vector Machine (OSVM) classifier, which resulted in a detection accuracy of 90% in identifying intense human activity.

In [28], a touchless system was developed that enabled human interaction with interfaces from a distance. The system incorporated a 2D array of RF sensors that were specifically designed to detect the proximity of the human body. The system was able to achieve an accuracy rate of 75% in detecting the distance and position of human hands when located at a distance of up to 2 inches.

In the study presented in [29], a novel approach for recognizing and classifying fundamental activities based on passive RF signals using deep learning was proposed.

The proposed system utilized two channels, namely reference and surveillance, wherein adaptative filtering was applied to eliminate the echo from the surveillance channels. Subsequently, a CNN was employed to classify the motion based on the time series signal. The offline classification accuracy of the proposed system was 90%, while the online classification accuracy was 70%.

In [3], a Radio-Frequency Identification (RFID) system for human motion recognition was proposed. The system was designed to operate in complex multipath environments, while being resilient to changes in the environment. In order to address the discontinuity issues in the RFID communication domain, a technique called Data Slicing was employed. Subsequently, a Synthetic Aperture Radar (SAR) algorithm was utilized to generate the fingerprint matrix corresponding to each motion. Finally, Dynamic Time Warping (DTW) was employed to match the motion fingerprint with the pre-existing database of fingerprints. The proposed system achieved an accuracy rate of 90%, which was reduced to 87% in the presence of environmental variations.

In a study presented in [30], researchers utilized the CSI obtained from ambient RF signals to detect the presence of humans. They encountered challenges such as the impact of collecting data in the presence of humans, and performing DL techniques on complex-valued signals. The researchers used both the phase and magnitude information to train a CNN. They found that the developed system was highly robust in different environments and was even more sensitive to human presence than Passive Infrared (PIR) sensors.

A study in [31] conducted a device-free system for fall detection which used CSI to identify activities. The system was evaluated in three different indoor scenarios using devices equipped with 802.11n Network Interface Controller (NIC). The study concluded that by using an SVM classifier, the system achieved a detection precision of 90%. However, if the Random Forest algorithm was used instead, the accuracy increased to 94%.

The study presented in [32] proposed an enhanced KNN scheme system for indoor localisation using existing WiFi RSSI. The proposed system utilizes the Spearman rank correlation coefficient to evaluate the similarity between rankings. In this system, RSSI values are obtained from the Access Point (AP)s and are used to form fingerprints, which are then utilized to construct a radio map during the training process. The Spearman distance is calculated based on the previously calculated coefficient, and a combination with the KNN approach is performed. The experimental results demonstrated a localisation error of less than 2.7 m.

In [33], a human posture classification system operating in FM radio bands was proposed to detect postures such as sitting, standing, and being out of the scene. Datasets were created from two SDR, then features were extracted from this dataset using ANOVA as a feature selection algorithm. Three classification techniques were applied in the classification stage: the sum of distances to all clusters' points, SVM, and KNN. The SVM achieved the best performance with a f1-score of 88%. This thesis builds upon the work presented in [33], further developing the proposed system and exploring new approaches to address the same problem, with the aim of improving the accuracy and robustness of

human posture classification in FM radio bands.

3 Experimental Setup and Data Extration

This chapter describes the implementation of the passive RF-sensing system, the methodology used for dataset collection, the specific scenarios targeted for detection and classification, and provides an analysis of the datasets produced and utilized in this thesis.

3.1 Experimental Setup

The proposed RF-sensing system is composed of two identical SDR devices, two identical directional antennas, and two Low-Noise Amplifier (LNA)s. The SDR device utilized in this research is the Nuand bladeRF 2.0 micro xA4. This device supports 2x2 MIMO streaming and includes two receiving antennas (RX) and two transmitting antennas (TX). Both the Field-Programmable Gate Array (FPGA) and the USB 3.0 peripheral controller of the bladeRF are fully programmable. A comprehensive list of the SDR's specifications is provided in Table 3.1.

Table 3.1: RF Specifications of the Nuand bladeRF 2.0 micro xA4 (adapted from [34]).

Specification	Min	Value	Max	Unit
RF Frequency Range (RX)	70	-	6000	MHz
RF Frequency Range (TX)	47	-	6000	MHz
ADC/DAC Sample Rate	0.521	-	61.44	MS/s
ADC/DAC Resolution	-	12	-	bits
RF Bandwidth Filter	<0.2	-	56	MHz
CW Output Power	-	+8	-	dBm

Figure 3.1 presents the Nuand bladeRF 2.0 micro xA4 block diagram, highlighting the TX bias-tee and RX bias-tee peripherals, which offer fully software-controllable power management. In the context of the RF-sensing system under investigation, two Nuand BT-200 wideband LNAs were utilized, each dedicated to a specific receiving channel, as shown in Figure 3.2.

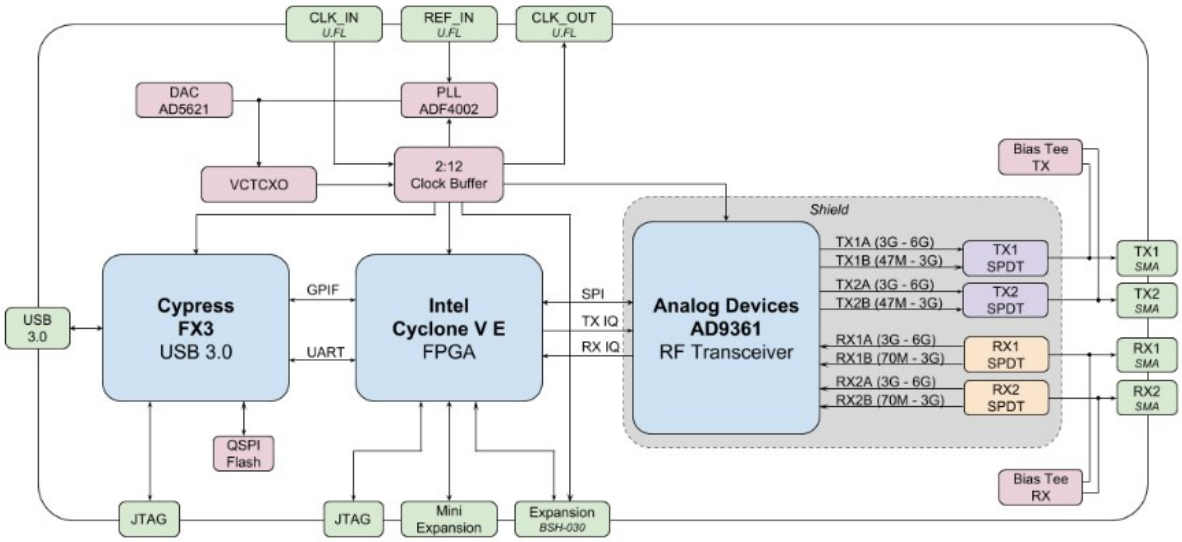

Figure 3.1: Block diagram of the Nuand bladeRF 2.0 micro xA4 (copied from [35]).

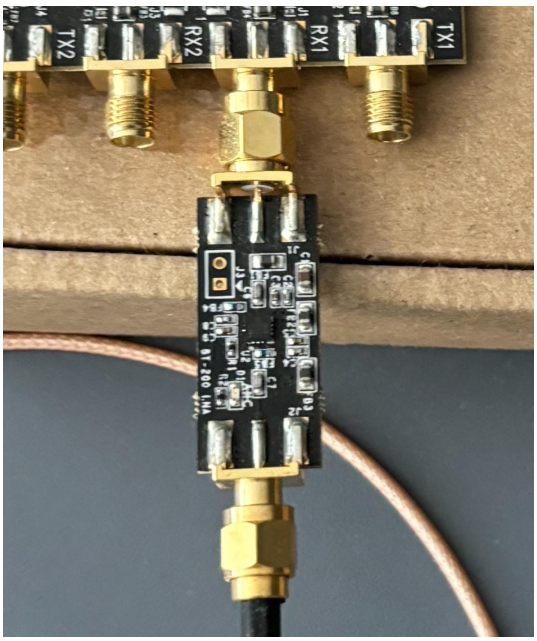

Figure 3.2: Nuand BT-200 wideband LNA.

Figure 3.3: Directional Antennas.

To mitigate potential sample losses due to USB sampling constraints, each of the two receiving (RX) chains is configured within separate SDRs. This configuration ensures that each SDR independently handles its data stream, reducing the likelihood of bottlenecks or overflows that could occur if both chains were managed by a single device. By distributing the processing load, this approach enhances the reliability of data acquisition and minimizes the risk of missing critical information during sampling.. Consequently, synchronization between the two bladeRF 2.0 micro units is essential during the sample acquisition process. This synchronization is achieved by linking the clocks of the two SDRs via a dedicated cable, as shown in Figure 3.4. Specifically, the CLK_OUT port of one device is connected to the CLK_IN port of the other. In this configuration, the clock of the first device is designated as the master clock, while the clock of the second device operates as the slave clock.

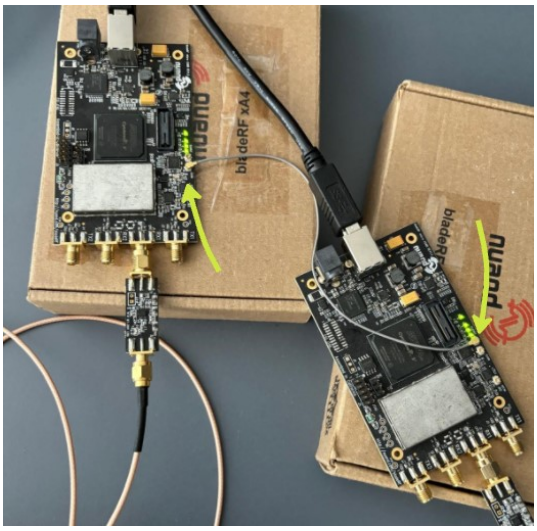

Figure 3.4: Linked Clocks.

Figure 3.5 below illustrates the passive RF-sensing system from the perspective of an individual. It is important to note that the surveillance antenna is oriented towards the individual whose posture is to be recognized, while the reference antenna is directed towards the external transmitting antenna, which provides the RF signal to be utilized. Additionally, the angle between the pointing direction of the reference antenna and the surveillance antenna is set to 90 degrees, making them orthogonal.

Figure 3.5: RF-sensing system's setup.

Figure 3.6 presents a schematic of the RF-sensing system's setup, including relevant measurements. As shown in the figure, the individual whose posture is to be classified is positioned at a horizontal distance of 78 cm from the surveillance antenna.

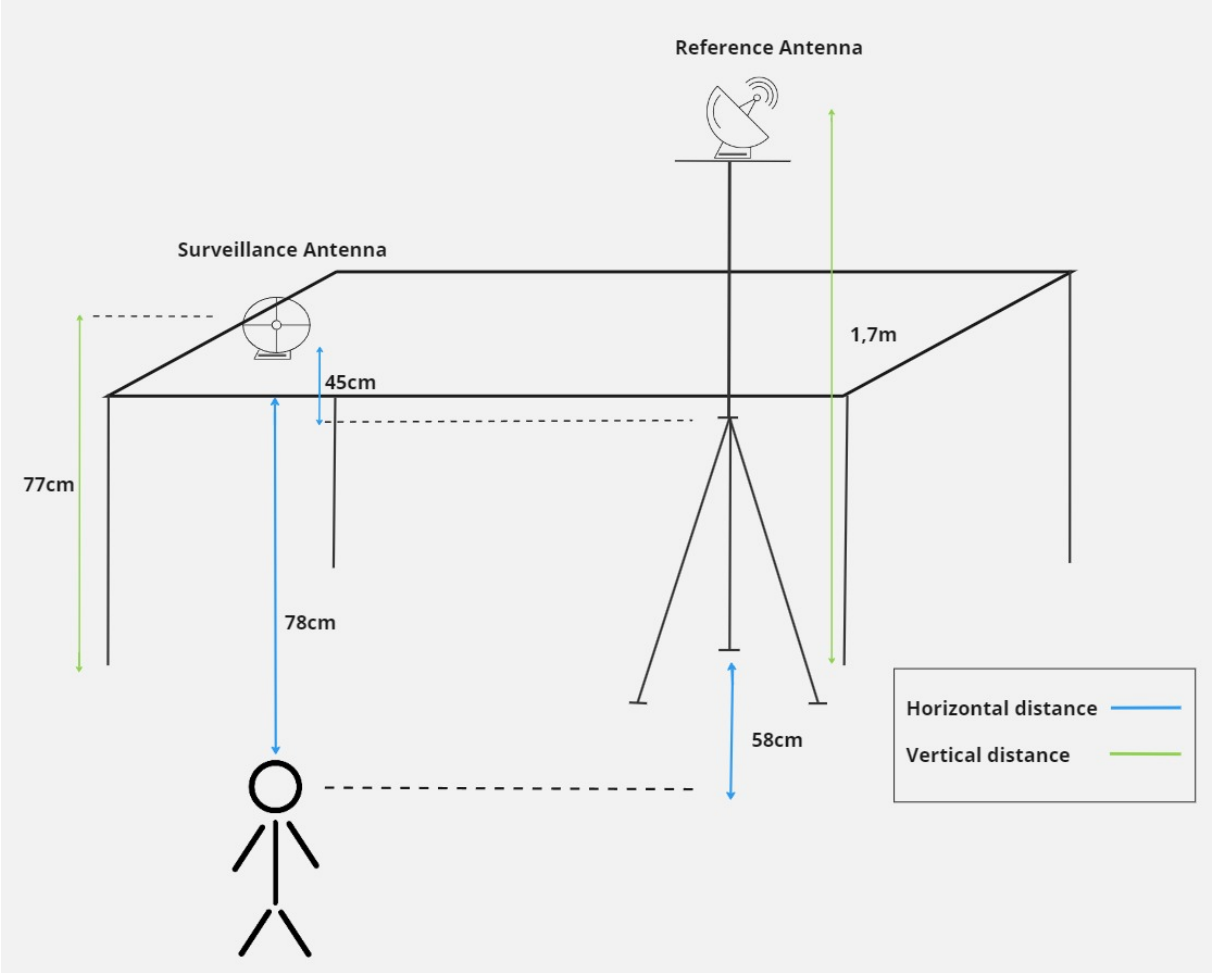

Figure 3.6: Schematic of the RF-sensing system's setup.

3.2 Data Acquisition Methodology

To facilitate the execution of various tasks, such as loading FPGA bitstreams or flashing firmware files on the Nuand bladeRF software-defined radio systems, the `bladeRF-cli` utility is employed. To avoid the need to manually load the xA4 49KLE Altera FPGA each time the bladeRF is used, an FPGA bitstream was written to the SPI flash of both devices. This process can be accomplished by executing the code in Figure 3.7 using `bladeRF-cli`. Upon booting, the firmware checks for the presence of FPGA metadata and loads the FPGA if a bitstream is detected.

```
# bladeRF command line interface and test utility (1.5.0-0.2016.06-2)
bladeRF-cli -L path/to/hostedx(size).rbf
```

Figure 3.7: FPGA load code lines.

As previously mentioned, the clocks of the two bladeRF 2.0 micro devices must first

be synchronized. This synchronization ensures precise time alignment between the reference samples (data collected by the reference SDR) and the surveillance samples (data collected by the surveillance SDR), which is essential for accurate sample processing. This synchronization can be achieved by executing the following code lines within the scriptable interactive mode of the `bladeRF-cli` utility, as shown in Figure 3.8.

```
#bladeRF command line interface and test utility (1.5.0-0.2016.06-2)
bladeRF-cli-i
#open reference bladeRF usb port
open libusb: instance=0
#set reference bladeRF's clock as master
set clock_out enable
#open surveillance bladeRF usb port
open libusb: instance=1
#set surveillance bladeRF's clock as slave
set clock_sel external
```

Figure 3.8: Synchronisation code lines.

The bladeRF 2.0 micro integrates seamlessly with various open-source software development tools, including GNU Radio [36], GQRX [37], and others, through the `libbladeRF` library.

The GNU Radio

GNU Radio is an open-source software developed to analyze, design, and implement SDR systems. This toolkit is implemented in both C++ and Python; the core framework of the toolkit is written in C++, and several user tools are developed in Python. While this work was being developed, GNU Radio was selected to implement the SDR system because of the remarkable ability of this toolkit in developing Graphical User Interface (GUI)s.

GNU Radio applications, also known as flowgraphs, comprise several signal-processing blocks. These blocks help flow data streams by transferring data from the input ports to the output ports, making it easy to process data. GNU Radio provides a host of example blocks and elements for the user to employ, including filters, equalizers, decoders, demodulators, sinks and sources. If no specific block is available, users can produce their block and place it in the block library.

In this thesis, the GNU Radio Companion (GRC) was chosen to be the GUI tool. GRC only works with Python code and can be used with any other Python integrated GUI framework. In addition, GRC can also convert a flowgraph into a Python script which is then compiled. This aspect is beneficial since the generated Python script will be executable from the terminal.

The packages needed for running GNU Radio applications were then compiled with the help of PyBOMBS [38], a package manager and builder for GNU Radio. Like most GNU Radio source codes, PyBOMBS is not very system-specific and is tasked with downloading,

compiling, and installing GNU Radio. During this thesis, GNU Radio version 3. 8 was utilized.

3.3 Data Characterisation

The initial stage of the data acquisition process involved the selection of an appropriate illuminator, which refers to the RF transmitting source. Following a comprehensive evaluation, the M80 Manteigas radio station, an FM radio broadcasting station owned by Beiras FM, was chosen for this purpose. Located in Lisbon (postal code 1099-049), the station operates at a frequency of 104.3 MHz with a Photosynthetically Active Radiation (PAR) value of 500 W. As indicated in Table 3.2, this transmission frequency falls within the VHF range of the RF spectrum. Notably, this particular broadcast band was preferred over alternative options, such as Wi-Fi communications, due to its characteristic of continuous signal transmission.

Table 3.2: Radio frequency spectrum bands.

Band Name	Frequencies	Wavelengths
Extremely low frequency (ELF)	3-30 Hz	100.000-10.00 Km
Super low frequency (SLF)	30-300 Hz	10.000-1.00 Km
Ultra low frequency (ULF)	300-3000 Hz	1.000-100 Km
Very low frequency (VLF)	3-30 KHz	100-10 Km
Low frequency (LF)	30-300 KHz	10-1 Km
Medium frequency (MF)	300-3.000 KHz	1.000-100 m
High frequency (HF)	3-30 MHz	100-10 m
Very high frequency (VHF)	30-300 MHz	10-1 m
Ultra high frequency (UHF)	300-3.000 MHz	100-10 cm
Super high frequency (SHF)	3-30 GHz	10-1 cm
Extremely high frequency (EHF)	30-300 GHz	10-1 mm
Terahertz high frequency (THF)	300-3.000 GHz	1-0,1 mm

After the selection of the illuminator, the next process is the data extraction, which is done in two phases. First, to streamline the research's first steps, we look at the M80 station's frequency range in its entirety. This approach enables us to embrace a wide range and get a picture of the M80 in as much detail as possible, which is crucial to address the research objectives. Next, our attention is focused on the frequency range connected with the operation of the stereo pilot. Here, we will determine if the data within this range can give adequate information to classify.

Regarding the classification scenarios detailed in Table 3.3, the individual will assume the poses according to the distances specified in Figure 3.6. Each pose corresponds to a distinct class and is described as follows:

- **Class 0:** The individual stands in front of a chair, facing the surveillance antenna.

Table 3.3: Class and Posture Description.

Class	Posture Description
0	Standing in front of a chair
1	Sitting on a chair
2	None

The arms should be positioned alongside the legs, which must be aligned and close together.

- **Class 1:** The individual sits on a chair, also facing the surveillance antenna. The arms should rest on the legs with palms facing downward, and the legs should remain close together.
- **Class 2:** This class indicates the absence of a person in front of the system, thus characterizing the environment. In this scenario, the individual should be approximately 170 cm away from the chair and remain stationary, as depicted in Figure 3.9.

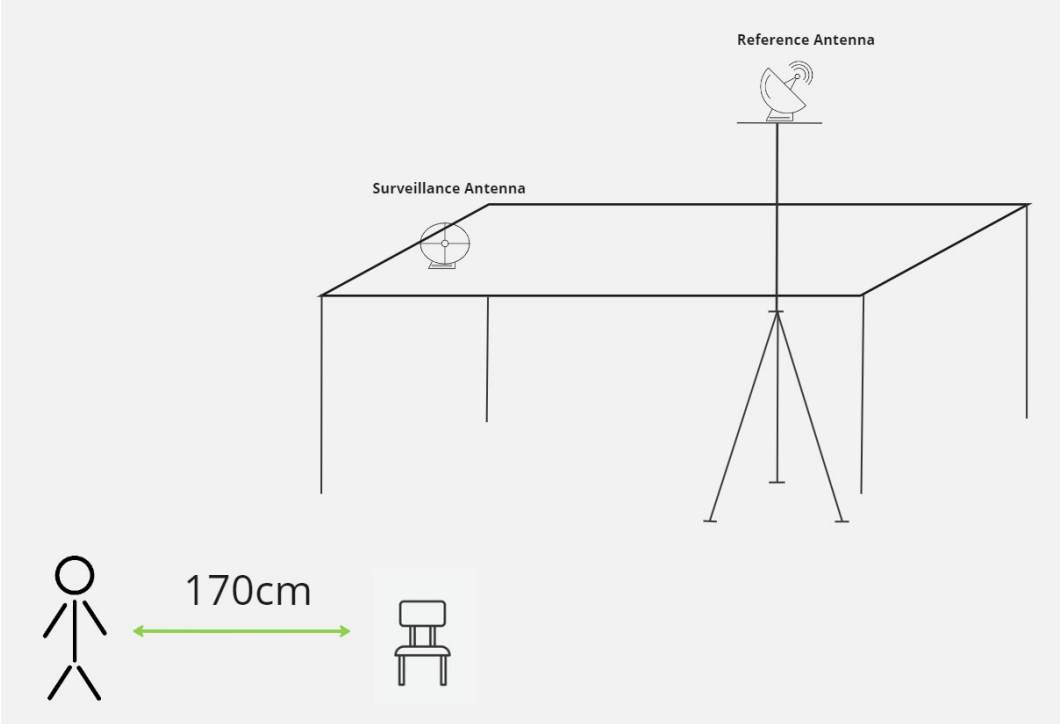

Figure 3.9: Schematic of a person performing Class 2.

The typical spectrum of a composite baseband FM signal comprises several key components, as illustrated in Figure 3.10. On the far left, the mono audio component represents the sum of the audio signals from both the left and right channels. Next, the stereo audio signals are encoded as the differences between these channels, allowing for stereo

sound reproduction. The 19 kHz stereo pilot tone is a crucial element that helps receivers identify whether the transmission is in stereo mode. Additionally, the Radio Broadcast Data System (RBDS) carries supplementary information, such as the station name and music genre. Other subcarriers, including DirectBand and audio subcarriers, may be present to transmit further data. Together, these components create a composite FM signal that delivers both audio and additional information to listeners.

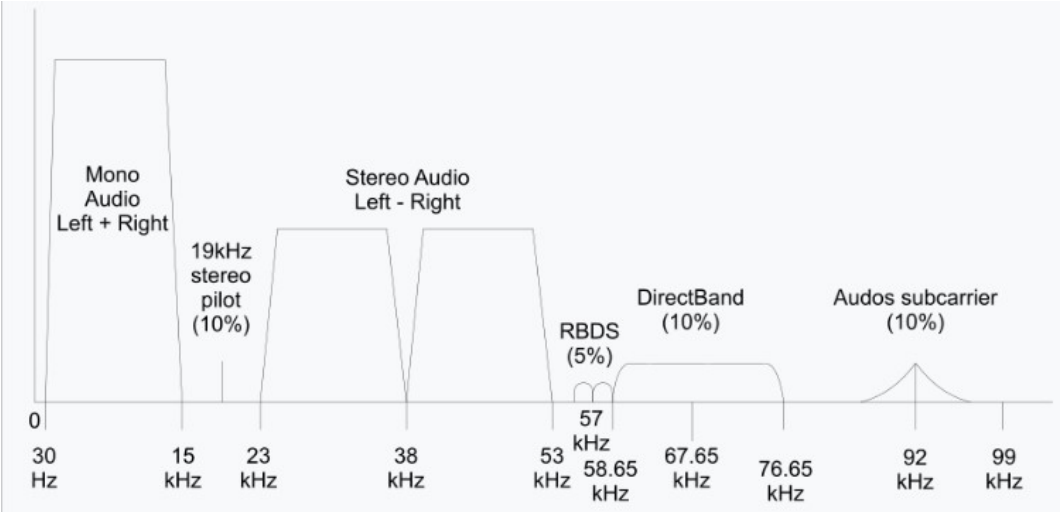

Figure 3.10: Typical spectrum of composite baseband FM signal.

The data for both approaches will be collected using Python scripts to generate two distinct flowgraphs. Each approach will use different parts of the spectrum as illuminators.

Approach 1 will utilize the entire FM signal bandwidth allocated for a single radio station. Figure 3.11 illustrates the flowgraph for Approach 1, which captures the full spectrum of the FM signal dedicated to a single radio station. This approach provides a comprehensive view of the FM signal by encompassing all frequency components within the FM band.

Approach 2 will focus on the 19 kHz frequency band, corresponding to the stereo pilot tone in FM transmissions. The flowgraph for Approach 2, shown in Figure 3.12, is designed to isolate and analyze this narrow frequency range. This targeted approach allows for a detailed examination of the stereo pilot tone while excluding other parts of the FM signal.

Each flowgraph will be explained in detail, highlighting how it captures and processes the relevant frequency bands for its respective approach.

The flowgraph for Approach 1 shown in Figure 3.11 is used to process the entire bandwidth of the FM signal.

The first two blocks in this flowgraph are entitled "Osmocom Source" and represent the source blocks, symbolizing the two RX channels and the two bladeRF 2.0 microdevices. For each source block, the center frequency was set to the broadcast frequency of the radio station, which was 104.3 MHz. The sample rate was set to 4 MegaSamples/second (MS/s),

while the bias-tee gain was set at 45 dB.

The analog-to-digital converter (ADC)s within these blocks capture the In-phase and Quadrature (I/Q) Components from each RX channel. After that, the samples undergo two preliminary pre-processing blocks: "Skip Head" and "Remove DC Spike."

The "Skip Head" block removes the first N samples received from the source block, where N is the number of samples that corresponds to 10 µs of data acquisition. This quantity is chosen to minimize the possible initial deviation from each bladeRF device. Moreover, the "Remove DC Spike" block uses an Infinite Impulse Response (IIR) filter to remove the center frequency I/Q DC spike, which further cleans the signal.

Afterward, a Low Pass Filter block is applied to the data. The filter uses a Hamming window to pass frequencies below the selected cutoff frequency of 100 kHz and attenuate those higher than this threshold, with the cutoff frequency set to 100 kHz because the spectrum extends only up to 99 kHz. The stop-band and pass-band transition width is 10 kHz, the gain is 2 dB, and the decimation rate is 16. The decimation rate reduces the sample rate to 250 kS/s (4 MS/s divided by 16). The purpose of this filter is to constrain the data within the bandwidth of the baseband FM signal, which is illustrated with all the spectrum components in Figure 3.10.

The final block of the flowgraph is used to write a stream to a binary file. The "File Sink" block generates binary files from the N items that come from the previous block. The generated binary file is composed of 8-byte I/Q pairs, meaning that the file contains a series of `float32` values in the IQIQIQ order (i.e., 4 bytes for the real part of the complex number and 4 bytes for the imaginary part). Consequently, two binary files are produced—one for the reference antenna and another for the surveillance antenna.

Approach 2 modifies the flowgraph present in Approach 1 to extract the stereo pilot frequency as shown in Figure 3.12. This will be further modified by the addition of several key blocks.

A "WBFM Receiver block" is added to the model. The quadrature rate is set to 250 kS/s, and audio decimation is set to 1. This block demodulates the FM signal and produces a baseband signal. The quadrature rate sets the sample rate for the demodulated output, and audio decimation determines how much to downsample the audio signal. After that, a Band Pass Filter is inserted along the flow and set with a decimation of 2, gain equal to 10, and sample rate of 250 kS/s. The filter parameters are a low cutoff of 18.9 kHz, a high cutoff of 19.1 kHz, a transition width of 20 kHz, a Hamming window function, and a beta of 6.76. This filter isolates the stereo pilot frequency by allowing only the frequencies within this narrow band to pass through while attenuating frequencies outside this range.

In Approach 2, the decimation for the Band Pass Filter is set to 2 to maintain equivalence with the original sample rate. The WBFM Receiver block halves the number of samples since its input is complex-valued I/Q data, which it converts into a real-valued signal, thus virtually reducing the sample rate by half. Since this sample rate has already been reduced, to restore it for the following steps of processing to 250 kS/s, the Band Pass Filter will be set to have a decimation of 2. That way, it will keep the desired sample rate while

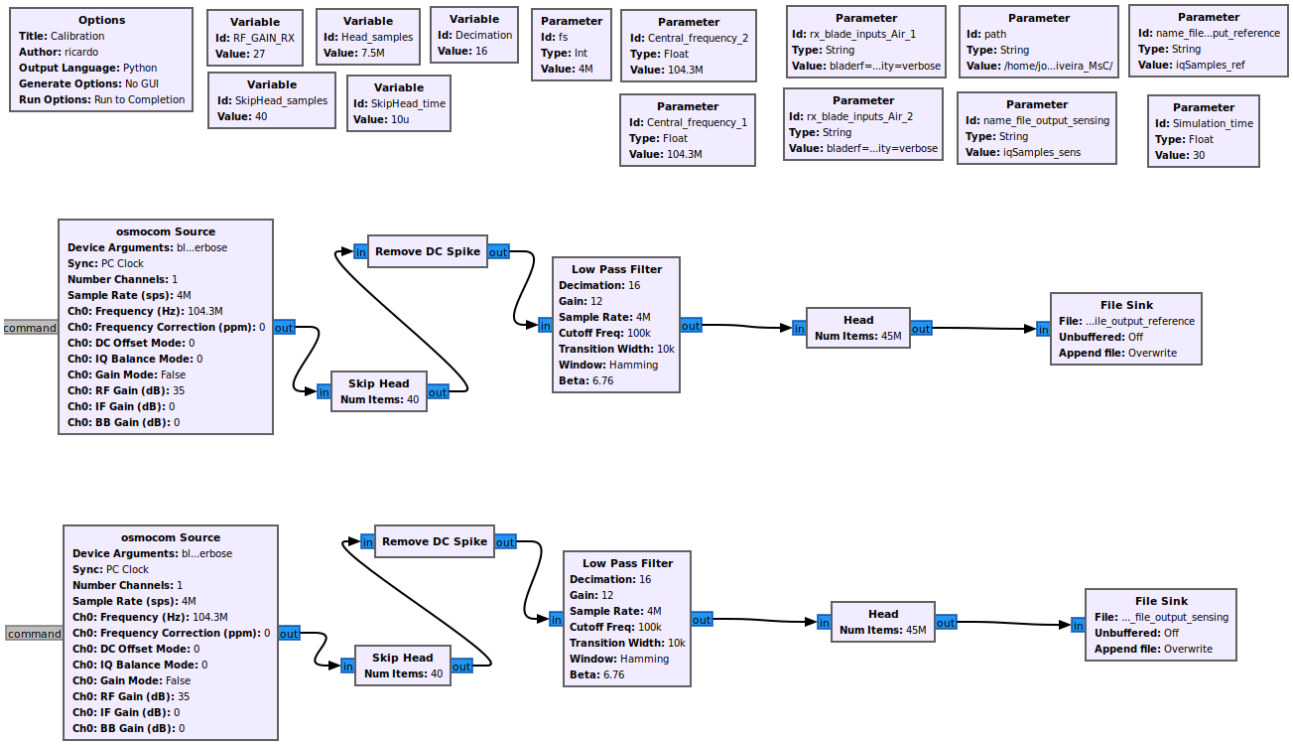

Figure 3.11: GNU Radio Companion flowgraph for offline data acquisition over the entire FM spectrum

eliminating unwanted frequencies and isolating the stereo pilot frequency.

An expanded flowgraph of Approach 2 with the additional blocks is illustrated in Figure 3.12. The final configuration acts well to extract the stereo pilot frequency from the FM signal.

3.4 Dataset Extraction and Downsampling

For each flowgraph, ten distinct datasets were extracted using the corresponding Python scripts. This approach was adopted not only to ensure sufficient data for subsequent processes but also to increase data diversity. By collecting multiple datasets, we aimed to minimize potential interference and account for any variations or fluctuations in the final FM signal. Each dataset consists of three minutes of data collected sequentially for each class (0, 1, and 2) and each antenna type (surveillance and reference). Specifically, for each dataset, we extracted 3 minutes of data from class 0, followed by 3 minutes from class 1, and finally 3 minutes from class 2, resulting in a single dataset. This process was repeated to create a total of ten datasets. Each dataset comprises six files, with each file having a size of 360 MB. Although the datasets are distinct due to their sequential extraction, they maintain the same structure and content across all ten datasets.

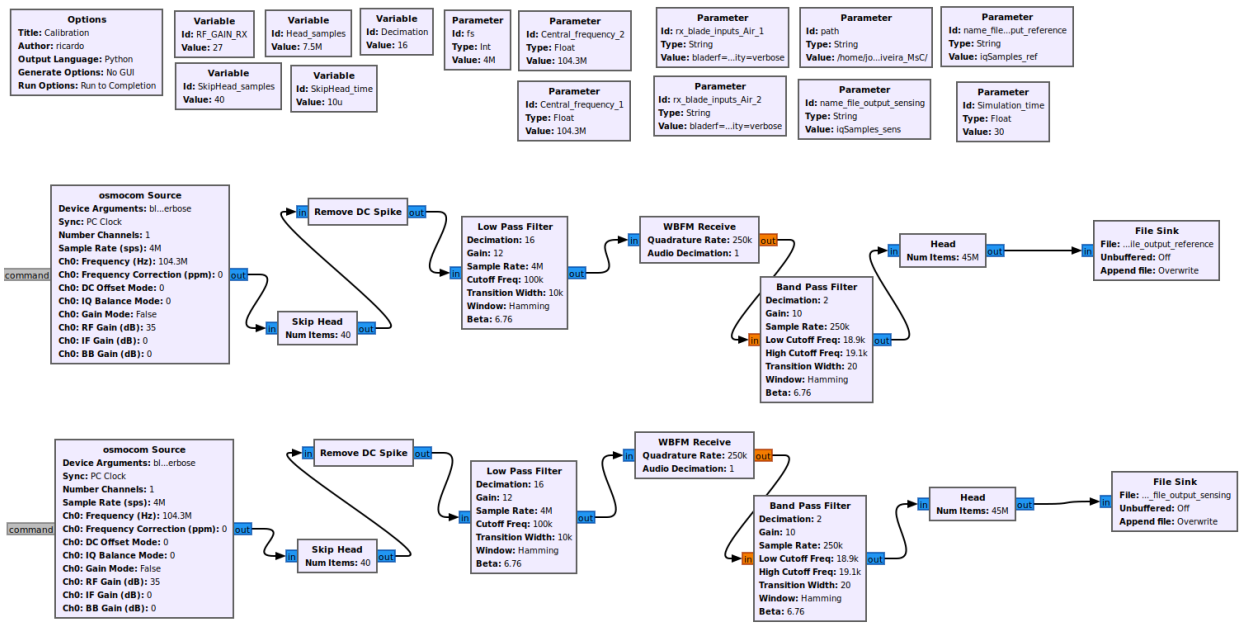

Figure 3.12: GNU Radio Companion flowgraph for offline data acquisition and isolation of the 19 KHz stereo pilot.

3.4.1 Data Size and Structure

Each 360 MB file contains 45 million samples. This is derived from a sampling rate of 250,000 samples per second over a duration of 180 seconds (3 minutes) as follows:

$$250{,}000\,\text{samples/second} \times 180\,\text{seconds} = 45{,}000{,}000\,\text{samples}.$$

After extracting all datasets from each flowgraph, they were combined into a single, unified dataset. As a result, the final dataset for each flowgraph approach consists of six files, each with a size of 3.6 GB

3.4.2 Downsampling Process

Given the substantial size of the data files (3.6 GB each, across a total of 6 files), and the need to convert this data into a more manageable format, a downsampling process was applied. This process was designed to reduce the total amount of data while preserving its essential characteristics.

3.4.2.1 Original Data Characteristics

- **Sampling Rate:** 250,000 samples per second.
- **Samples per 10 ms:**

$$250{,}000\,\text{samples/second} \times 0.01\,\text{seconds} = 2{,}500\,\text{samples}$$

- **Total Samples for Two Antennas per 10 ms:**

$$2 \times 2{,}500 = 5{,}000 \text{ samples}$$

3.4.2.2 Downsampling Procedure

- **Selection Method:** The dataset was iterated over by selecting every 5th sample, reducing the total number of data points by a factor of 5.
- **Post-Downsampling Samples:**

$$\frac{5{,}000}{5} = 1{,}000 \text{ IQ samples}$$

  Each IQ sample consists of a real part and an imaginary part. Thus, after downsampling, the dataset contains 1,000 IQ samples, each with both components.

- **Data Representation in CSV Files:** Each row in the CSV file contains 2,000 samples, representing a 10 ms window of data. This is because each IQ sample, comprising both real and imaginary parts, is stored separately. Specifically, each row is organized as follows:
  - Real part of the data from the Reference Antenna
  - Imaginary part of the data from the Reference Antenna
  - Real part of the data from the Surveillance Antenna
  - Imaginary part of the data from the Surveillance Antenna

  This organization ensures that each row contains both components for all 1,000 IQ samples, providing a complete representation of the 10 ms window.

3.4.3 CSV File Structure

Each row in the CSV file efficiently captures the downsampled data. Specifically:

- **Each CSV File Contains:** 2,000 samples per row, corresponding to 10 ms of data.
- **Total Number of Rows:** Each dataset spans 180 seconds (3 minutes) with data organized in 10 ms intervals. Therefore, the total number of rows in each CSV file is:

$$10 \text{ ms intervals per second} \times 180 \text{ seconds} \times 10 \text{ datasets} = 180{,}000 \text{ rows}$$

- **File Consolidation:** The data was organized into fewer files for simplicity. Instead of having 6 separate files per approach (one for each class and antenna combination), the datasets from both antennas for each class were combined into a single CSV file. Thus, there are 3 CSV files per approach, one for each class (0, 1, and 2).

By following this detailed downsampling and structuring approach, the data was effectively reduced in size while preserving its essential features, making it more manageable for further analysis and processing.

4 Features Analysis

In this chapter, we present a comprehensive analysis of the features used in the proposed methodology. The feature engineering process begins with the application of VAEs, using two distinct approaches. The first approach leverages the complete band of data, while the second focuses on a specific 19 kHz frequency band. For each approach, different sets of latent variables—10, 25, and 50—are tested to evaluate their influence on feature extraction and model performance. Additionally, histograms are generated for the three classes under study to assess the separability of the classes and determine whether classification is feasible based on these features.

In 4.2, we analyze statistical features derived from the dataset used in Approach 1 (complete band). These features summarize important data patterns and, similarly, histograms are created for each of the three classes to investigate the potential for classification.

In Section 4.3 focuses on phase features, derived from the datasets associated with Approach 2 (19 kHz frequency band). These features provide insights into the signal's phase information, and histograms are also generated here for the three classes to explore class separability in the phase domain.

Section 4.4 explores the application of FFT to the dataset from Approach 2, focusing on converting time-domain signals into the frequency domain. This section evaluates FFT-based feature extraction using different sample sizes—1,024, 25,000, and 45 million—to examine how the sample size impacts feature quality and computational efficiency. However, unlike the previous sections, FFT features are not explored through histograms, as the focus here is on the scalability of feature extraction. It is important to note that phase feature and FFT were tested exclusively for Approach 2, as this approach did not achieve the desired success when using the VAE.

4.1 Feature Engineering throug Variational Autoencoder

This study employed a VAE to extract features. The VAE uses a loss function that combines two components: the Mean Squared Error (MSE), which measures reconstruction loss, and the Kullback-Leibler (KL) divergence, which regularizes the latent space to follow a normal Gaussian distribution.

The VAE network architecture is fixed, consisting of an encoder with three dense layers of 2000, 500, and 100 units/neurons, followed by a latent space where the number of latent variables is varied between 10, 25, and 50. The decoder mirrors the encoder with layers of 100, 500, and 2000 units/neurons. While the overall architecture remains unchanged, the only variable component is the number of latent variables, which is adjusted to explore its effect on feature extraction.

For activation functions, Rectified Linear Unit (ReLu) is used in the intermediate layers of both the encoder and decoder, as it helps avoid the vanishing gradient problem. The final layer of the decoder uses a sigmoid activation function to ensure that output values are constrained between 0 and 1.

The VAE was trained on a dataset containing a total of 540,000 lines of data extracted from CSV files, divided equally into 180,000 lines per class for each of the three classes. This approach was applied to both the stereo pilot method and the full-band approach, as previously described. Data was normalized using the MinMaxScaler technique, which scales values between 0 and 1, prior to training.

The VAE was trained for 1,000 epochs with a batch size of 100, using the RMSProp optimizer, which adapts the learning rate based on the moving average of squared gradients. Throughout the training process, the MSE was used as the key metric for monitoring the model's performance.

After training, only the encoder component of the VAE was saved in the .h5 file format, as it is the critical part needed for subsequent feature extraction and analysis. This decision was made based on the encoder's ability to generate meaningful and distinguishable features, making it essential for future tasks involving feature extraction and classification.

To visually assess the class separability, histograms were generated for all latent variables in each case, allowing for an evaluation of the distribution and differentiation of the classes.

4.1.1 Approach 1 – Complete Band

In this approach, the VAE exhibited effective training performance, with a consistently low loss that converged over time through the use of the full band. This suggests that the model could accurately and efficiently capture the fundamental patterns in the data. Figure 4.1 presents a plot of the loss function over the training epochs, demonstrating this behaviour. It is important to note that the loss function exhibits consistent behaviour across all latent variable counts, which implies that the full band is used to achieve consistent

learning performance across various model configurations.

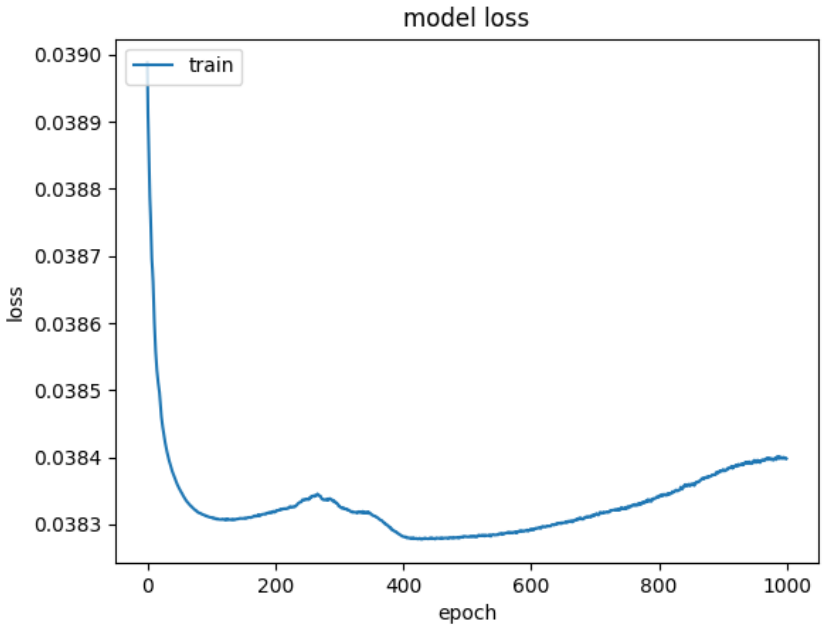

Figure 4.1: Loss curve of the VAE training process for Approach 1.

4.1.1.1 10 Latent Variables

Among the 10 extracted features, we present the histograms of the two features that showed the most significant separation between the classes (Figures 4.2 and 4.3). These features were selected based on their ability to effectively distinguish between the classes. It is possible to observe some separation between all the classes, which is a good indicator that classification through this approach could be feasible.

The remaining histograms for all 10 features are provided in Appendix A for reference.

4.1.1.2 25 Latent Variables

Among the 25 extracted features, we present the histograms of the two features that demonstrated the greatest separation between the classes (Figures 4.4 and 4.5). These features were chosen based on their strong ability to effectively differentiate the classes. It is evident that some degree of separation exists between all the classes, which suggests that classification using this approach could be promising.

The remaining histograms for all 25 features are provided in Appendix B for reference.

4.1.1.3 50 Latent Variables

Out of the 50 extracted features, we present the histograms of the two that showed the most pronounced separation between the classes (Figures 4.6 and 4.7). These features were selected for their exceptional ability to clearly distinguish between the classes. A

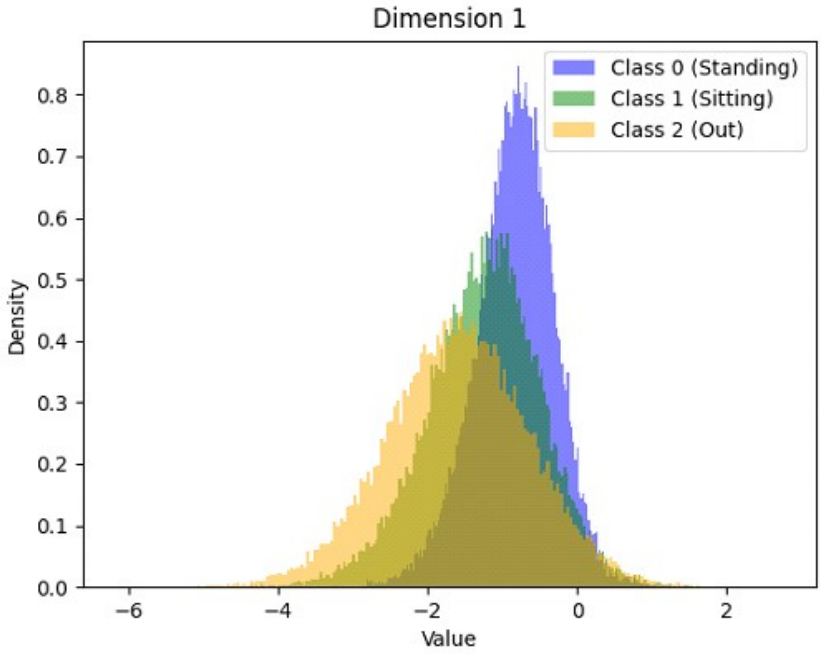

Figure 4.2: Histogram of the feature 1.

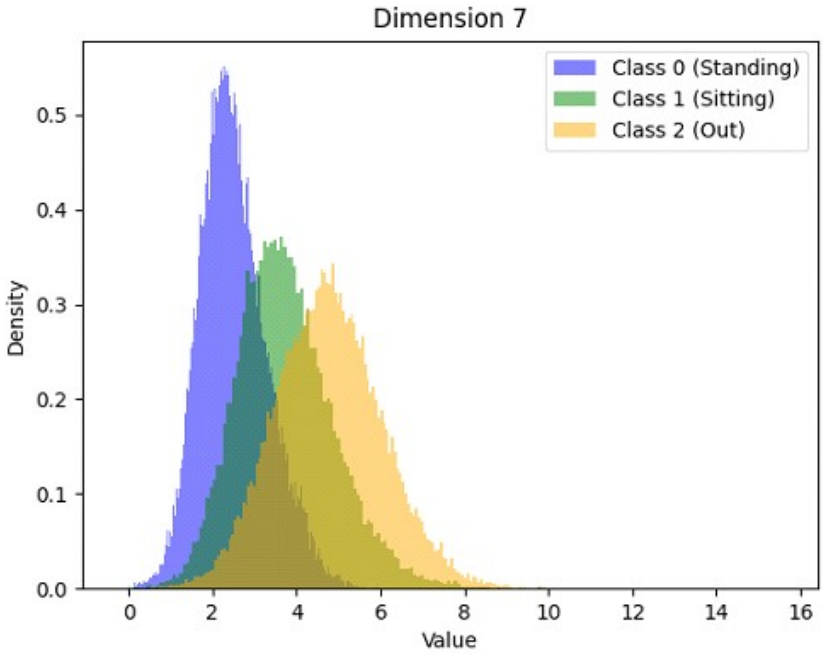

Figure 4.3: Histogram of the feature 7.

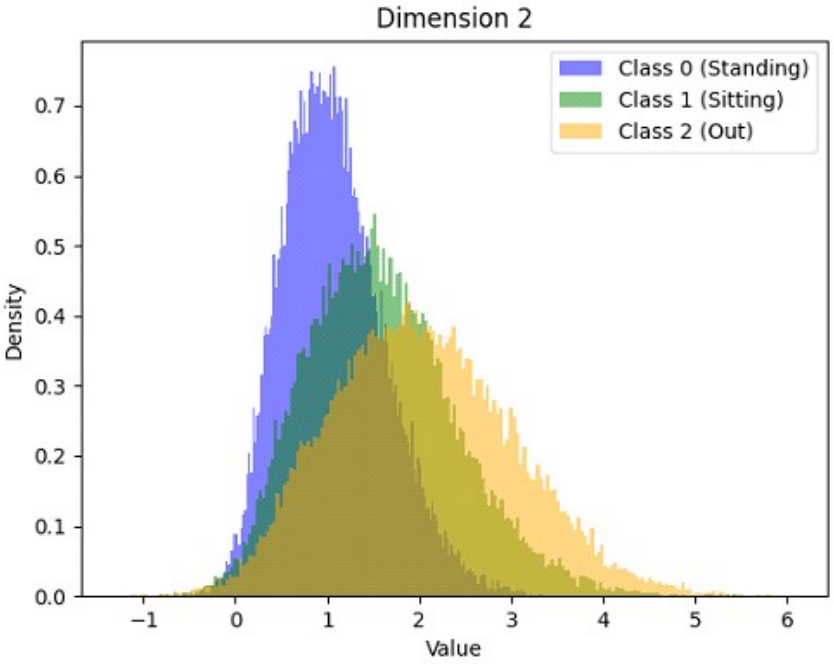

Figure 4.4: Histogram of the feature 2.

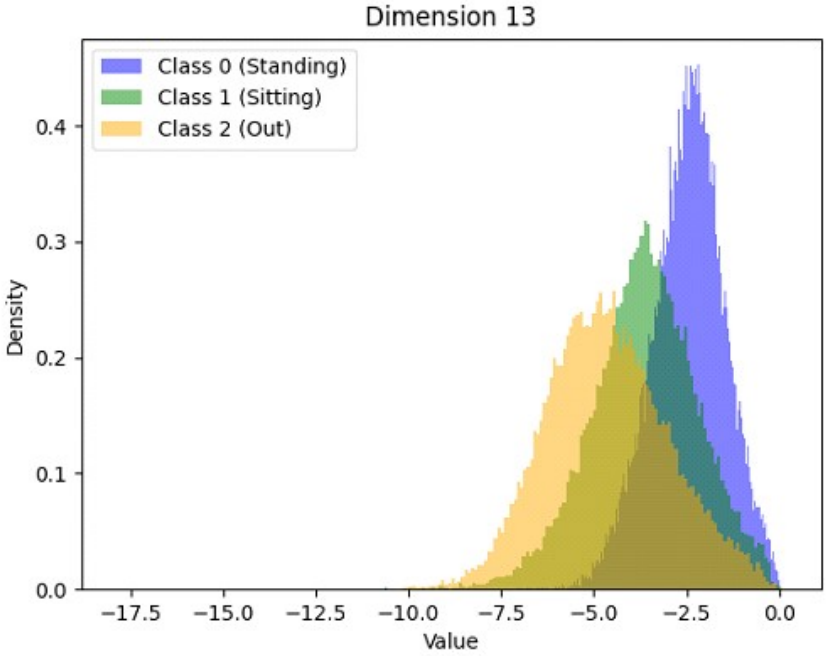

Figure 4.5: Histogram of the feature 13.

visible separation between all the classes can be observed, indicating that classification using this approach may be effective

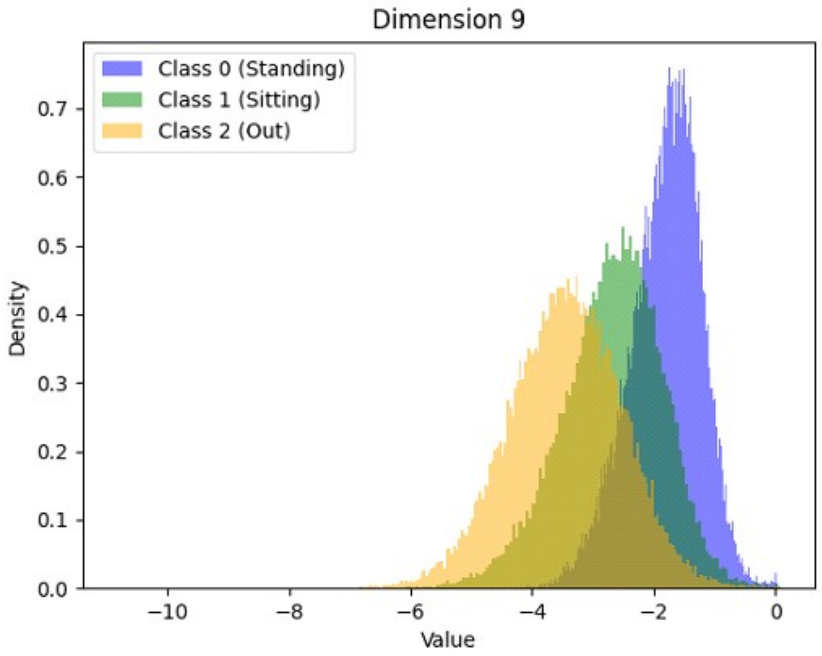

Figure 4.6: Histogram of the feature 9.

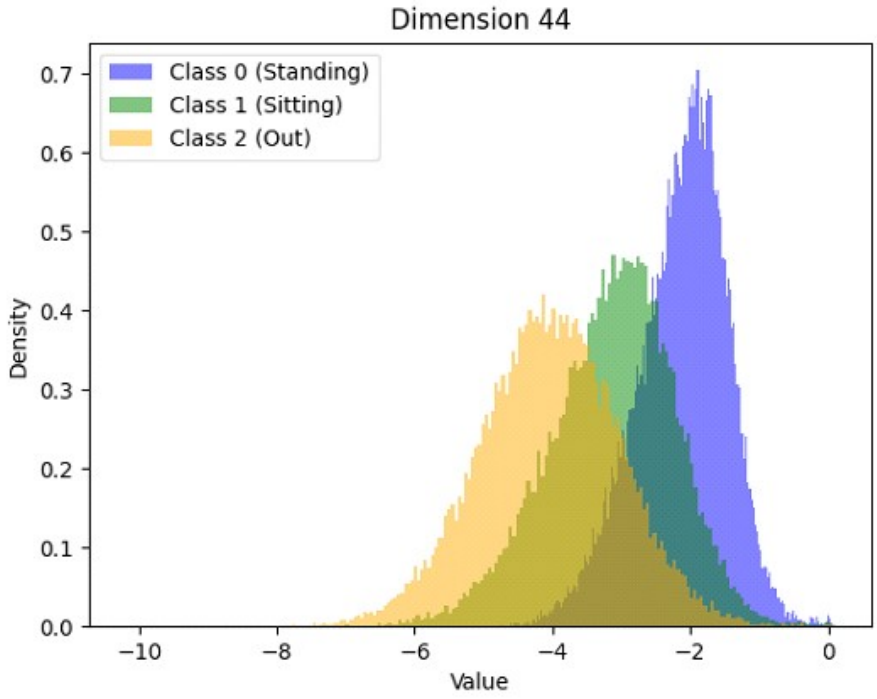

Figure 4.7: Histogram of the feature 44.

The remaining histograms for all 50 features are provided in Appendix C for reference.

4.1.2 Approach 2 – 19 kHz

In this approach, using only the stereo pilot frequency, the VAE still demonstrated effective training performance, with a consistently low loss that converged over time. This suggests that the model efficiently learned and accurately captured the underlying patterns in the data with the reduced frequency input. As shown in Figure 4.8, we present a plot of the loss function over the training epochs, illustrating this behavior. It is important to note that the loss function shows a similar pattern across all latent variables, indicating consistent learning performance across different model configurations when using only the stereo pilot frequency.

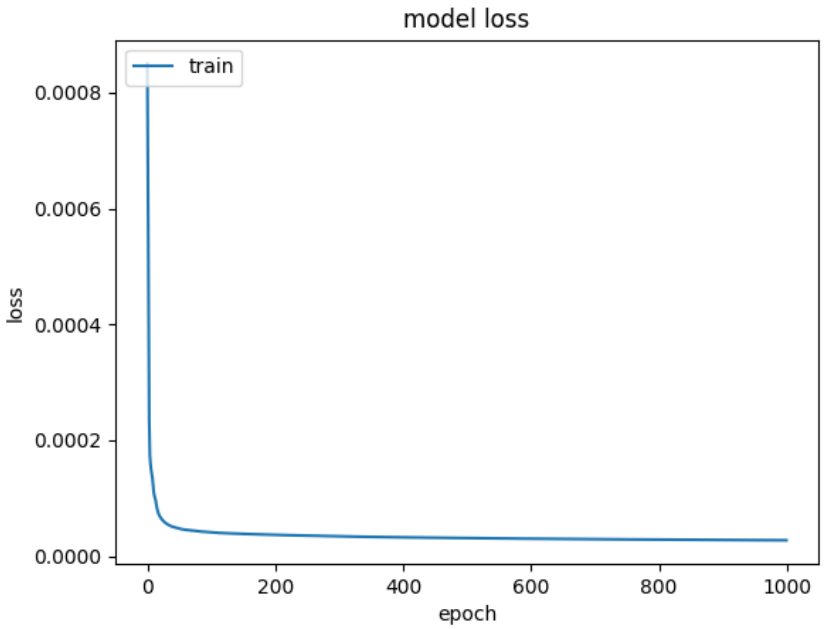

Figure 4.8: Loss curve of the VAE training process for Approach 2 – 19 kHz.

4.1.2.1 10 Latent Variables

The 10 extracted features were analyzed, and two were selected for histogram presentation (Figures 4.9 and 4.10) due to their most noticeable distinction between classes, even though the separation is not very pronounced. These features were chosen for their relatively superior ability to differentiate between the classes. However, despite being the best features, we can observe that there is almost complete overlap between the classes, suggesting that classification using this approach could be quite difficult.

The remaining histograms for all 10 latent features are provided in Appendix D for further reference.

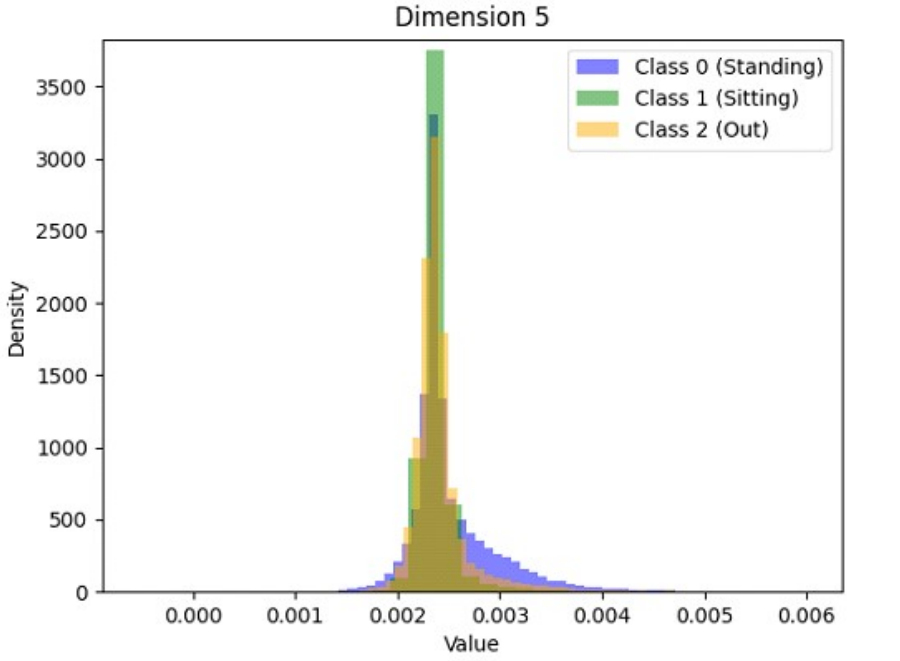

Figure 4.9: Histogram of the feature 5.

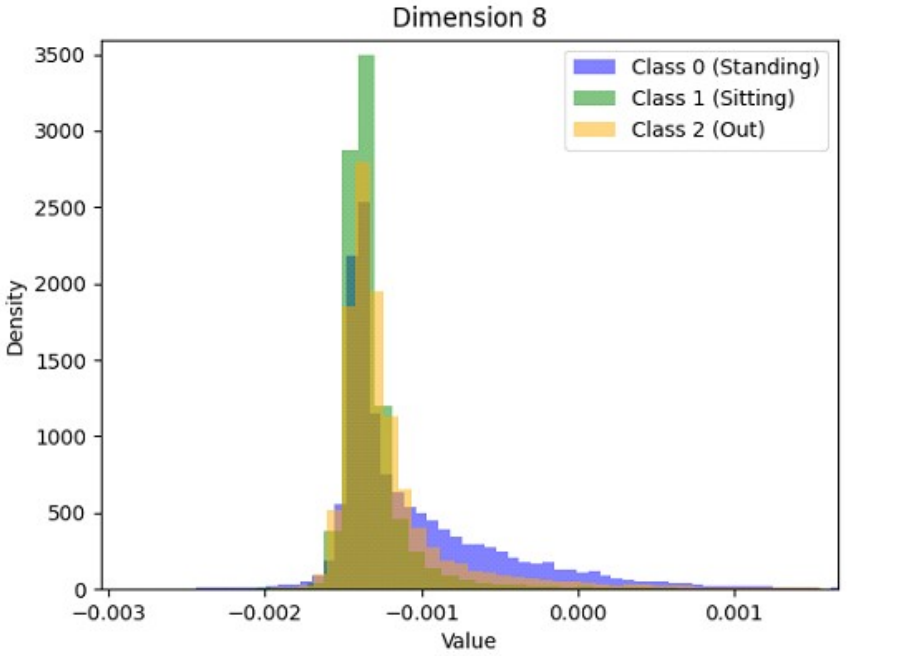

Figure 4.10: Histogram of the feature 8.

4.1.2.2 25 Latent Variables

After analyzing the 25 extracted features, two were selected for histogram presentation (Figures 4.11 and 4.12) due to their clearest distinction between classes. These features were chosen for their relatively better ability to differentiate between the classes. Nevertheless, even though these are the most distinguished features, there is considerable overlap between the classes. This suggests that using this approach for classification could be quite difficult.

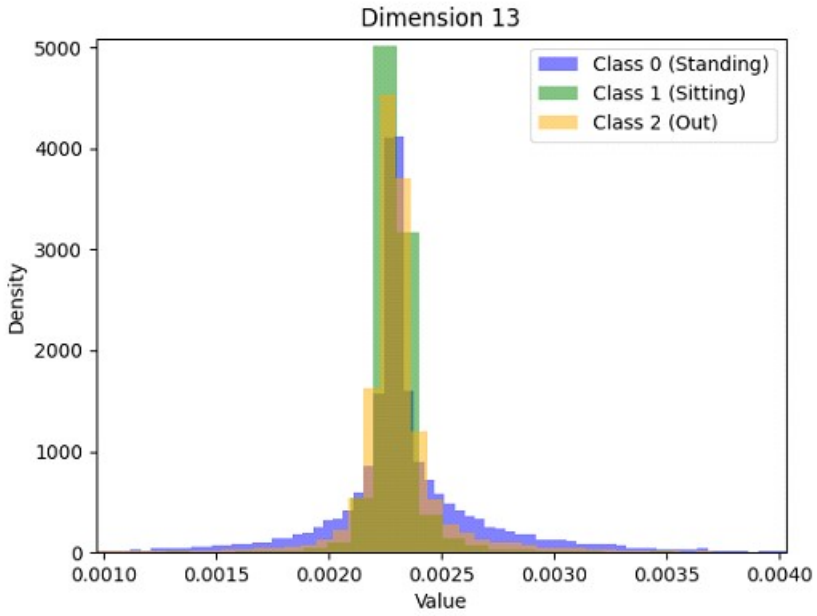

Figure 4.11: Histogram of the feature 13.

The remaining histograms for all 25 features are provided in Appendix E for reference.

4.1.2.3 50 Latent Varibles

In evaluating the 50 extracted features, two were selected for histogram presentation (Figures 4.13 and 4.14) due to their most significant distinction between classes. These features were chosen for their relatively superior ability to differentiate between the classes. Nonetheless, although these features stand out, there is significant overlap between the classes, indicating that classification using this approach could be quite difficult.

The remaining histograms for all 50 latent features are provided in Appendix F for further reference.

As we analyze the histograms, it becomes evident that, visually, the differences in class separation are minimal across the various numbers of latent variables tested. Specifically, the level of separation when using 10 latent variables is nearly indistinguishable from that observed with 25 or 50 latent variables. This suggests that varying the number of latent variables does not significantly impact the visual differentiation of the classes.

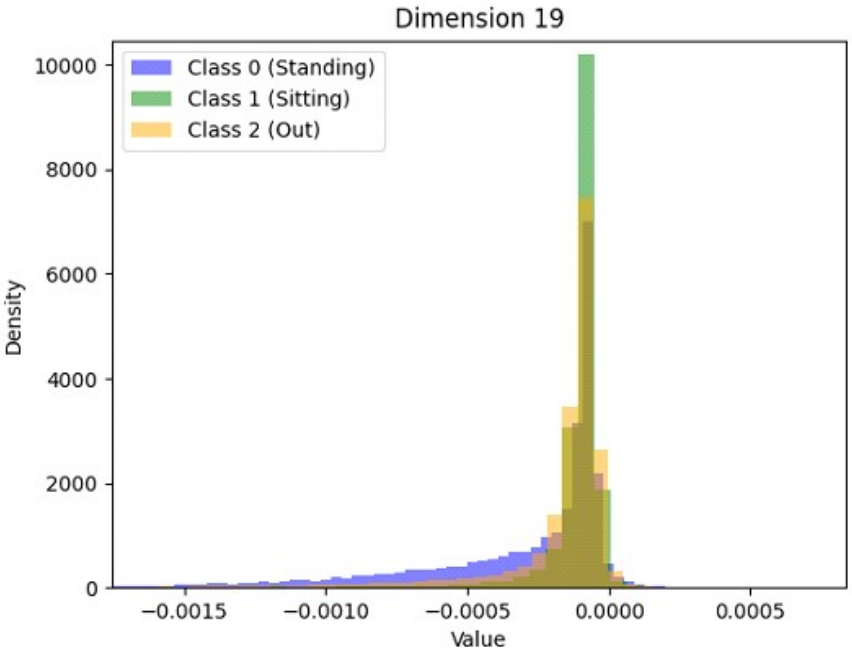

Figure 4.12: Histogram of the feature 19.

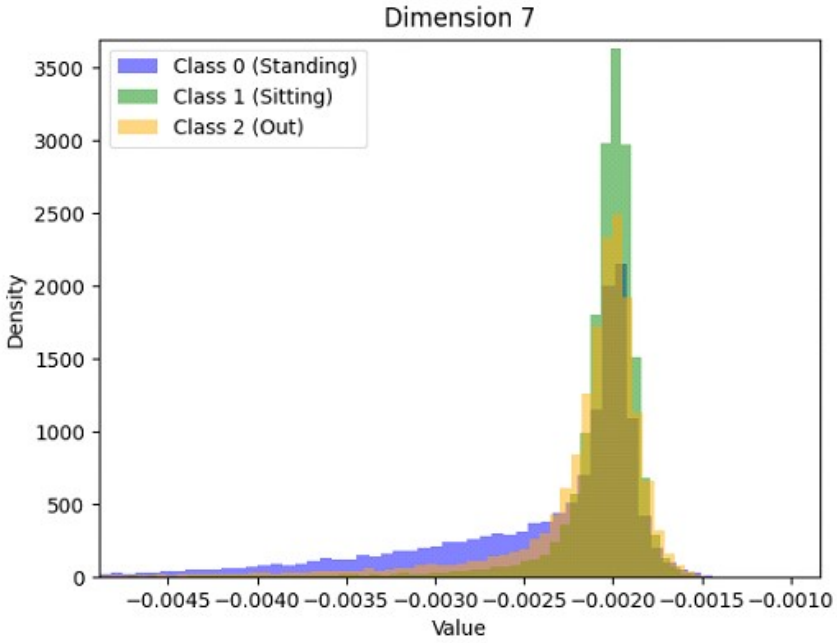

Figure 4.13: Histogram of the feature 7.

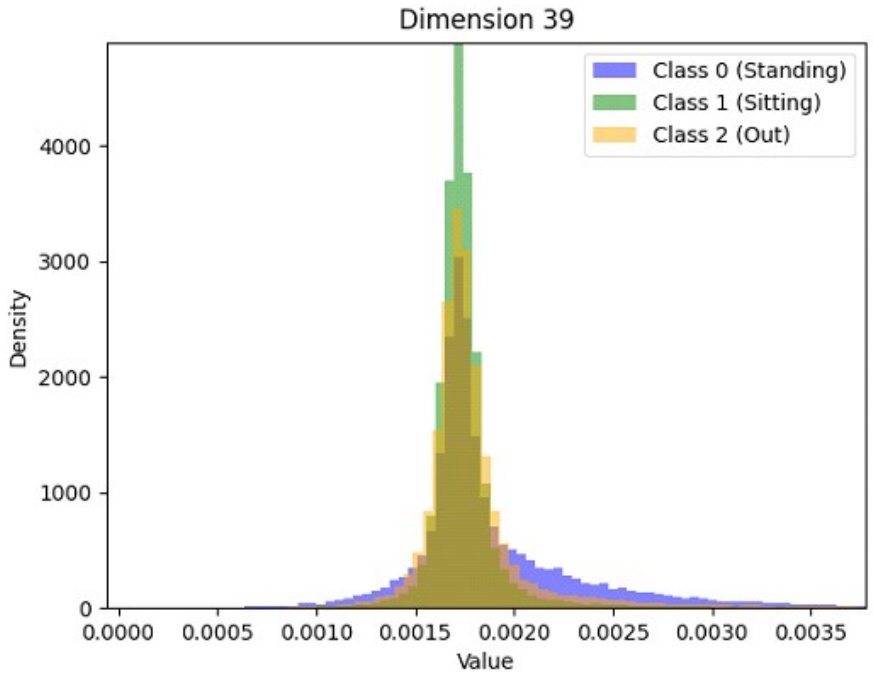

Figure 4.14: Histogram of the feature 39.

However, when we compare the results of Approach 1 with those of Approach 2, notable distinctions emerge. In Approach 1, there is some degree of separation between the classes, indicating that the features extracted in this method capture distinct characteristics that facilitate classification. Conversely, Approach 2 demonstrates considerable overlap between the classes, suggesting that the extracted features are not sufficiently discriminative to effectively differentiate the classes.

It is important to note that these observations are based solely on visual analysis. From the histograms, we can already infer that Approach 1 is likely to be significantly better for classification compared to Approach 2.

4.2 Statistical Features

In this subsection, we will generate histograms to examine the statistical characteristics of the dataset and evaluate the distinctions between classes. We will exclusively utilize the dataset from Approach 1 for this analysis.

Both approaches utilize a VAE for feature extraction. However, Approach 1 allows us to observe some degree of separation between the classes, whereas Approach 2 does not provide sufficient class distinctions. Despite the VAE being effective at identifying features, it failed to achieve clear separation in Approach 2. Therefore, we will not use the dataset from Approach 2 for our statistical feature analysis, as it does not offer the necessary clarity for distinguishing between classes.

In this analysis, we utilized the following statistical features:

1. Skewness Reference Antenna
2. Skewness Surveillance Antenna
3. Max Absolute Reference Antenna
4. Max Absolute Surveillance Antenna
5. Max Absolute Difference between Antennas
6. Standard deviation (Real Part) Reference Antenna
7. Standard deviation (Real Part) Surveillance Antenna
8. Standard deviation Absolute Reference Antenna
9. Standard deviation Absolute Surveillance Antenna
10. Standard deviation Absolute Difference between Antennas
11. Median Absolute deviation Reference Antenna
12. Median Absolute deviation Surveillance Antenna
13. Mean Reference Antenna
14. Mean Surveillance Antenna
15. Mean Difference between Antennas

Among these features, we present histograms for the two that exhibited the most significant separation between classes, which are shown in Figures 4.15 and 4.16. These histograms illustrate the features with the clearest distinction in class separation. The histograms for all other features are provided in Appendix G.

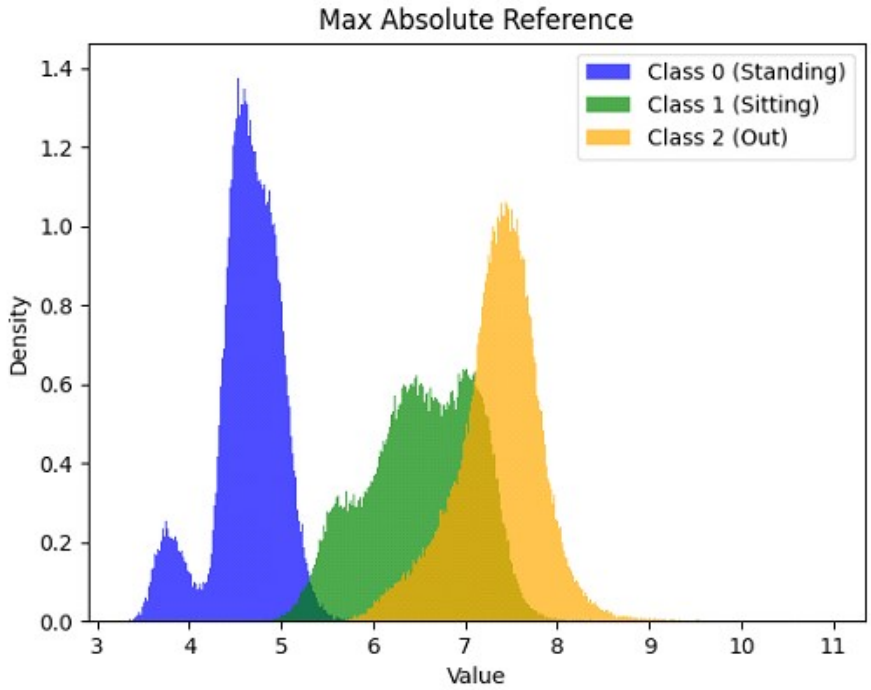

Figure 4.15: Histogram of Max Absolute Reference Antenna.

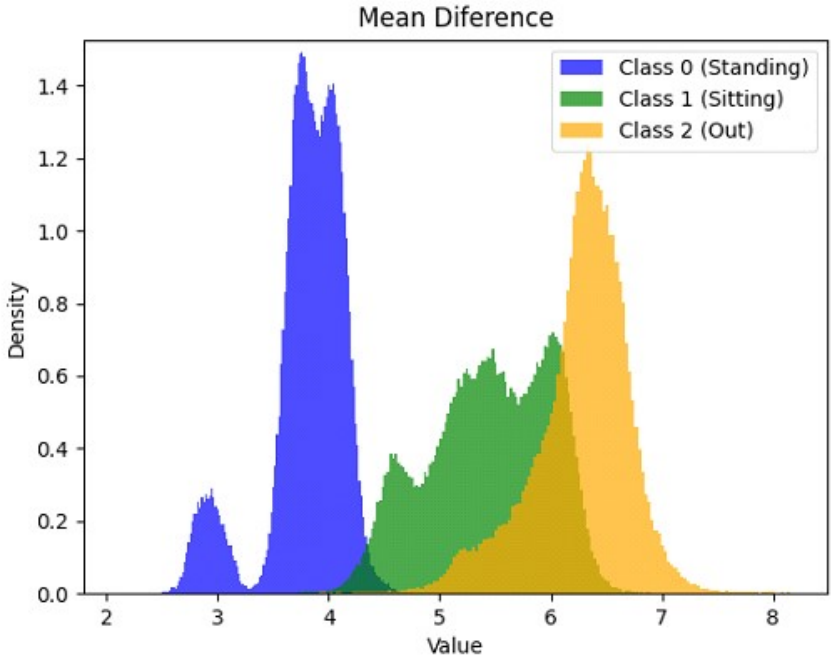

Figure 4.16: Histogram of Mean Difference between Antennas.

It is noteworthy that the majority of the features exhibit considerable separation between classes, suggesting that these types of features may be useful in the classification process.

4.3 Phase Features

Since the VAE did not perform well for Approach 2, and we did not use the dataset from this approach for statistical feature analysis, we will now attempt to extract insights utilizing the signal phase.

The phase difference is calculated by first computing the average phase of 2,500 samples from each antenna. Specifically, we calculate the average phase for 2,500 samples from the reference antenna and the average phase for 2,500 samples from the surveillance antenna. Each set of 2,500 samples corresponds to 10 milliseconds of data from each antenna. The phase difference is then determined by taking the difference between these two average phases.

In Figure 4.17, we present a histogram of the phase difference, which illustrates a complete overlap between the phases, indicating a lack of separation between the classes. Due to this lack of separation, it will be impossible to classify using this feature. Therefore, we will not proceed with this feature.

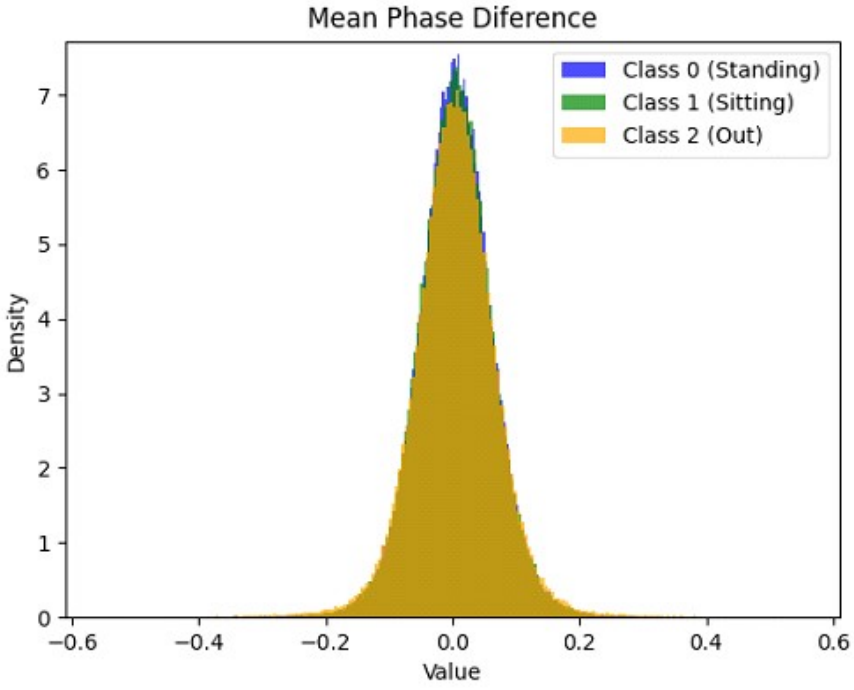

Figure 4.17: Histogram of Phase Difference between Antennas.

Given this complete overlap, we will endeavour to derive some findings via correlation analysis. Correlation assesses how strongly and in which direction two variables are linearly related. By examining correlations, our goal is to uncover any possible relationships that may assist in differentiating between categories.

We provide two figures: Figure 4.18 illustrates the relationship between the phase of the reference antenna and that of the surveillance antenna. Figure 4.19 depicts the correlation between phase differences and the phase of the reference antenna. In both

figures, there is some separation between the classes at the extremes; however, the number of samples in these regions is insufficient to draw reliable conclusions. In the remaining areas, there is a complete overlap between the classes, rendering them indistinguishable.

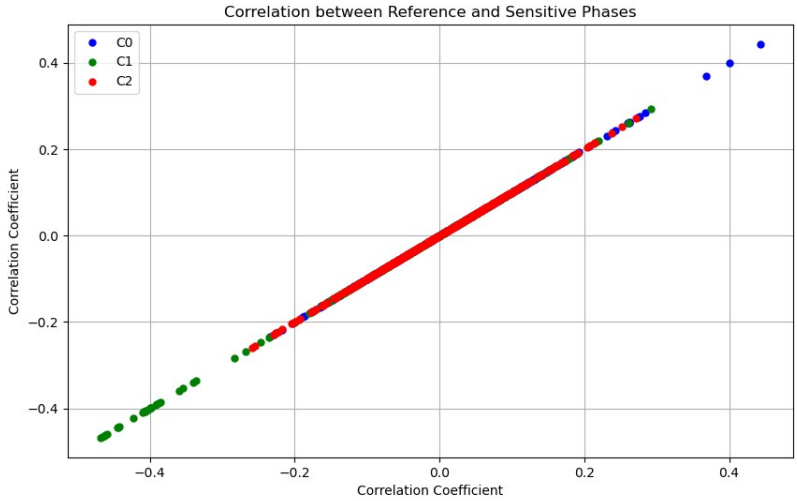

Figure 4.18: Correlation Between Surveillance Antenna and Reference Antenna Phases.

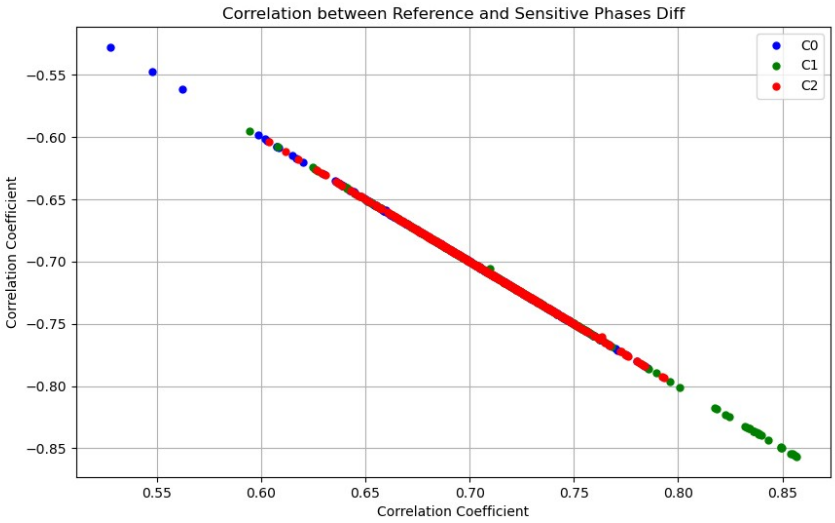

Figure 4.19: Correlation Between Phase Difference and Reference Antenna Phases.

Given that the approach using the phase of the stereo pilots did not produce successful results, we have decided to discontinue it.

4.4 Fast Fourier Transform

Since the phase analysis method could not produce interesting results, we will conduct further research using the FFT. The FFT is a mathematical tool that converts a time-domain signal into its frequency components, enabling the analysis of the frequencies contained

within the data. This approach aids in uncovering patterns and periodicities that may not be readily apparent in the time domain.

For this analysis, we utilized the dataset from Approach 2. A random portion corresponding to 10% of the entire dataset was selected, and FFTs were calculated for three variations: one using 1,024 samples, another with 25,000 samples, and finally, the complete subset of data, covering 3 minutes of data per class, amounting to a total of 45 million samples. However, since the results were quite similar across all variations, only the results for the 25,000 samples will be presented

4.4.1 25,000 Samples

The FFT results for 25,000 samples of the reference and surveillance antenna will be presented, with separate graphs for each class.

4.4.1.1 Reference Antenna

In this subsection, we can see three figures related to the reference antenna for Class 0, Class 1, and Class 2, respectively, labeled as Figures 4.20, 4.21, and 4.22. Each figure presents the FFT representation for 25,000 samples, which corresponds to 10 ms of data. While there is a slight difference in the magnitude peaks, this may not be sufficient for classification purposes.

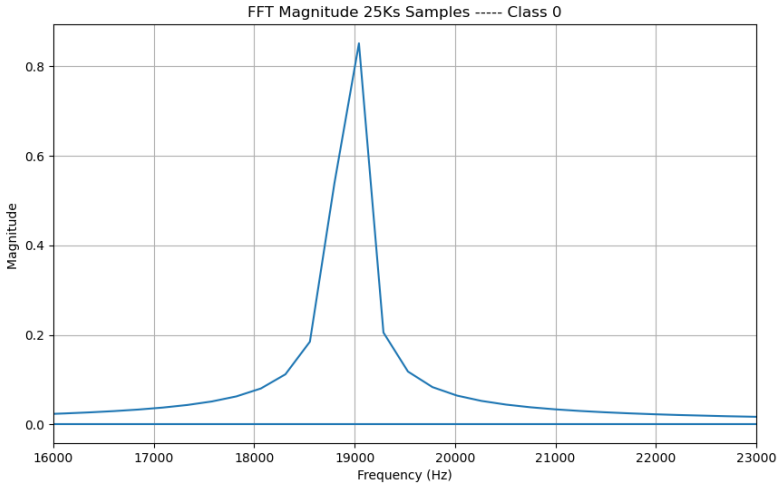

Figure 4.20: FFT for 25,000 Samples (Reference Antenna, Class 0).

4.4.1.2 Surveillance Antenna

In this subsection, we examine three figures pertaining to the surveillance antenna for Classes 0, 1, and 2, which are labeled as Figures 4.23, 4.24, and 4.25. Each figure displays the FFT representation for 25,000 samples, corresponding to 10 ms of data. Although

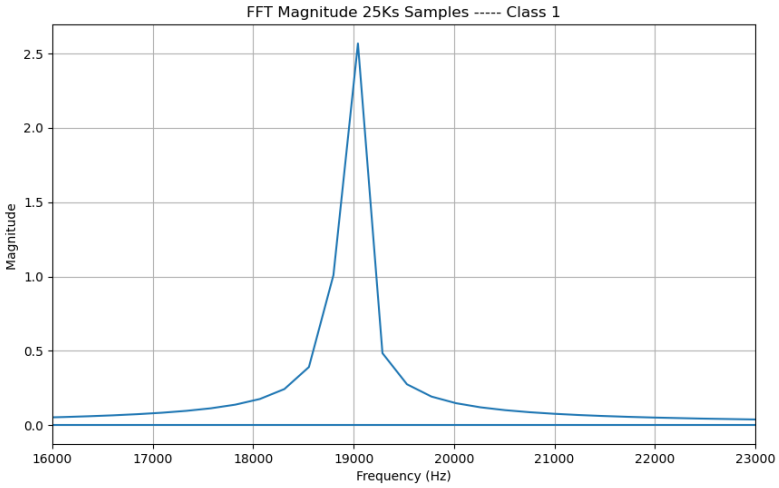

Figure 4.21: FFT for 25,000 Samples (Reference Antenna, Class 1).

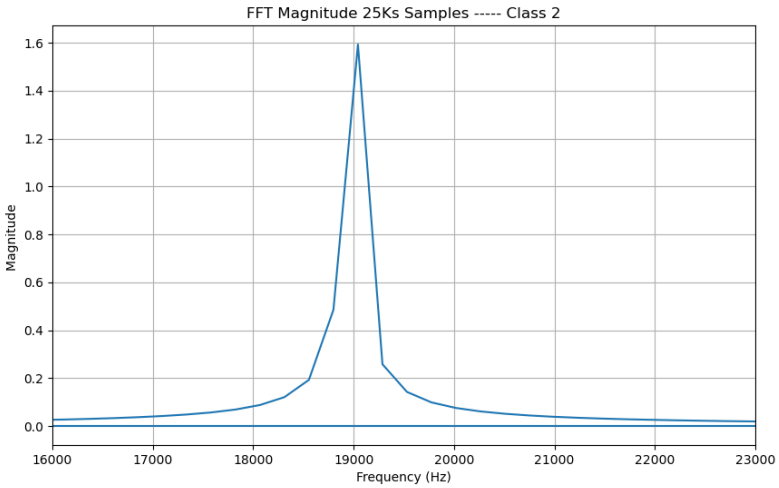

Figure 4.22: FFT for 25,000 Samples (Reference Antenna, Class 2).

there is a slight variation in the magnitude peaks, this information may not be enough for classification purposes.

The FFT analysis can provide insights into how different the classes are; however, it does not yield sufficient information for effective classification. While there may be slight differences in values between classes in some cases, using alternative datasets could lead to significant variations in these results. Moreover, relying on the order of values—whether from highest to lowest or from lowest to highest—poses a challenge, as the relative rankings of classes can change. For instance, even if certain classes consistently have higher values, the order may not remain stable across different datasets. This lack of consistency makes it impossible to use the FFT data reliably for classification purposes.

As such, we will abandon this approach in favor of more stable and reliable methods.

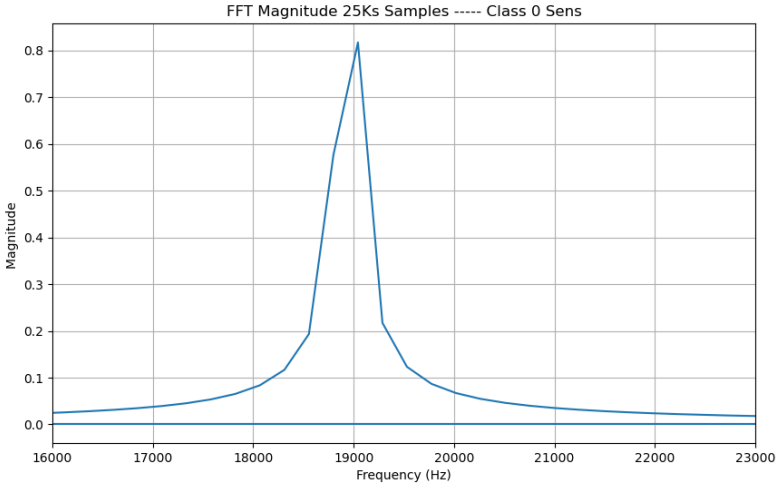

Figure 4.23: FFT for 25,000 Samples (Surveillance Antenna, Class 0).

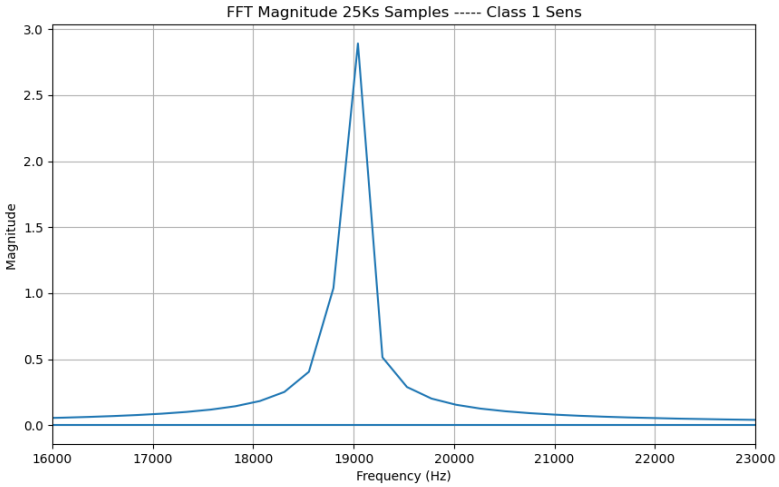

Figure 4.24: FFT for 25,000 Samples (Surveillance Antenna, Class 1).

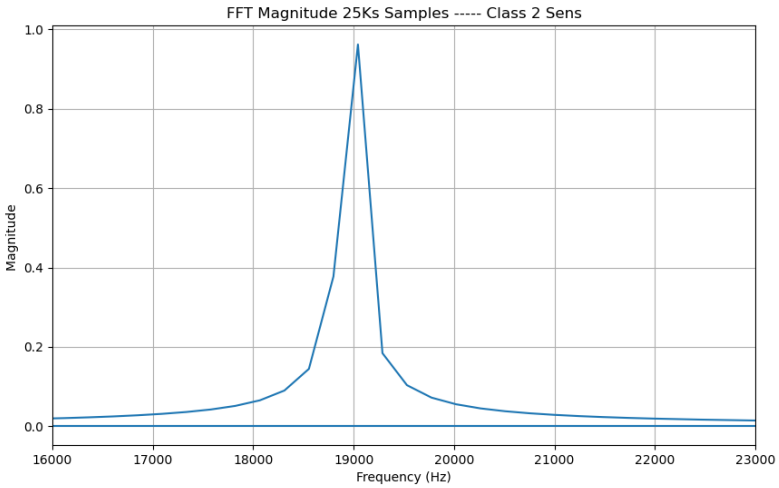

Figure 4.25: FFT for 25,000 Samples (Surveillance Antenna, Class 2).

5 Classification and Performance Evaluation

5.1 Classification

This chapter focuses on the classification and evaluation of the datasets described in Chapter 3. To accomplish this, we utilize a neural network. The structure of the neural network model is outlined as follows:

```
def create_model(input_shape, num_classes):
    model = Sequential()
    model.add(Dense(64, activation='relu', input_shape=(input_shape,)))
    model.add(Dropout(0.2))
    model.add(Dense(32, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    model.compile(optimizer='adam', loss='categorical_crossentropy',
    metrics=['accuracy'])
    return model
```

This model consists of several layers:

- **Input Layer**: The input layer is characterized by the `input_shape` parameter, which specifies the number of features adopted in the network. This component receives the input data.
- **First Hidden Layer**: This layer has a complete connection and comprises 64 neurons, employing the ReLu activation function. The ReLu activation introduces non-linearity to the model, enabling it to capture complex patterns effectively.
- **Dropout Layer**: A layer for dropout is incorporated with a rate set at 0.2. Dropout is a regularisation method, effectively mitigating overfitting by randomly deactivating some input units throughout training. This compels the model to acquire more resilient characteristics that are not overly reliant on particular neurons.

- **Second Hidden Layer**: Another dense layer featuring 32 neurons and utilizing a ReLu activation function. This layer continues to refine the information obtained from the initial hidden layer.

- **Output Layer**: The final layer is a fully connected layer with several neurons corresponding to the total number of categories. The softmax activation function generates a probability distribution across the classes, facilitating multi-class classification.

One-hot encoding was implemented to prepare the labels for neural network training. This method involves the conversion of each class identifier into a binary vector, with the vector's length being equivalent to the number of classes. Except for the position associated with the class identifier, which is set to 1, all vector elements are set to 0. For example, the labels are encoded in the following manner for a three-class problem:

- Class 0: [1, 0, 0] - Class 1: [0, 1, 0] - Class 2: [0, 0, 1]

The Adam optimizer assembles the model, an effective optimization algorithm for deep learning model training. Categorical cross-entropy is the loss function that is appropriate for multi-class classification assignments. Accuracy is employed as a metric to assess the model's performance, as it quantifies the percentage of correctly classified instances. To prepare the datasets for training and validation, 80% of the data was allocated for training, while the remaining 20% was reserved for validation. This split was performed using a train-test split method, followed by shuffling to randomize the distribution of samples. This process ensures that both training and validation sets are representative of the overall dataset, minimizing bias and enhancing the model's robustness. The trained model was saved to a file in the `.h5` format after each neural network was trained. This enables the model to be easily stored and retrieved, thereby facilitating further analysis or deployment without retraining the model from the ground up.

The dataset will be classified into the specified classes, and its performance will be assessed based on accuracy and other pertinent metrics using this neural network architecture.

5.1.1 Variational Autoencoder

This section presents the methodology for extracting features from the dataset by employing a pre-trained VAE. The VAE model was previously trained and saved as a `.h5` file, which will be loaded to produce latent features that accurately represent the input data. The architecture of the VAE has been explained in detail in Section 4.1.

The dataset will be transformed into these latent space representations using the encoder component of the VAE. The extracted features will subsequently be used as the input for classification by the neural network. The `input_shape` of the neural network is determined by the number of latent variables specified by the VAE, thereby guaranteeing that the feature dimensions and the network are compatible.

We aim to improve the neural network's performance by supplying it with informative features derived from the original dataset using the VAE for feature extraction.

5.1.1.1 19Khz - Stereo Pilot

In this subsection, we evaluate the performance of the neural network using the dataset focused on the 19 kHz stereo pilot signal, which corresponds to the dataset from Approach 2. We will present the training graphs for various numbers of latent variables to understand how these configurations influence the loss and accuracy functions of the neural network. By examining these training graphs, we aim to observe how different numbers of latent variables affect the model's learning dynamics and convergence behavior.

10 Latent Variables

In this paragraph, we analyze the training performance of the neural network using 10 latent variables for a three-class classification task. Below, I will present Figure 5.1, which depicts the loss function, and Figure 5.2, which illustrates the accuracy function during the training of the neural network. Each figure provides insights into the model's learning process and effectiveness in classifying the data

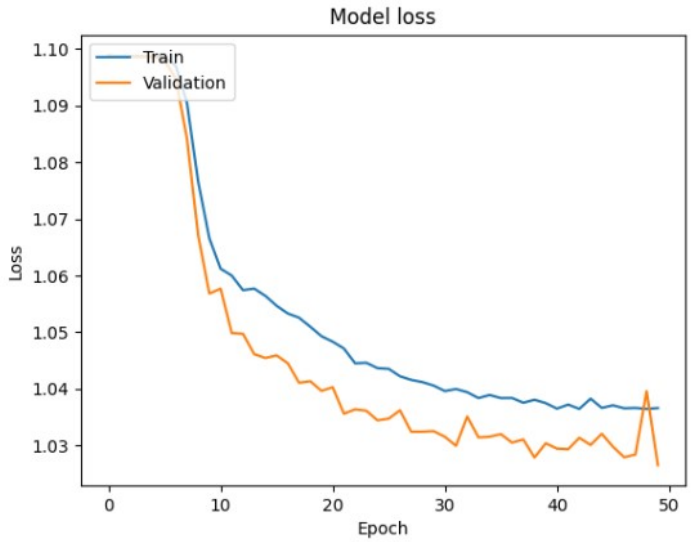

Figure 5.1: Training and validation loss over epochs.

Loss Analysis: The training and validation loss curves are very close, both converging to approximately 1.04. This proximity suggests the model is not overfitting, as it performs similarly on both the training and validation sets. However, the high loss value indicates that the model struggles to learn effectively from the data.

Accuracy Analysis: The accuracy for both training and validation stabilizes around 0.46, showing that the model correctly classifies 46% of the instances. The close alignment of the accuracy curves confirms that the model is generalizing well. However, the low

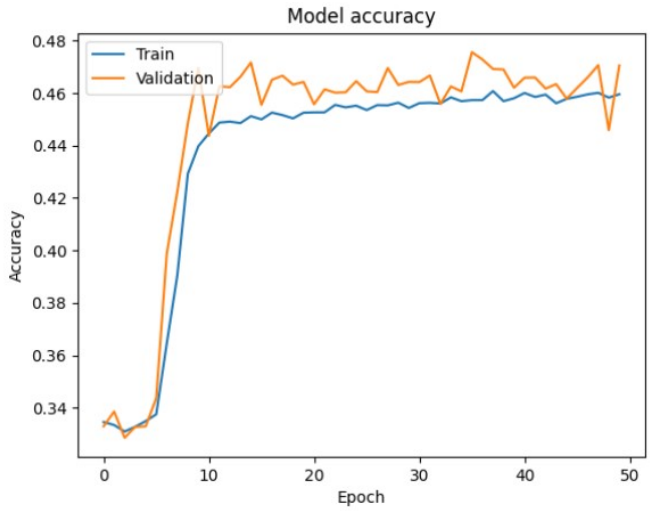

Figure 5.2: Training and validation accuracy over epochs.

accuracy indicates difficulty distinguishing between the three classes with the current configuration.

25 Latent Variables

In this paragraph, we examine the training performance of the neural network that employs 25 latent variables for a three-class classification task. Below, I will present Figure 5.3, which shows the loss function, and Figure 5.4, which demonstrates the accuracy function throughout the training of the neural network. Each figure offers valuable insights into the model's learning behavior and its effectiveness in classifying the data.

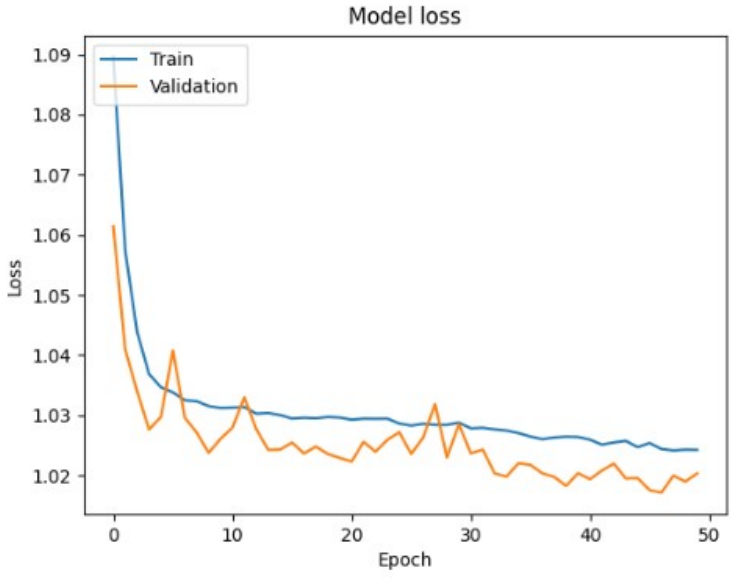

Figure 5.3: Training and validation loss over epochs for 25 latent variables.

Loss Analysis: As observed in the previous subsection, the loss for training and validation remains close, converging to approximately 1.03. This indicates that the model continues to face challenges in learning effectively despite the increase in latent variables.

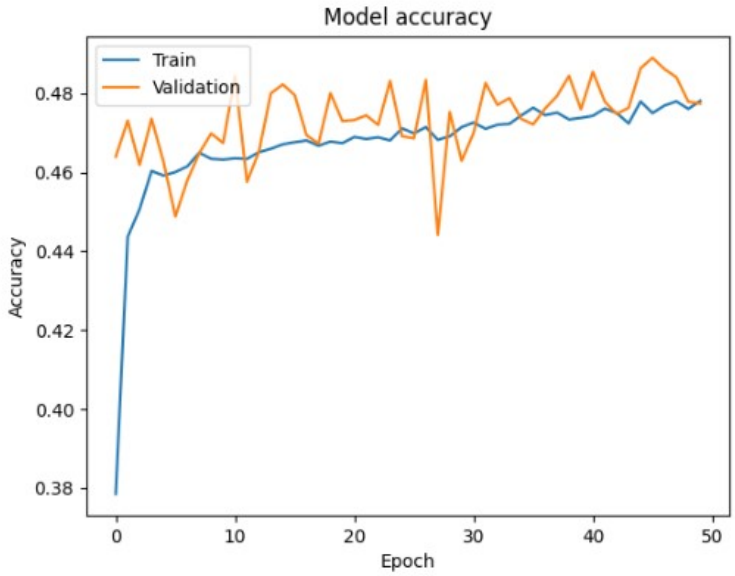

Figure 5.4: Training and validation accuracy over epochs for 25 latent variables.

Accuracy Analysis: The accuracy for training and validation stabilizes around 0.47. This slightly improved over the 0.46 accuracy observed with 10 latent variables, but the overall performance trend remains consistent. The model's ability to distinguish between classes shows only marginal enhancement with the increased number of latent variables.

50 Latent Variables

In this paragraph, we focus on the training performance of the neural network utilizing 50 latent variables for a three-class classification task. Below, I will introduce Figure 5.5, which illustrates the loss function, alongside Figure 5.6, which depicts the accuracy function during the training process. Both figures shed light on the model's learning dynamics and its capability in accurately classifying the data.

Loss Analysis: As observed with 25 latent variables, the loss for both training and validation converges to approximately 1.02. This consistency suggests that the model still encounters challenges in effectively minimizing the loss, even with more latent variables.

Accuracy Analysis: The accuracy for training and validation stabilizes around 0.49. This represents a slight improvement over the 0.47 accuracy observed with 25 latent variables, but the overall performance trend remains similar. The model's capacity to distinguish between the classes has seen only marginal enhancement with the increase in latent variables, indicating that further improvements may require additional adjustments or alternative strategies.

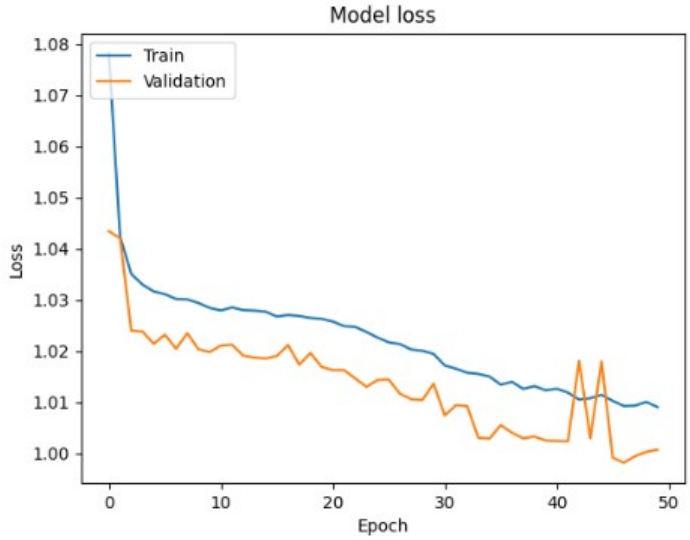

Figure 5.5: Training and validation loss over epochs for 50 latent variables.

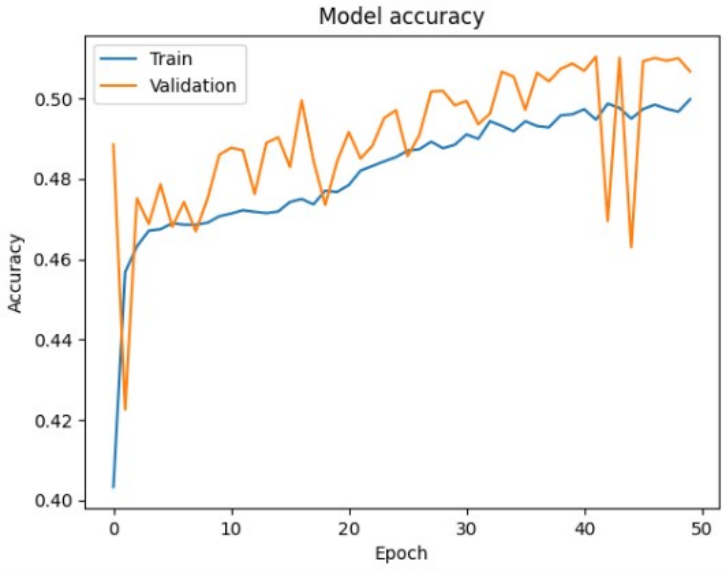

Figure 5.6: Training and validation accuracy over epochs for 50 latent variables.

5.1.1.2 Complete Band

In this subsection, we evaluate the neural network's performance using the dataset focused on the Complete Band of a single FM channel, which is sourced from Approach 1. We will present the training graphs for various numbers of latent variables to understand how these configurations influence the loss and accuracy functions of the neural network. By examining these training graphs, we aim to observe how different numbers of latent variables affect the model's learning dynamics and convergence behavior. Based on the histogram insights, we expect to achieve better results with the Complete Band dataset than with the 19 kHz stereo pilot signal.

10 Latent Variables

This paragraph analyzes the training performance of the neural network using 10 latent variables for a three-class classification task. Below, I will present Figure 5.7, which reflects the loss function, and Figure 5.8, which outlines the accuracy function throughout the training phase. Each figure provides critical insights into the learning process of the model and its effectiveness in data classification.

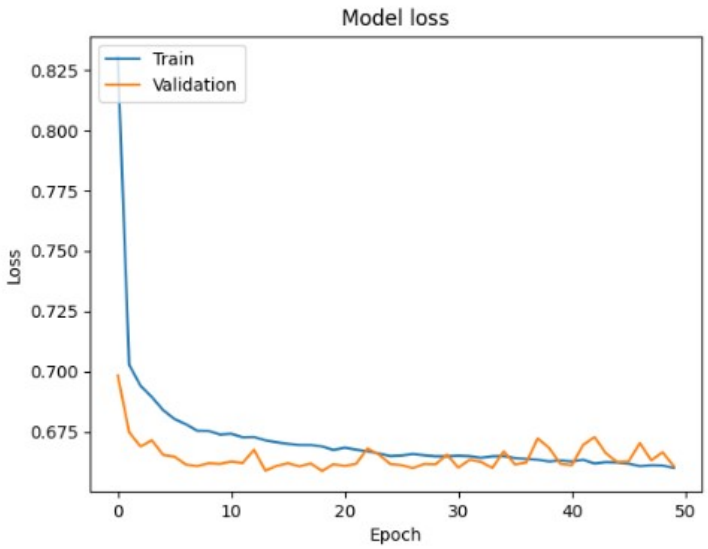

Figure 5.7: Training and validation loss over epochs.

Loss Analysis: The training and validation loss curves are closely aligned, converging to approximately 0.675. This represents a significant improvement over previous configurations, with higher loss. The close alignment of the curves indicates that the model is not overfitting and is performing consistently on both the training and validation sets, showing a more effective learning process.

Accuracy Analysis: The accuracy for both training and validation stabilizes around 0.68, a notable improvement compared to previous results where accuracy was lower. The close alignment of the accuracy curves further confirms that the model is generalizing

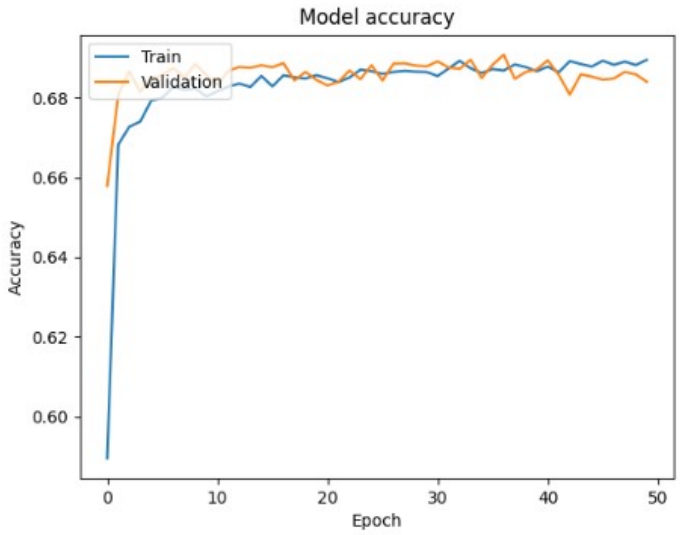

Figure 5.8: Training and validation accuracy over epochs.

well, and the increased accuracy indicates a better capability in distinguishing between the three classes with the current configuration.

25 Latent Variables

In this paragraph, we assess the training performance of the neural network that incorporates 25 latent variables for a three-class classification task. Below, I will present Figure 5.9, which represents the loss function, and Figure 5.10, which illustrates the accuracy function during the network's training. Both figures give insights into the model's learning process and how well it classifies the data.

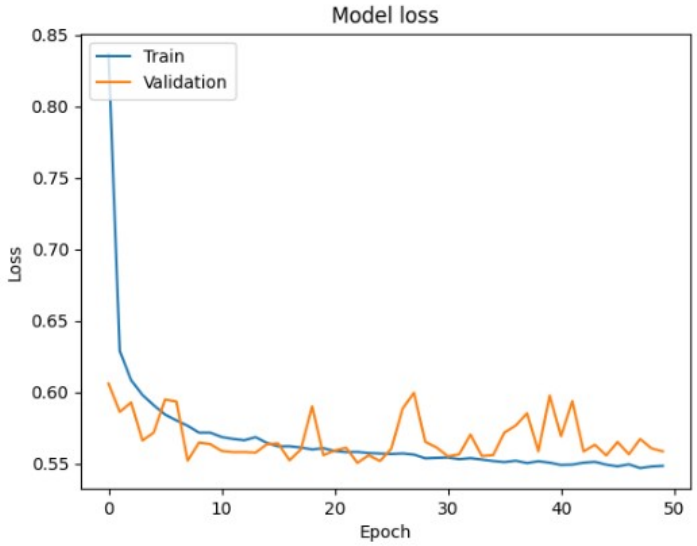

Figure 5.9: Training and validation loss over epochs with 25 latent variables.

Loss Analysis: The loss curves for both training and validation converge to approximately 0.56, showing a considerable improvement over the results obtained with fewer latent variables. This lower loss indicates that the model has become more adept at learning from the data, and the close alignment of the curves suggests effective training without overfitting.

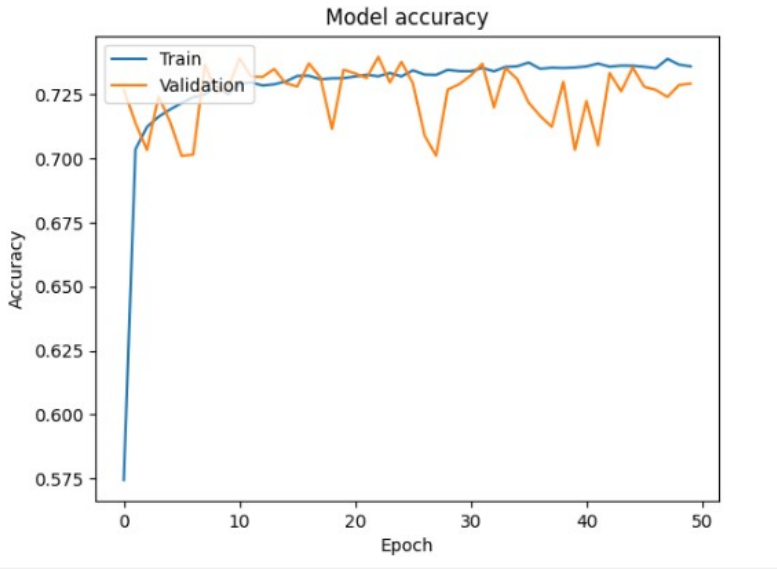

Figure 5.10: Training and validation accuracy over epochs with 25 latent variables.

Accuracy Analysis: The accuracy has improved significantly, stabilizing around 0.72 for both training and validation. This enhancement in accuracy indicates that the model is now more proficient at classifying the three classes accurately. The close alignment of the accuracy curves further confirms that the model is generalizing well with the increased number of latent variables, resulting in better overall performance.

50 Latent Variables

In this paragraph, we evaluate the training performance of the neural network using 50 latent variables aimed at a three-class classification task. Below, I will introduce Figure 5.11, showcasing the loss function, and Figure 5.12, which presents the accuracy function over the training period. Each figure provides important insights into the model's learning capabilities and its effectiveness in classifying the data.

Loss Analysis: The loss curves for both training and validation converge to approximately 0.60. While this represents a slight increase compared to the loss observed with 25 latent variables, the model still shows consistent learning trends. The higher loss indicates that the model's performance has slightly degraded with the additional latent variables, suggesting that the increase in complexity might not have led to better learning outcomes.

Accuracy Analysis: The accuracy stabilizes around 0.72 for training and 0.70 for validation. Although these values are close to those achieved with 25 latent variables, they represent a slight decrease. This marginal reduction in accuracy suggests that while

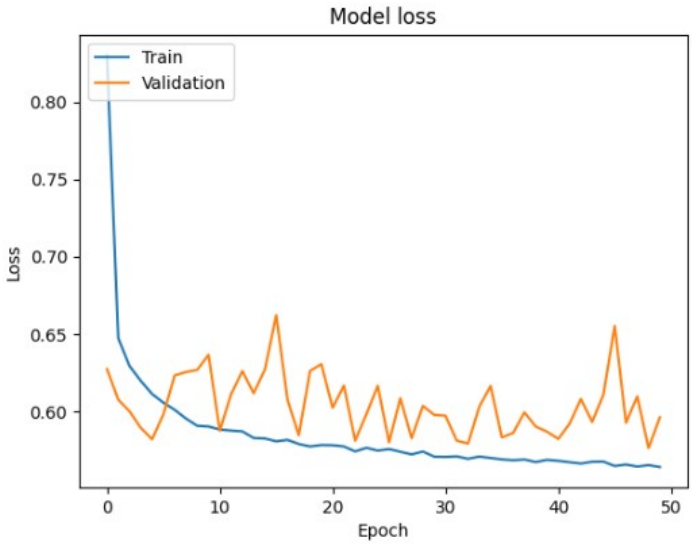

Figure 5.11: Training and validation loss over epochs with 50 latent variables.

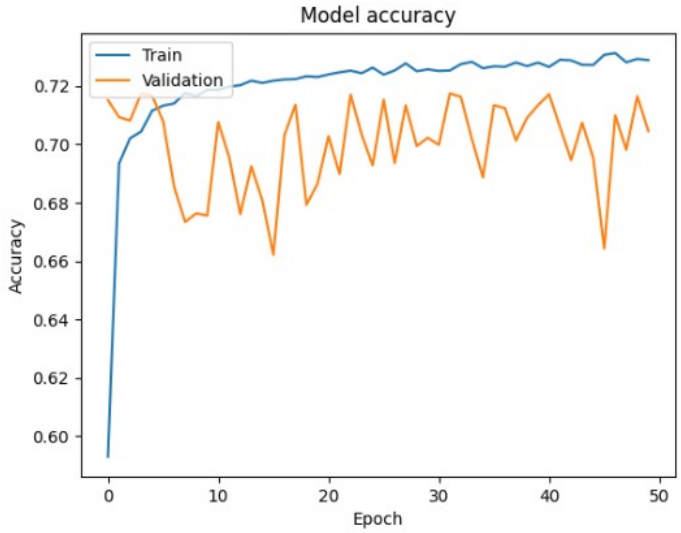

Figure 5.12: Training and validation accuracy over epochs with 50 latent variables.

the model performs reasonably well, the increase in latent variables has not significantly improved and may have introduced additional complexity that does not translate into better generalization.

For all the cases presented above, both for Approach 1 and Approach 2, the computational impact of increasing or decreasing the number of latent variables was practically negligible.

5.1.2 Statistical Features

In this subsection, we will use the 15 statistical features previously presented in Section 4.2 as inputs to the neural network, utilizing the dataset from Approach 1, specifically the Complete Band of a single FM channel. Our objective is to evaluate the performance of the neural network when these statistical features are used for training. In Figure 5.13, I present the graph of the loss function, and in Figure 5.14, I show the accuracy function.

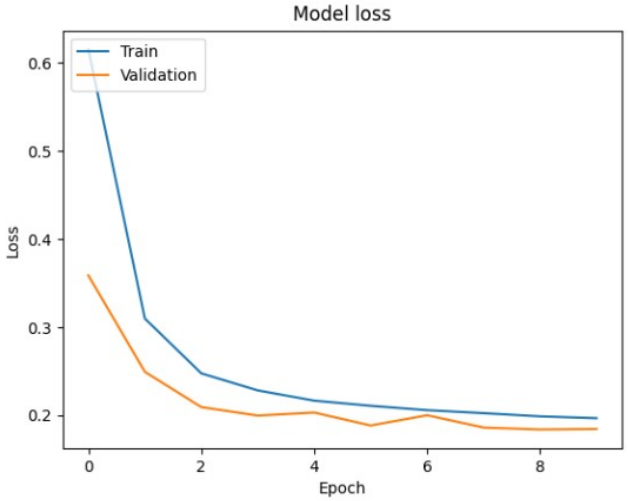

Figure 5.13: Training and validation loss over epochs with 15 statistical features.

Loss Analysis: The loss curves for both training and validation converge to approximately 0.2, indicating a substantial improvement in the model's ability to learn from the data. This lower loss value suggests that the neural network is effectively capturing the underlying patterns within the dataset, and the close alignment of the training and validation loss curves demonstrates that the model is being trained efficiently without overfitting.

Accuracy Analysis: The accuracy has significantly improved, stabilizing around 0.9 for training and validation. These high accuracy levels indicate that the model is highly effective at correctly classifying the three classes. The close alignment of the accuracy curves for training and validation confirms that the model is generalizing very well, showcasing strong performance with the increased number of latent variables.

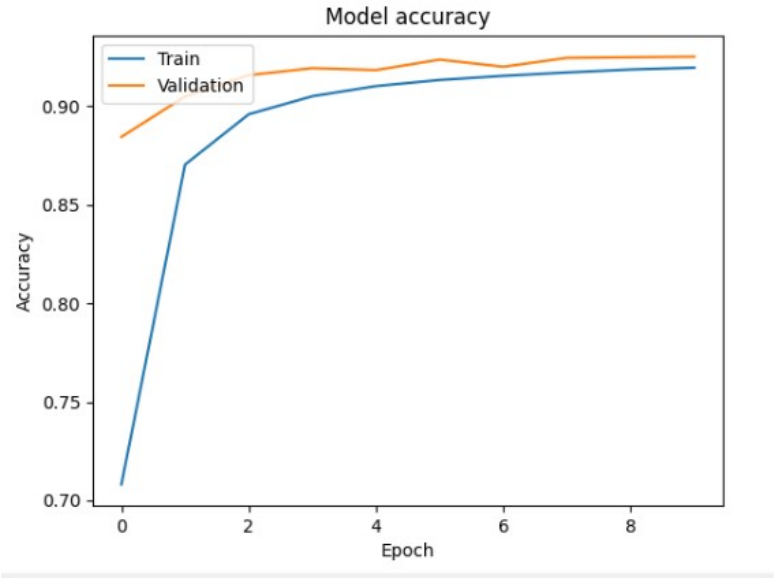

Figure 5.14: Training and validation accuracy over with 15 statistical features.

5.2 Performance Evaluation - Results and Discussion

In this section, we will present and analyze the performance metrics for the different cases referred to in this section, as explained in 2.5. The metrics, including accuracy, precision, recall, and F1-score, will be detailed in tables for each approach. After presenting all the data, we will briefly discuss the results, followed by a conclusion summarizing the key insights and implications of our findings.

5.2.1 VAE - Approach 2

Here, we focus on Approach 2 using the VAE. We will present this approach's specific performance metrics and confusion matrices. These results will help us evaluate how well Approach 2 captures the dataset's characteristics and performs effectively in the three-class classification task. The confusion matrices will provide a detailed breakdown of the model's classification performance, highlighting areas of strength and potential improvement.

5.2.1.1 10 Latent Variables

This paragraph presents the classification metrics of the model in Table 5.1, along with the confusion matrix displayed in Figure 5.15 for Approach 2, which utilizes 10 latent variables. The results show an accuracy of 0.46, which remains relatively low and unsatisfactory. The confusion matrix indicates that the model frequently predicted Class 1, leading to a greater number of predicted instances compared to the actual cases; however, this also highlights a significant number of errors. Therefore, we can conclude that the method of

extracting 10 features using a VAE and employing a neural network for classification does not yield a viable system.

Class	Precision	Recall	F1-Score
Class 0	0.59	0.47	0.52
Class 1	0.42	0.85	0.55
Class 2	0.45	0.08	0.13
Accuracy	0.46		

Table 5.1: Performance metrics for each class for the Aproach 2 with 10 Latent Variable.

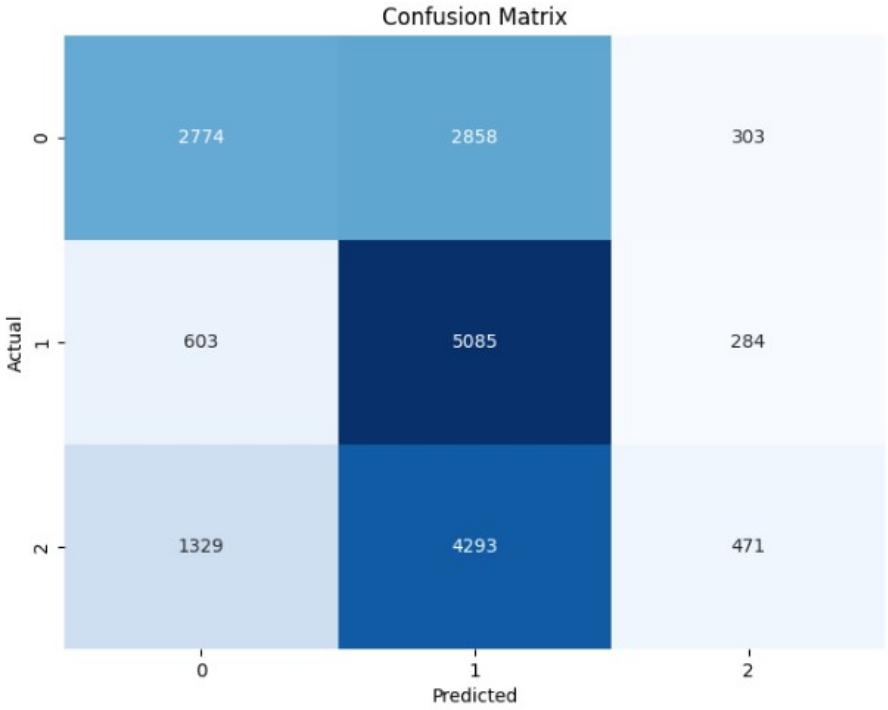

Figure 5.15: Confusion matrix for the Aproach 2 with 10 latent variables.

5.2.1.2 25 Latent Variables

In this paragraph, we present the classification metrics of the model in Table 5.2, along with the confusion matrix shown in Figure 5.16 for Approach 2, with 25 latent variables. The results indicate an accuracy of 0.48, which is still relatively low and unsatisfactory. The confusion matrix reveals that the model often predicted Class 1, resulting in a higher count of predicted instances compared to the actual cases; however, this also signifies a considerable number of errors. Thus, we can conclude that the approach of extracting 25 features using a VAE and applying a neural network for classification does not produce a viable system. Additionally, increasing the number of latent variables led to only a slight improvement in accuracy.

Class	Precision	Recall	F1-Score
Class 0	0.63	0.43	0.51
Class 1	0.43	0.81	0.56
Class 2	0.45	0.2	0.26
Accuracy	0.48		

Table 5.2: Performance metrics for each class for the Aproach 2 with 25 Latent Variable.

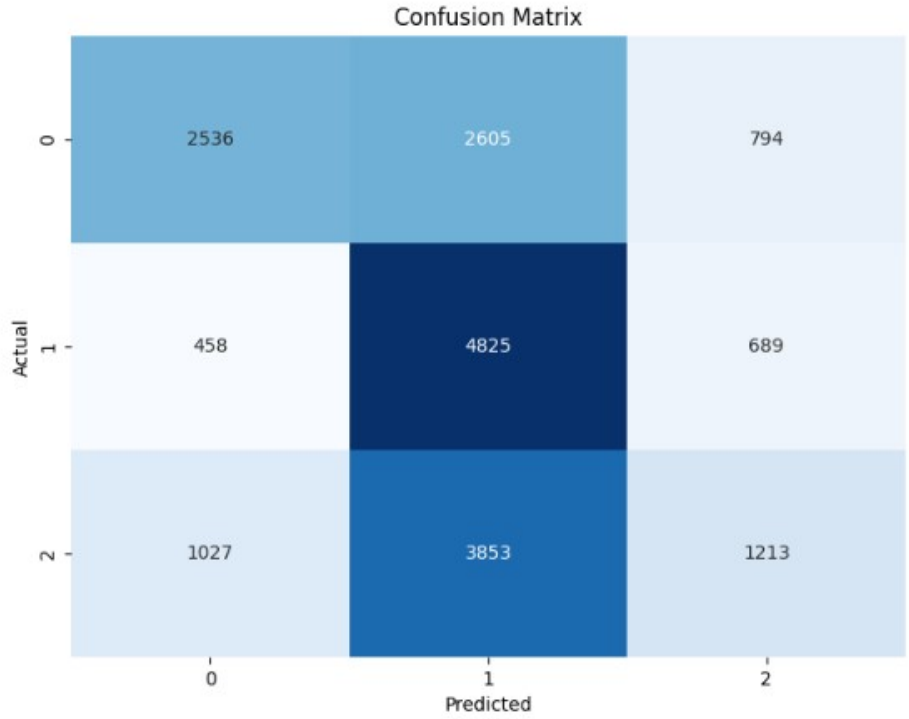

Figure 5.16: Confusion matrix for the Aproach 2 with 10 latent variables.

5.2.1.3 50 Latent Variables

In this paragraph, we present the classification metrics of the model in Table 5.3, along with the confusion matrix in Figure 5.17 for Approach 2, with 50 latent variables. The results reveal an accuracy of 0.5, which remains relatively low and unsatisfactory. The confusion matrix indicates that the model frequently predicted Class 1, which corresponds to a higher number of predicted instances compared to the actual cases; however, this also reflects a significant number of errors. Therefore, we can conclude that the method of extracting 50 features using a VAE and utilizing a neural network for classification does not yield a viable system. Moreover, increasing the number of latent variables resulted in only a modest enhancement in accuracy.

For Approach 2, where only the frequency sensing of the stereo pilot was used, we conclude that this method does not provide the necessary information for effective classification. The best accuracy achieved was 50%, indicating that the model's performance

Class	Precision	Recall	F1-Score
Class 0	0.61	0.46	0.52
Class 1	0.46	0.68	0.55
Class 2	0.47	0.37	0.42
Accuracy	0.50		

Table 5.3: Performance metrics for each class for the Aproach 2 with 50 Latent Variable.

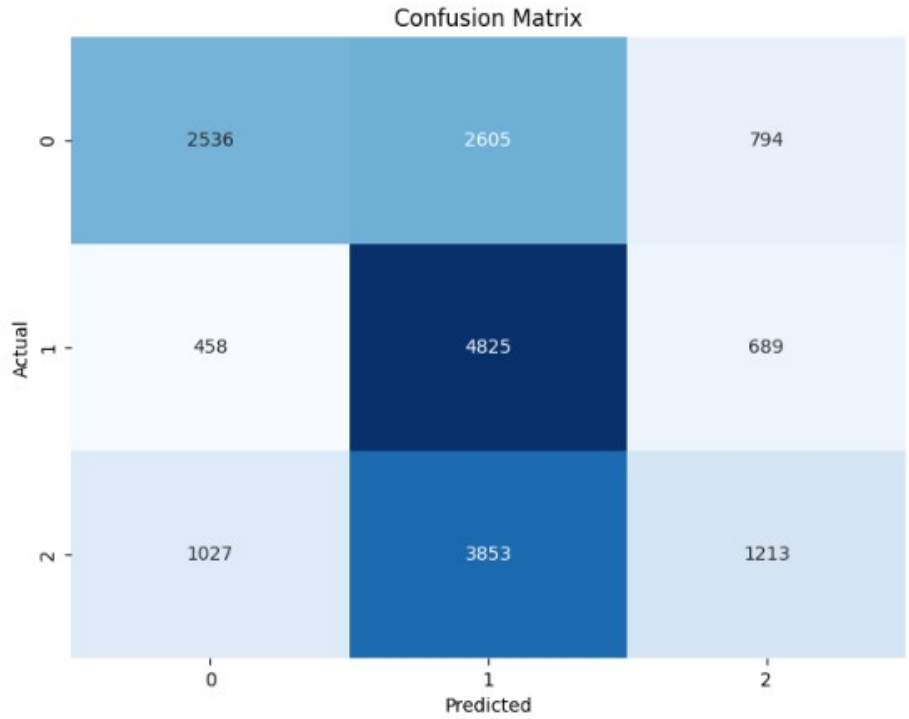

Figure 5.17: Confusion matrix for the Aproach 2 with 50 latent variables.

was essentially at chance level for a three-class problem. The confusion matrices reveal that the model frequently predicted Class 1, but this performance is misleading, as it often confuses Class 1 with Class 2 and Class 0, resulting in many outputs being classified as Class 1. Although the model showed the highest performance for Class 1, the significant overlap with the other classes reflects a major concern. Additionally, while increasing the number of features extracted through the VAE led to only a minimal enhancement in accuracy, this marginal improvement may not justify the added complexity. Despite these observations, the overall results were unsatisfactory, demonstrating the limitations of relying solely on the stereo pilot frequency for classification.

5.2.2 VAE - Approach 1

Here, we focus on Approach 1, which utilizes a different strategy for the classification task. We will present this approach's specific performance metrics and confusion matrices. These results will allow us to assess how effectively Approach 1 captures the dataset's

characteristics and performs in the three-class classification task. The confusion matrices will offer a detailed view of the model's classification performance, revealing its strengths and potential improvement areas.

5.2.2.1 10 Latent Variables

This paragraph presents the classification metrics of the model in Table 5.4, along with the confusion matrix displayed in Figure 5.18 for Approach 1, with 10 latent variables extract by a VAE. The results show an accuracy of 0.69, which is a more satisfactory outcome. The confusion matrix reveals that the classes with the fewest errors are Class 2 and Class 0, meaning it is very rare for Class 0 to be predicted as Class 2, and vice versa. We can also observe that there are more correct predictions for Classes 0 and 2, while Class 1, though still producing reasonably good results, has the fewest true positives compared to the other classes. Overall, the results are promising and suggest that this approach is viable for classification.

Class	Precision	Recall	F1-Score
Class 0	0.81	0.75	0.78
Class 1	0.54	0.51	0.52
Class 2	0.70	0.80	0.75
Accuracy	0.69		

Table 5.4: Performance metrics for each class for the Aproach 1 with 10 Latent Variable.

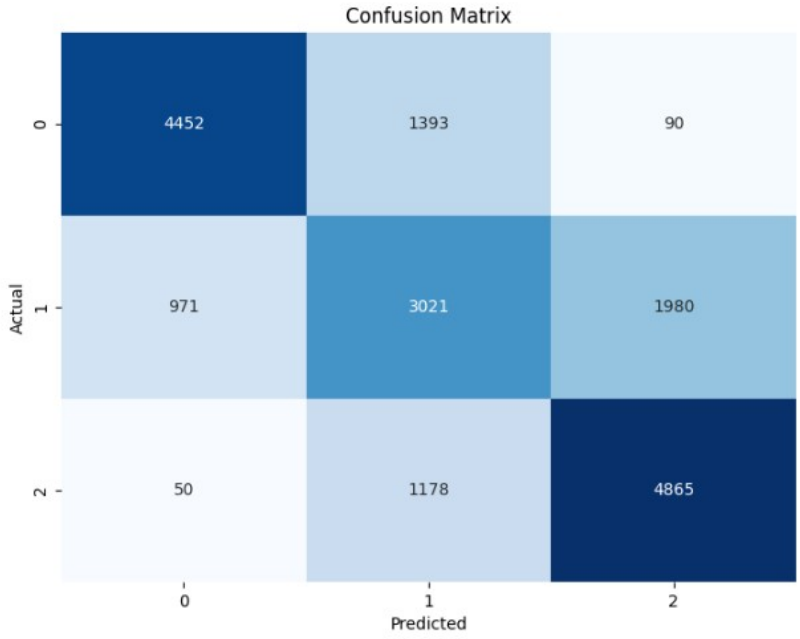

Figure 5.18: Confusion matrix for the Aproach 1 with 10 latent variables.

5.2.2.2 25 Latent Variables

This paragraph presents the classification metrics of the model in Table 5.5, along with the confusion matrix displayed in Figure 5.20 for Approach 1, which uses 25 latent variables extracted through a VAE. The results show an accuracy of 0.72, reflecting a slight improvement compared to using only 10 latent variables. The confusion matrix demonstrates that Classes 0 and 2 had the fewest errors, with the model rarely misclassifying one as the other. Although Class 1 has the lowest number of true positives, the results for this class are still reasonable overall. The improvement in performance with 25 latent variables suggests that increasing the feature set enhances classification, making this approach appear viable for the task.

Class	Precision	Recall	F1-Score
Class 0	0.78	0.86	0.81
Class 1	0.60	0.53	0.56
Class 2	0.78	0.78	0.78
Accuracy	0.72		

Table 5.5: Performance metrics for each class for the Aproach 1 with 25 Latent Variable.

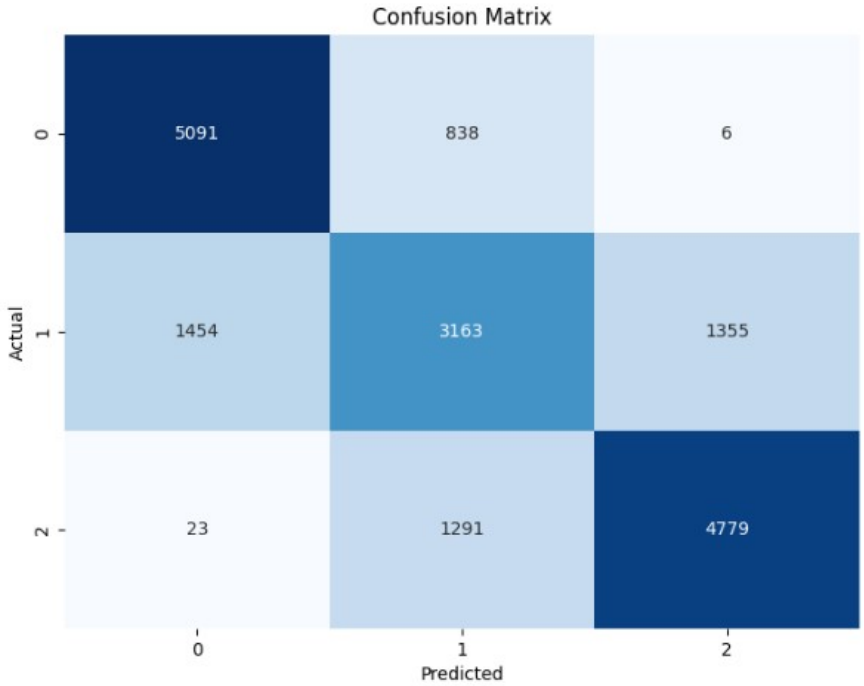

Figure 5.19: Confusion matrix for the Aproach 1 with 25 latent variables.

5.2.2.3 50 Latent Variables

This paragraph presents the classification metrics of the model in Table 5.6, along with the confusion matrix displayed in Figure 5.20 for Approach 1, utilizing 50 latent variables extracted through a VAE. The results show an accuracy of 0.7, which is slightly lower than the 0.72 achieved with 25 latent variables. The confusion matrix highlights that the model still performed best with Classes 0 and 2, with few misclassifications between them. However, Class 1 continues to show the fewest true positives, though its performance remains reasonable. Despite the increased number of latent variables, the slight drop in accuracy suggests diminishing returns, making the 25-variable model a more efficient option for classification.

Class	Precision	Recall	F1-Score
Class 0	0.74	0.91	0.81
Class 1	0.57	0.51	0.54
Class 2	0.80	0.69	0.74
Accuracy	0.70		

Table 5.6: Performance metrics for each class for the Aproach 1 with 50 Latent Variable.

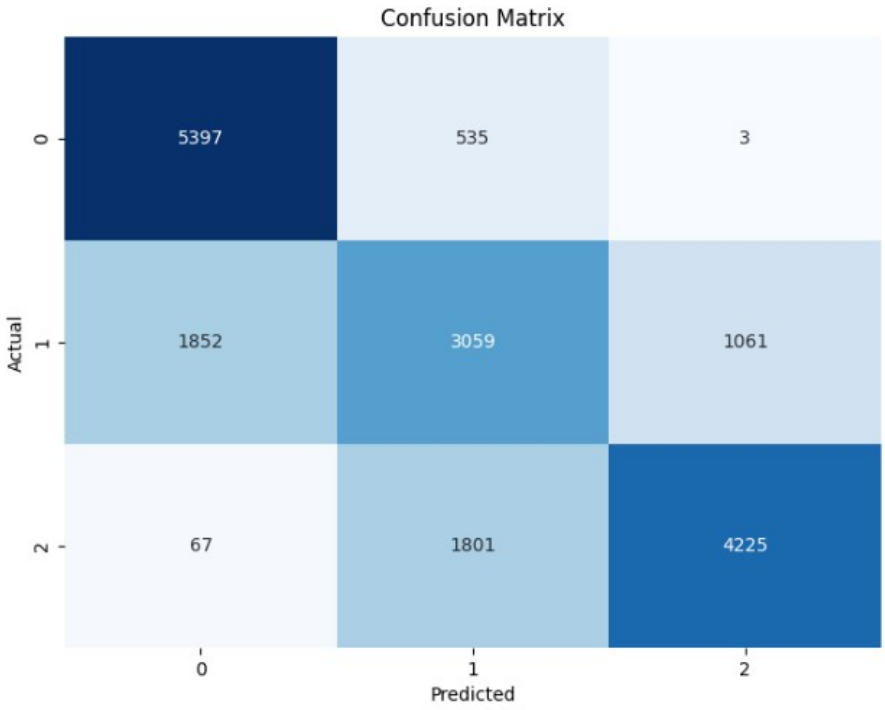

Figure 5.20: Confusion matrix for the Aproach 1 with 50 latent variables.

For Approach 1, where we utilized the complete band sensing, the results were notably improved compared to Approach 2. The best accuracy achieved was 0.72 with 25 latent variables, demonstrating a significant enhancement in performance. All models in this

approach displayed similar behavior, with improvements across the board. However, it is important to note that while the model frequently predicts Class 1 correctly, it has a lower number of true positive cases compared to the other classes. This suggests that although the model performs better with the complete band sensing, Class 1 is not classified as accurately as Classes 0 and 2, indicating room for further improvement.

5.2.3 Statistical Features

Here, we focus on the approach utilizing statistical features, representing a different classification task strategy. This analysis is based on the dataset from Approach 1, specifically the Complete Band of a single FM channel. We will present this approach's specific performance metrics in Table 5.7 and the confusion matrix in Figure 5.21. These results will allow us to evaluate how effectively the statistical features capture the dataset's characteristics and enhance performance in the three-class classification task. The confusion matrices will provide a detailed view of the model's classification performance, highlighting strengths and areas where further improvement is needed.

Class	Precision	Recall	F1-Score
Class 0	0.99	1.00	0.99
Class 1	0.84	0.95	0.89
Class 2	0.95	0.83	0.88
Accuracy	0.92		

Table 5.7: Performance metrics for each class for the complete band with statistical features.

The results showed a significant improvement in the approach utilizing statistical features compared to previous methods. The best accuracy achieved was 0.92, indicating excellent performance. While the model still exhibits some errors, particularly with Class 1, these are minimal. Overall, statistical features have proven to be highly effective, leading to a substantial boost in classification accuracy and demonstrating the model's ability to perform well with the given dataset.

5.3 Real-time Prototype

In this section, we detail the real-time operation of the prototype. For this purpose, we selected the two best-performing cases identified previously: the full-band case using 25 latent variables and the case utilizing statistical features.

A new flowgraph was developed, resembling the one depicted in Figure 5.22. However, instead of storing the data in a file, this flowgraph incorporates a GNU Radio block known as `ZMQ Push`. The `ZMQ Push` block is a key component that facilitates data transmission to a ZeroMQ (ZMQ) socket. It operates by pushing data to a connected ZMQ Pull socket over a TCP connection, enabling real-time data streaming between different system

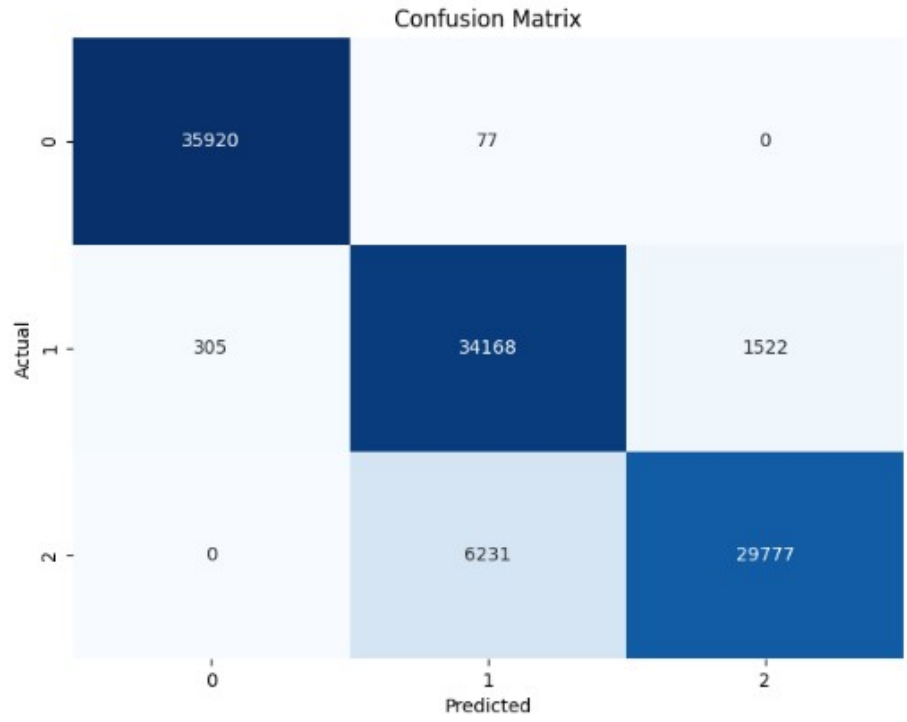

Figure 5.21: Confusion matrix for the complete band with statistical features.

components. This setup allows for efficient and seamless communication, crucial for real-time processing.

Subsequently, a Python script is employed to load the necessary `h5` files and listen to the TCP port. The script reads 20 ms of data at a time from the TCP port and applies an autoencoder for the 25 latent variables or utilizes the 15 statistical features, as appropriate. Following this, classification is carried out using a neural network, also loaded from an `h5` file. The neural network makes a decision every 20 ms, but the prototype's final decision is based on the most frequent decision made over a 1-second window.

For a demonstration of the prototype in operation, please refer to the following video: Prototype Demonstration Video.

The results indicate that the model exhibits a high level of accuracy, albeit with some minor errors. Its performance is nearing the capabilities of an active radar system, demonstrating significant potential for real-time applications. Further refinements and testing are expected to enhance the model's accuracy and reliability. Notably, the performance of the approach using statistical features and the approach based on the VAE showed similar results, suggesting that both methods are equally effective for the given task

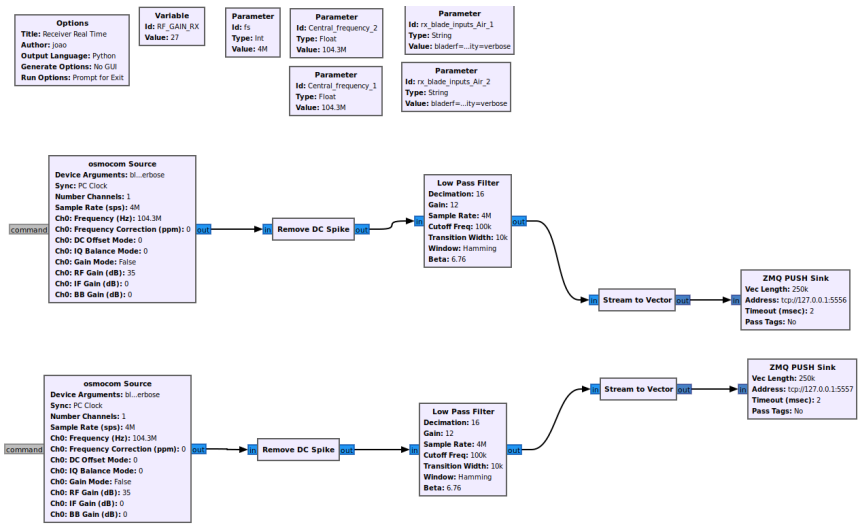

Figure 5.22: Flowgraph used for real-time data streaming and processing.

6 Conclusions

In this chapter, we summarize our research findings and discuss the implications of our results. We also outline potential directions for future research and development.

6.1 Final Considerations

During this thesis, we explored a different approach by attempting to sense the frequency of the stereo pilot. However, this method was unsuccessful, as we concluded that there was insufficient information at that frequency for the intended purposes. Conversely, we obtained significantly positive results when sensing the entire frequency band. Implementing the VAE for feature extraction added value to the project by providing a distinct approach, but in terms of performance, it did not yield the best results. Various numbers of latent variables were tested, and we concluded that the best tradeoff for the number of features in this case was 25. We achieved interesting results offline with the VAE, demonstrating a fully functional prototype capable of classifying human poses in real time.

In addition, we employed 15 statistical features from the dataset of Approach 1, specifically the Complete Band of a single FM channel. This approach yielded the best results in offline evaluations, outperforming the VAE method. In real-time scenarios, however, both systems exhibited similar performance in classifying human poses.

6.2 Future Work

In light of the recent findings, there are multiple pathways that can be pursued for further research and development:

- **Enriching Datasets:** To improve the robustness and versatility of the model, it is recommended to expand the dataset by collecting data in more diverse positions. This includes postures such as standing with arms open, sitting with arms open,

and varying torso inclinations. Additionally, capturing data from scenarios where the person is out of the scene, i.e., at a minimum distance from the setup but in various locations, will help understand the system's performance under different spatial configurations.

- **Adding Additional Antennas:** Introducing an additional antenna to the current setup, increasing from two to three antennas, could provide richer data and improve the system's accuracy. This enhancement might lead to better spatial resolution and more precise feature extraction, thus enriching the overall setup.

- **Testing VAE Performance and Exploring New Classifiers:** Further investigation into the performance of the VAE is essential. This includes optimizing its configuration for better feature extraction and exploring its limits. Additionally, testing new classifiers that could offer improved real-time performance is suggested. Evaluating different machine learning algorithms may reveal more efficient or accurate options for this specific application.

Future work in these areas is expected to improve the model's performance and extend its applicability to a wider range of real-time monitoring and analysis tasks.

Bibliography

[1] S. Kianoush et al. "Device-Free RF Human Body Fall Detection and Localization in Industrial Workplaces". In: *IEEE Internet of Things Journal* 4.2 (2017), pp. 351–362. DOI: 10.1109/JIOT.2016.2624800 (cit. on pp. 1, 20).

[2] S. Sigg et al. "RF-Sensing of Activities from Non-Cooperative Subjects in Device-Free Recognition Systems Using Ambient and Local Signals". In: *IEEE Transactions on Mobile Computing* 1 (2013-03). DOI: 10.1109/TMC.2013.28 (cit. on pp. 1, 20).

[3] J. Zhao et al. "RF-Motion: A Device-Free RF-Based Human Motion Recognition System". In: *Wireless Communications and Mobile Computing* 2021 (2021-03), pp. 1–9. DOI: 10.1155/2021/1497503 (cit. on pp. 1, 21).

[4] H. D. Griffiths and C. J. Baker. *An introduction to passive radar*. Artech House, 2022 (cit. on pp. 4–6).

[5] Wikipedia contributors. *Passive radar — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Passive_radar&oldid=1191100914. [Online; accessed 13-January-2024]. 2023 (cit. on pp. 4, 6).

[6] S. Mercier et al. "Comparison of Correlation-Based OFDM Radar Receivers". In: *IEEE Transactions on Aerospace and Electronic Systems* 56.6 (2020), pp. 4796–4813. DOI: 10.1109/TAES.2020.3003704 (cit. on p. 8).

[7] J. Zheng et al. "The study of RSSI in wireless sensor networks". In: *2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*. Atlantis Press. 2016, pp. 207–209 (cit. on p. 9).

[8] M. A. A. Al-qaness et al. "Channel State Information from Pure Communication to Sense and Track Human Motion: A Survey". In: *Sensors* 19.15 (2019). ISSN: 1424-8220. DOI: 10.3390/s19153329. URL: https://www.mdpi.com/1424-8220/19/15/3329 (cit. on p. 9).

[9] A. Florio, G. Avitabile, and G. Coviello. "Multiple Source Angle of Arrival Estimation Through Phase Interferometry". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 69.3 (2022), pp. 674–678. DOI: 10.1109/TCSII.2022.3141247 (cit. on p. 10).

[10] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. "A review of feature selection methods on synthetic data". In: *Knowledge and Information Systems* 34 (2013), pp. 483–519. URL: https://api.semanticscholar.org/CorpusID:3092608 (cit. on p. 14).

[11] G. Chandrashekar and F. Sahin. "A survey on feature selection methods". In: *Computers Electrical Engineering* 40.1 (2014). 40th-year commemorative issue, pp. 16–28. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2013.11.024. URL: https://www.sciencedirect.com/science/article/pii/S0045790613003066 (cit. on p. 14).

[12] N. Sánchez-Maroño, A. Alonso-Betanzos, and M. Tombilla-Sanromán. "Filter Methods for Feature Selection – A Comparative Study". In: 2007-12, pp. 178–187. ISBN: 978-3-540-77225-5. DOI: 10.1007/978-3-540-77226-2_19 (cit. on p. 14).

[13] A. Zheng and A. Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. 1st. O'Reilly Media, Inc., 2018. ISBN: 1491953241 (cit. on p. 14).

[14] S. Indolia et al. "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach". In: *Procedia Computer Science* 132 (2018). International Conference on Computational Intelligence and Data Science, pp. 679–688. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.069. URL: https://www.sciencedirect.com/science/article/pii/S1877050918308019 (cit. on p. 15).

[15] R. Yamashita et al. "Convolutional neural networks: an overview and application in radiology". In: *Insights into imaging* 9 (2018), pp. 611–629 (cit. on p. 15).

[16] V. H. Phung and E. J. Rhee. "A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets". In: *Applied Sciences* 9.21 (2019). ISSN: 2076-3417. DOI: 10.3390/app9214500. URL: https://www.mdpi.com/2076-3417/9/21/4500 (cit. on p. 15).

[17] T. Wang et al. "Deep learning for wireless physical layer: Opportunities and challenges". In: *China Communications* 14.11 (2017), pp. 92–111. DOI: 10.1109/CC.2017.8233654 (cit. on p. 15).

[18] U. Michelucci. "An Introduction to Autoencoders". In: *CoRR* abs/2201.03898 (2022). arXiv: 2201.03898. URL: https://arxiv.org/abs/2201.03898 (cit. on p. 17).

[19] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016 (cit. on p. 17).

[20] L. Rabiner and B. Juang. "An introduction to hidden Markov models". In: *IEEE ASSP Magazine* 3.1 (1986), pp. 4–16. DOI: 10.1109/MASSP.1986.1165342 (cit. on p. 17).

[21] B. Everitt et al. *Cluster Analysis, 5th Edition*. John Wiley & Sons, 2011. URL: https://books.google.pt/books?id=3RFPzQEACAAJ (cit. on p. 17).

[22] K. Taunk et al. "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification". In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. 2019, pp. 1255–1260. DOI: 10.1109/ICCS45141.2019.9065747 (cit. on p. 18).

[23] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000 (cit. on p. 18).

[24] A. C. Braun, U. Weidner, and S. Hinz. "Support vector machines, import vector machines and relevance vector machines for hyperspectral classification — A comparison". In: *2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. 2011, pp. 1–4. DOI: 10.1109/WHISPERS.2011.6080861 (cit. on p. 18).

[25] Y. Yang, J. Li, and Y. Yang. "The research of the fast SVM classifier method". In: *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. 2015, pp. 121–124. DOI: 10.1109/ICCWAMTIP.2015.7493959 (cit. on p. 18).

[26] Y. Ren et al. "3D Human Pose Estimation Using WiFi Signals". In: *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. SenSys '21. Coimbra, Portugal: Association for Computing Machinery, 2021, 363–364. ISBN: 9781450390972. DOI: 10.1145/3485730.3492871. URL: https://doi.org/10.1145/3485730.3492871 (cit. on p. 20).

[27] J. Liu et al. "Multi-Target Intense Human Motion Analysis and Detection Using Channel State Information". In: *Sensors* 18.10 (2018). ISSN: 1424-8220. DOI: 10.3390/s18103379. URL: https://www.mdpi.com/1424-8220/18/10/3379 (cit. on p. 20).

[28] A. Alanis et al. *3D Gesture Recognition Through RF Sensing*. Tech. rep. MSR-TR-2014-81. 2014. URL: https://www.microsoft.com/en-us/research/publication/3d-gesture-recognition-through-rf-sensing/ (cit. on p. 20).

[29] S. Iqbal et al. "Indoor Motion Classification Using Passive RF Sensing Incorporating Deep Learning". In: *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. 2018, pp. 1–5. DOI: 10.1109/VTCSpring.2018.8417859 (cit. on p. 20).

[30] Y. Liu et al. "Harvesting Ambient RF for Presence Detection Through Deep Learning". In: *CoRR* abs/2002.05770 (2020). arXiv: 2002.05770. URL: https://arxiv.org/abs/2002.05770 (cit. on p. 21).

[31] Y. Wang, K. Wu, and L. M. Ni. "WiFall: Device-Free Fall Detection by Wireless Networks". In: *IEEE Transactions on Mobile Computing* 16.2 (2017), pp. 581–594. DOI: 10.1109/TMC.2016.2557792 (cit. on p. 21).

[32] Y. Xie et al. "An Improved K-Nearest-Neighbor Indoor Localization Method Based on Spearman Distance". In: *IEEE Signal Processing Letters* 23.3 (2016), pp. 351–355. DOI: 10.1109/LSP.2016.2519607 (cit. on p. 21).

[33] J. Pereira, E. Casmin, and R. Oliveira. "A Passive RF Testbed for Human Posture Classification in FM Radio Bands". In: *Sensors* 23.23 (2023). ISSN: 1424-8220. DOI: 10.3390/s23239563. URL: https://www.mdpi.com/1424-8220/23/23/9563 (cit. on p. 21).

[34] URL: https://www.nuand.com/bladerf-2-0-micro/ (cit. on p. 23).

[35] URL: https://www.nuand.com/product/bladerf-xa4/ (cit. on p. 24).

[36] G. Radio. *GNU Radio- The Free Open Source Radio Ecosystem. 2022*. https://www.gnuradio.org/. [Accessed 03-06-2024] (cit. on p. 28).

[37] A. Csete. *Gqrx SDR 2013;Open source software defined radio by Alexandru Csete, howpublished = https://www.gqrx.dk/, year = , note = [Accessed 03-06-2024],* (cit. on p. 28).

[38] argilo. *pybombs*. 2022. URL: https://github.com/gnuradio/pybombs (cit. on p. 28).

Appendix 1

In this appendix, we present the histograms for all 10 latent features extracted using the VAE using the complete band.

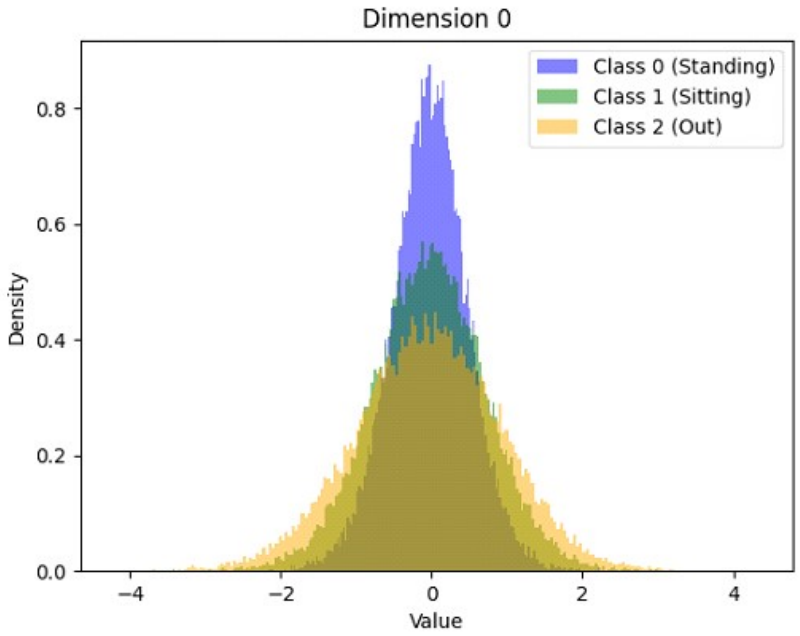

Figure A.1: Histogram of the feature 0.

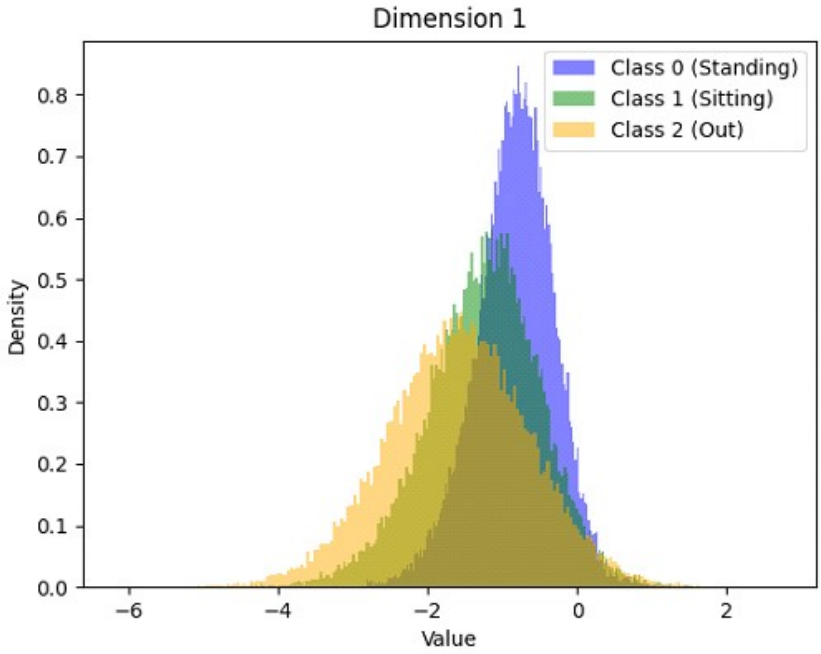

Figure A.2: Histogram of the feature 1.

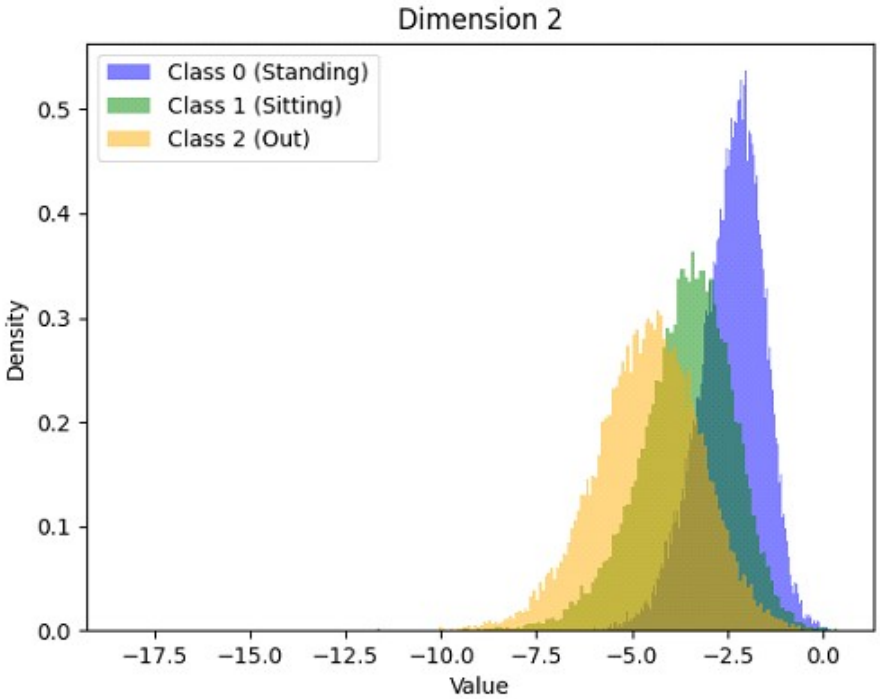

Figure A.3: Histogram of the feature 2.

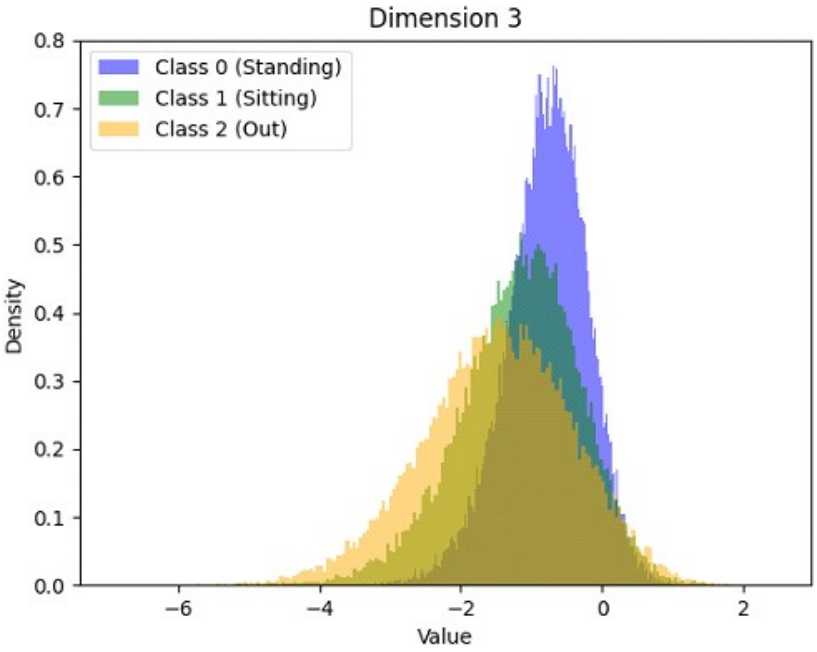

Figure A.4: Histogram of the feature 3.

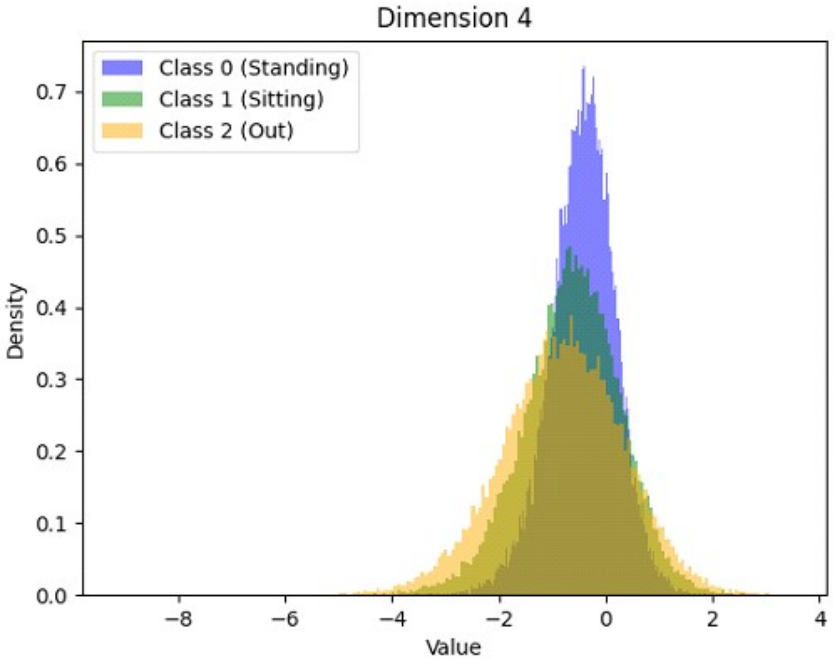

Figure A.5: Histogram of the feature 4.

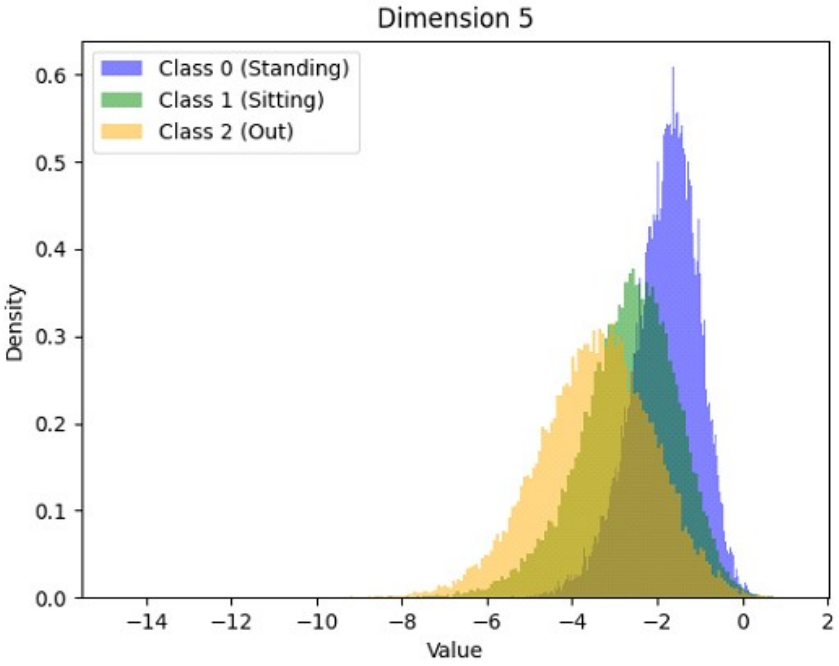

Figure A.6: Histogram of the feature 5.

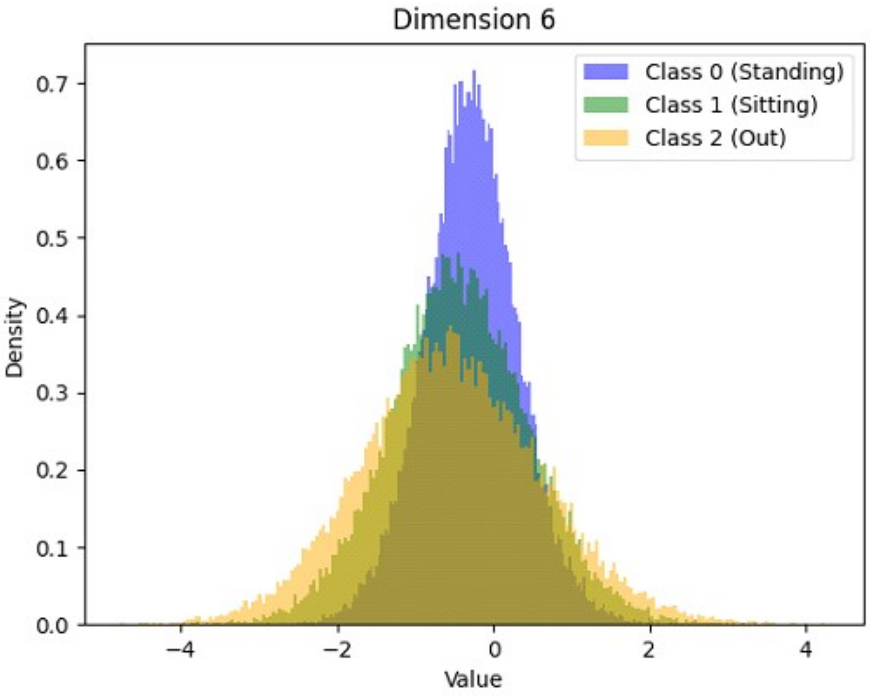

Figure A.7: Histogram of the feature 6.

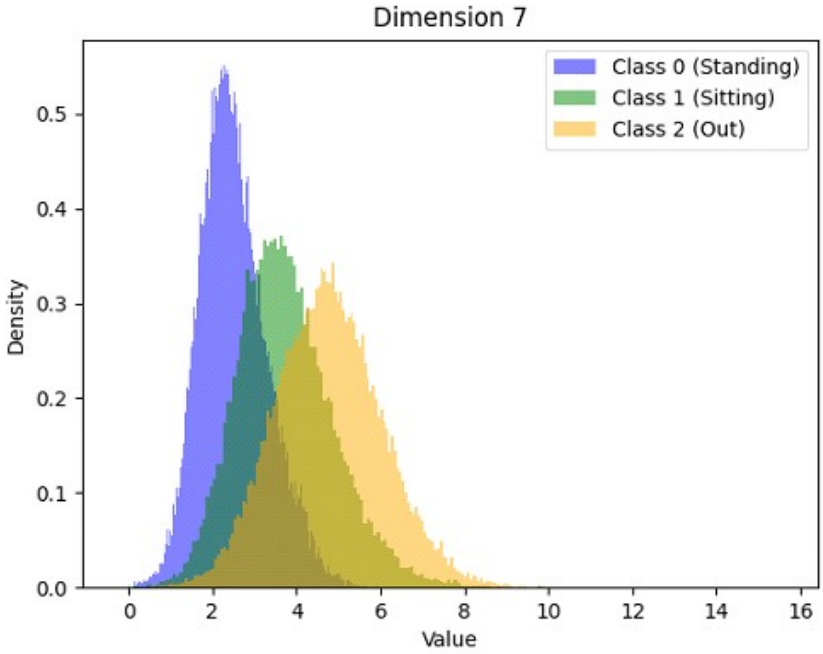

Figure A.8: Histogram of the feature 7.

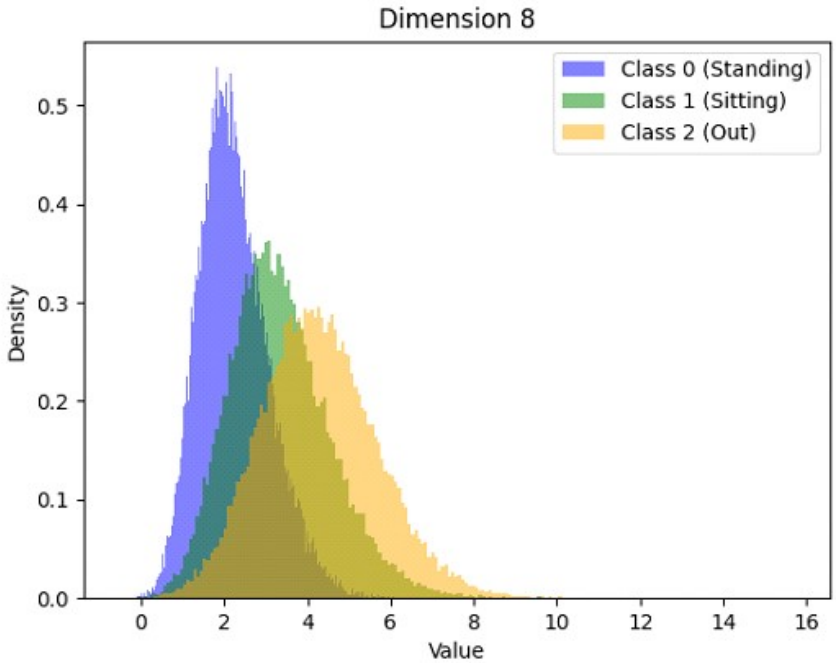

Figure A.9: Histogram of the feature 8.

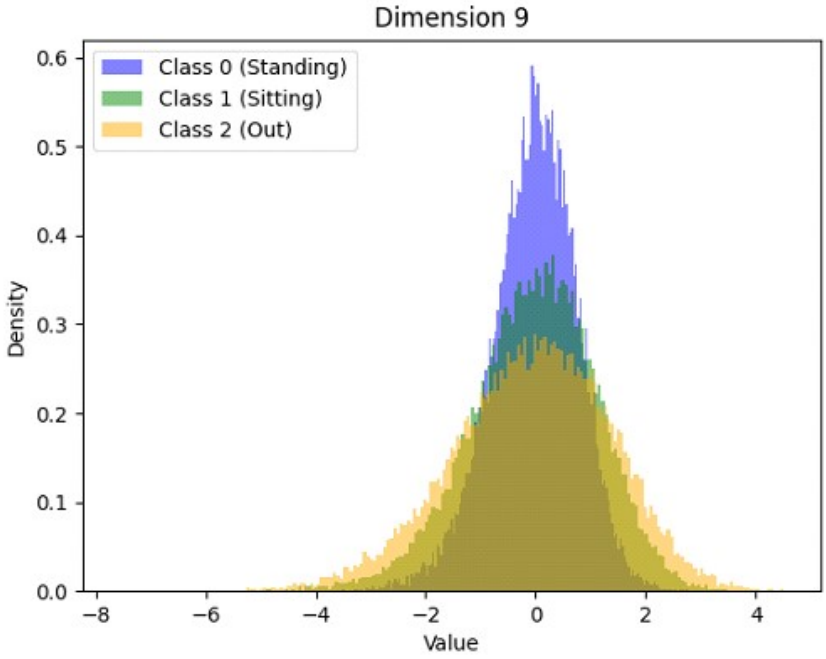

Figure A.10: Histogram of the feature 9.

Appendix 2

In this appendix, we present the histograms for all 25 latent features extracted using the VAE using the complete band.

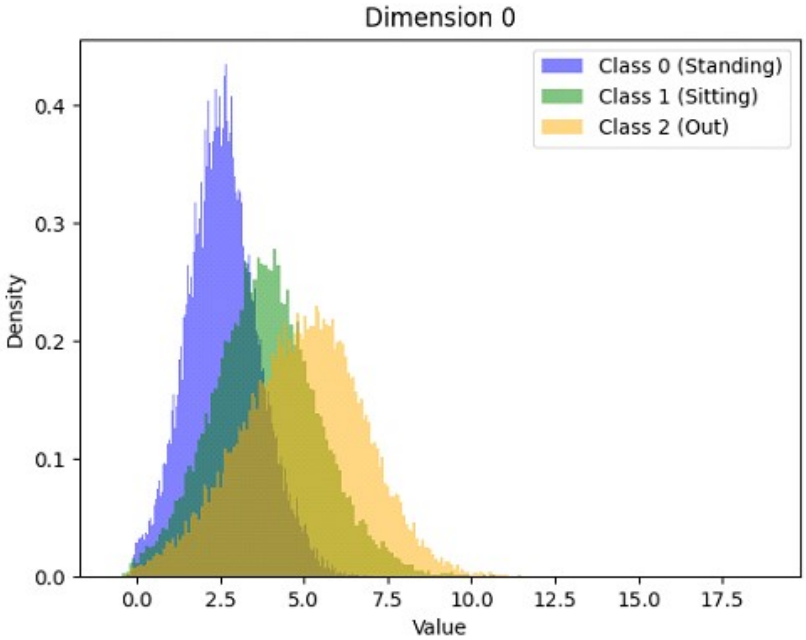

Figure B.1: Histogram of the feature 0.

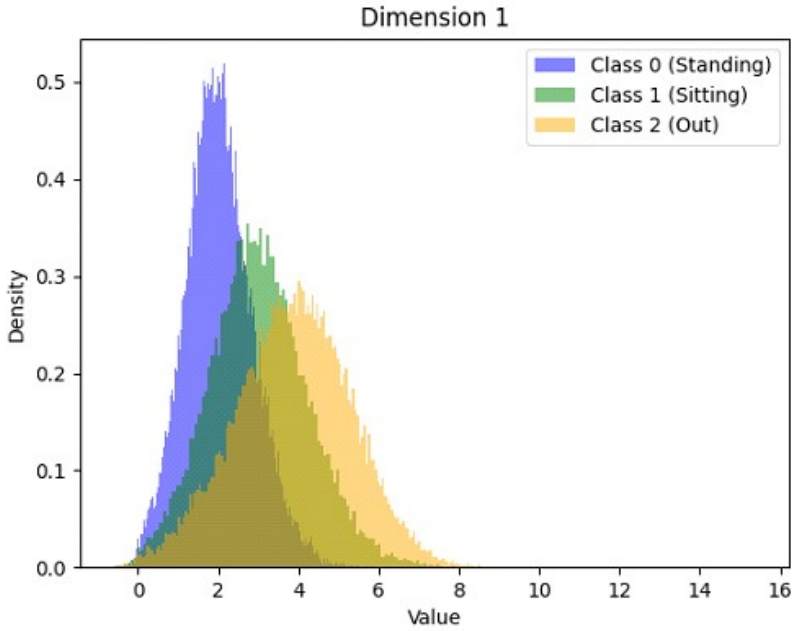

Figure B.2: Histogram of the feature 1.

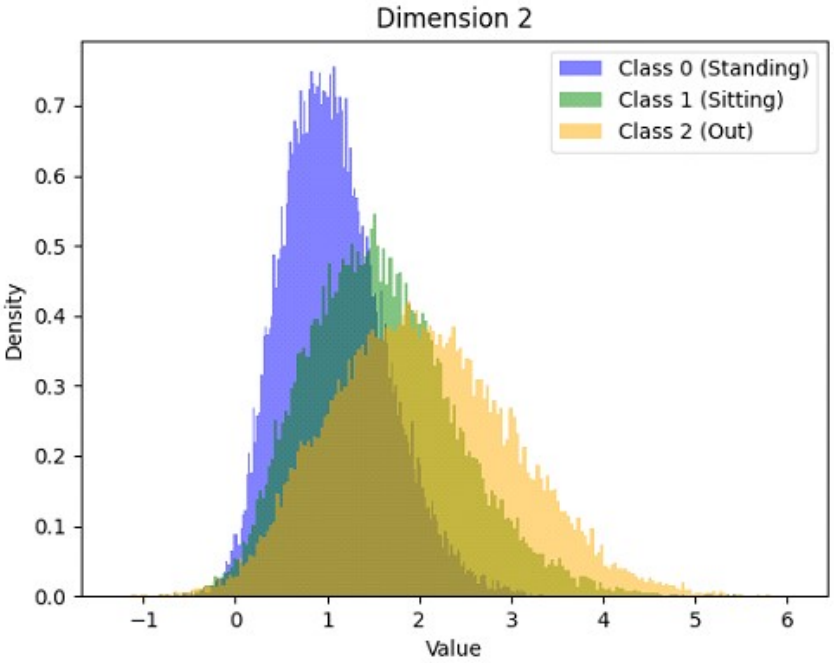

Figure B.3: Histogram of the feature 2.

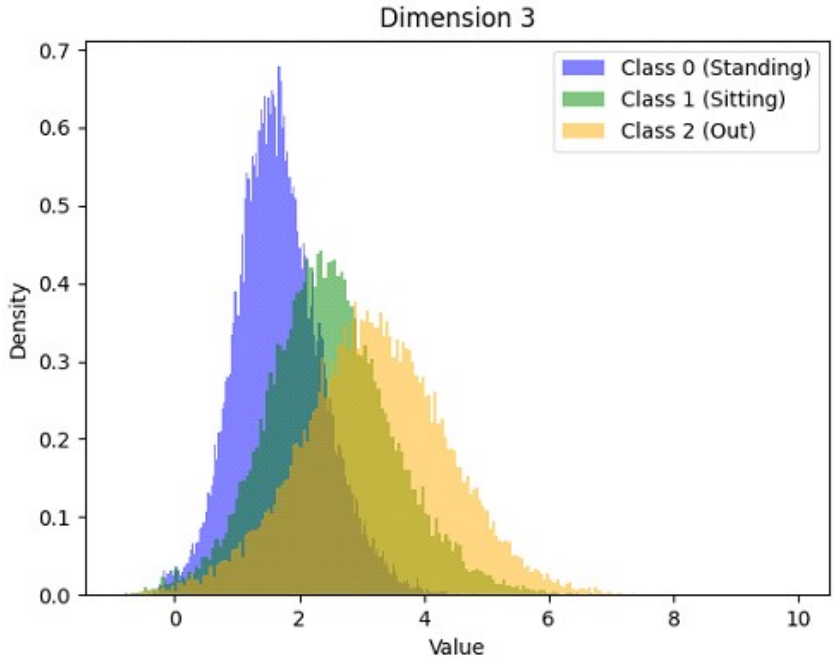

Figure B.4: Histogram of the feature 3.

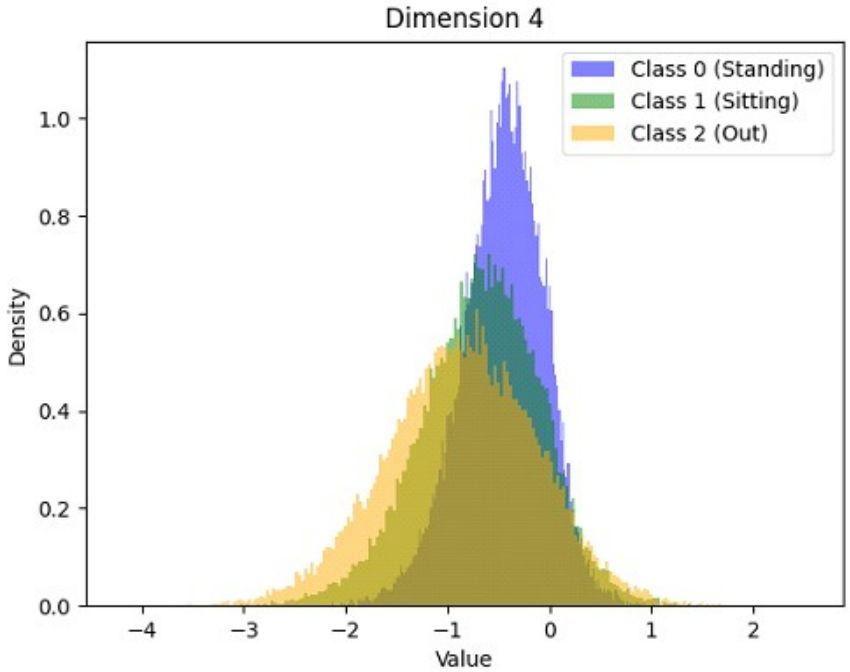

Figure B.5: Histogram of the feature 4.

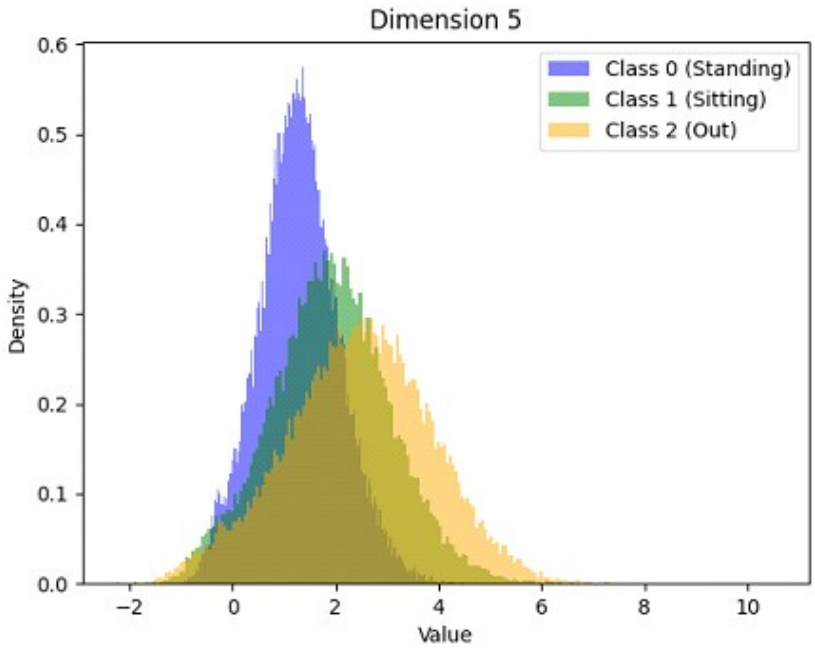

Figure B.6: Histogram of the feature 5.

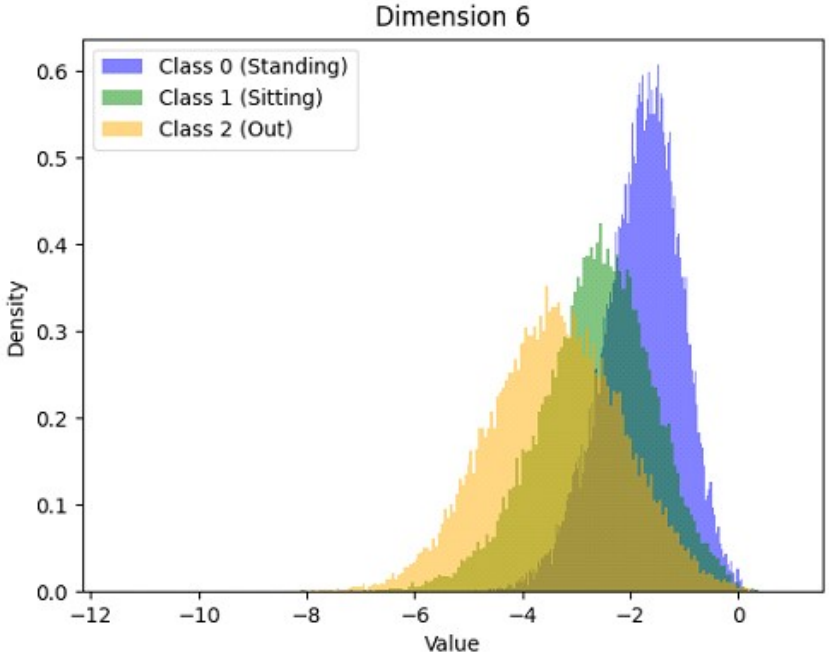

Figure B.7: Histogram of the feature 6.

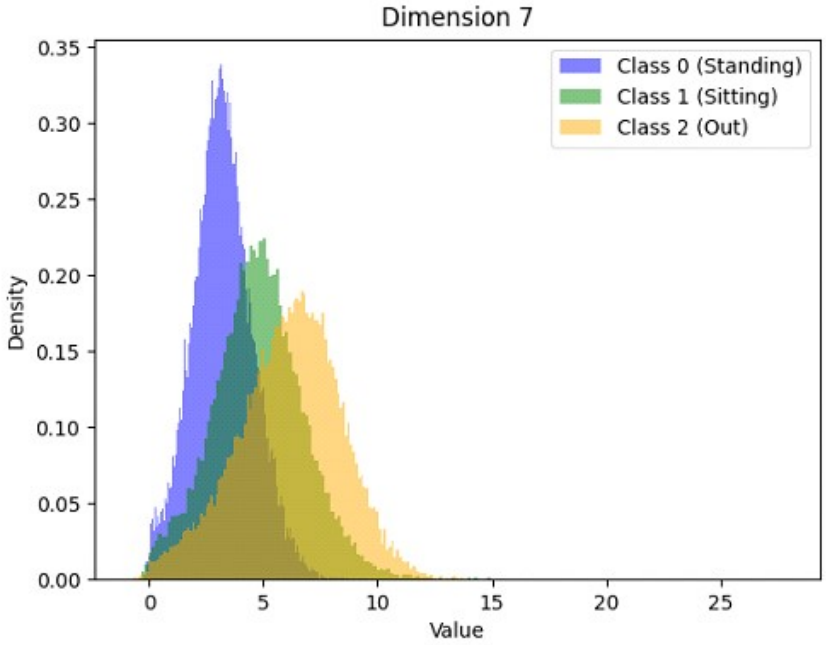

Figure B.8: Histogram of the feature 7.

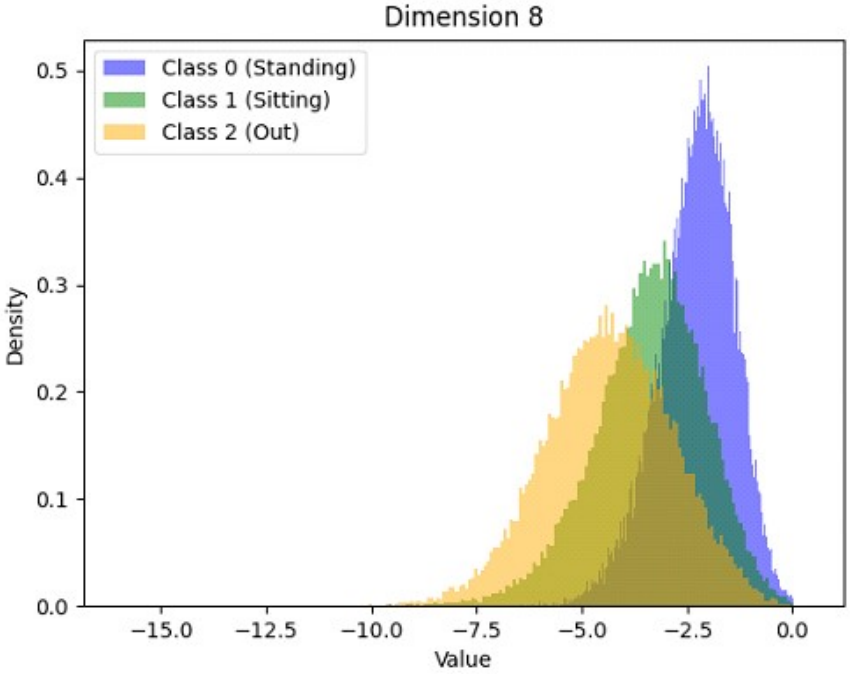

Figure B.9: Histogram of the feature 8.

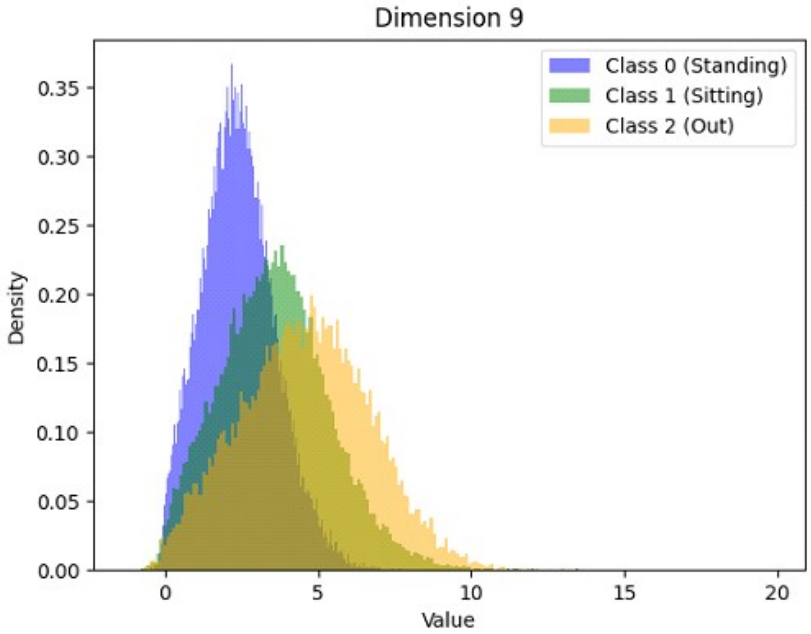

Figure B.10: Histogram of the feature 9.

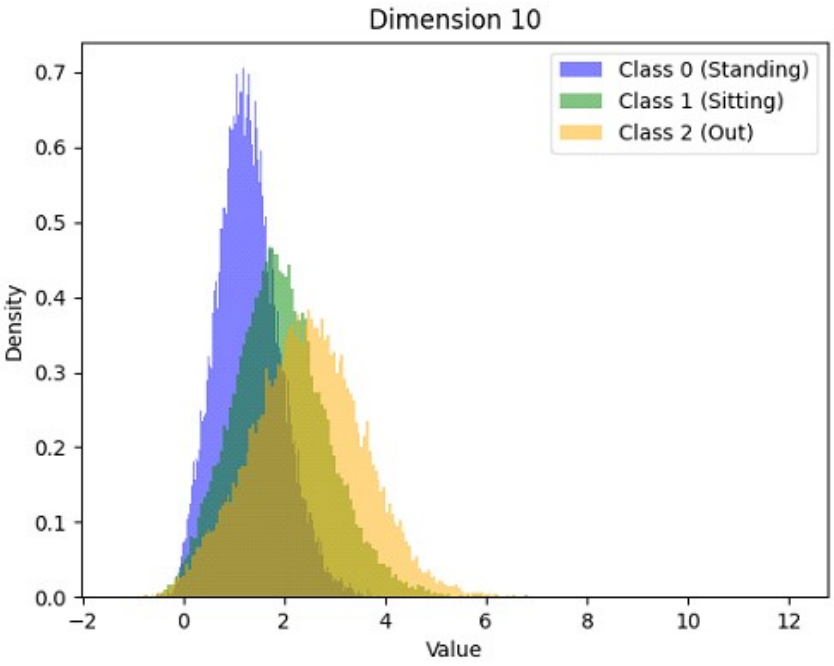

Figure B.11: Histogram of the feature 10.

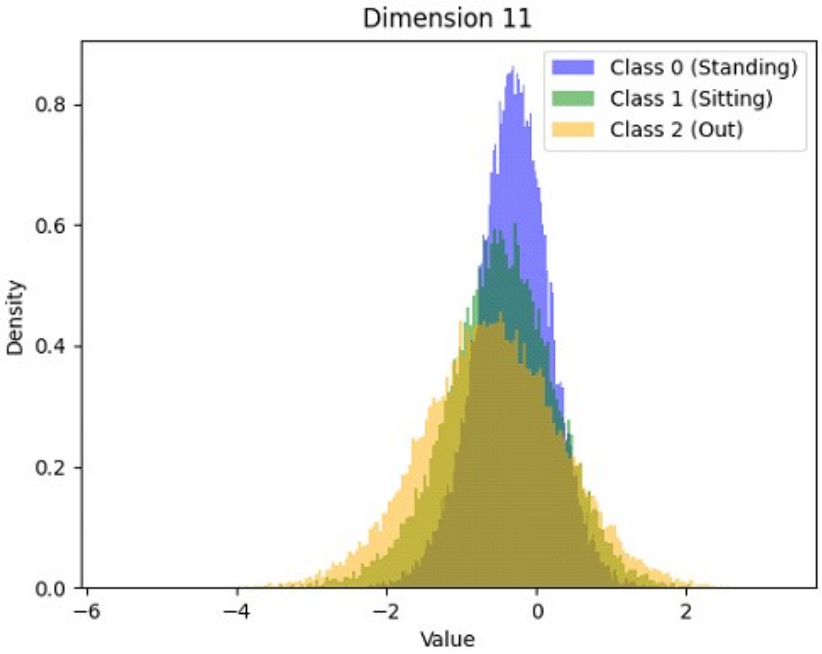

Figure B.12: Histogram of the feature 11.

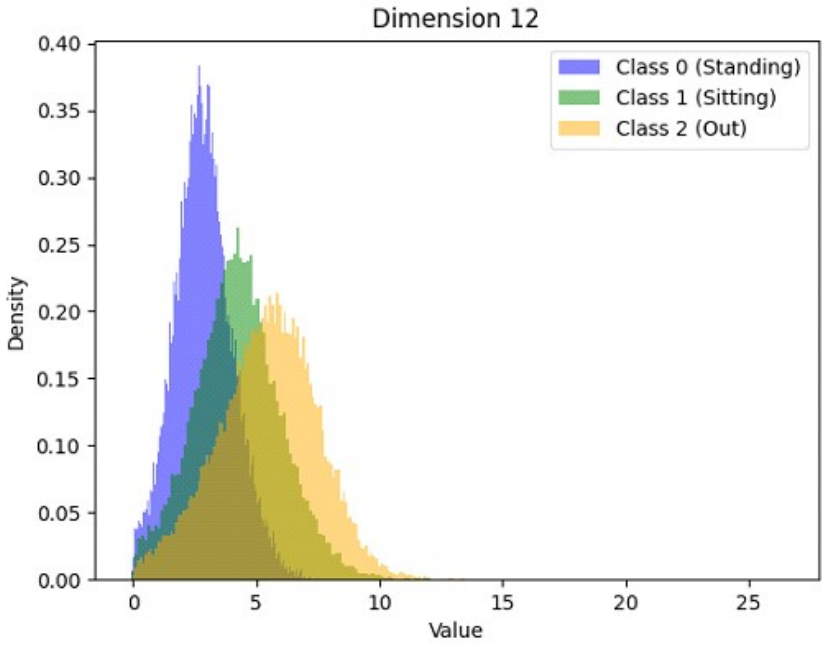

Figure B.13: Histogram of the feature 12.

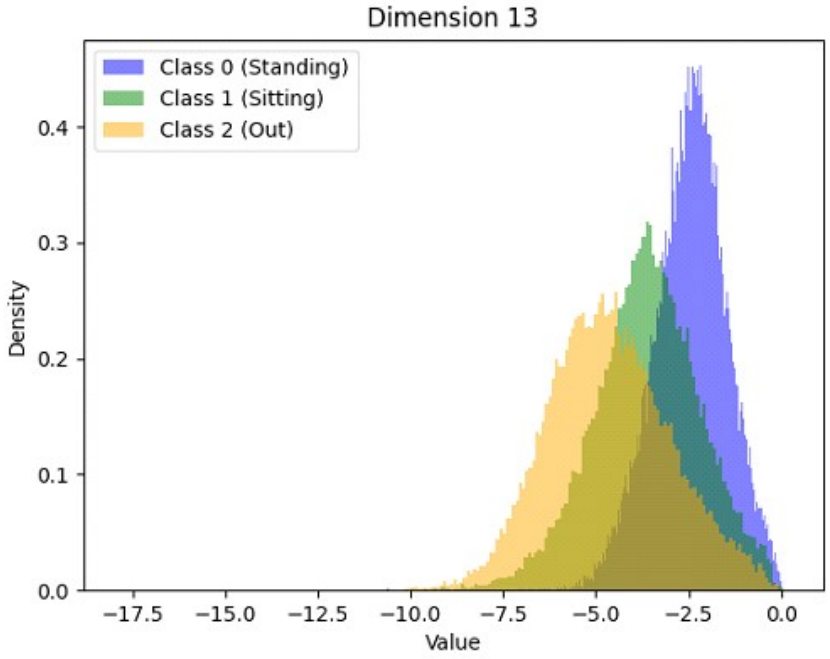

Figure B.14: Histogram of the feature 13.

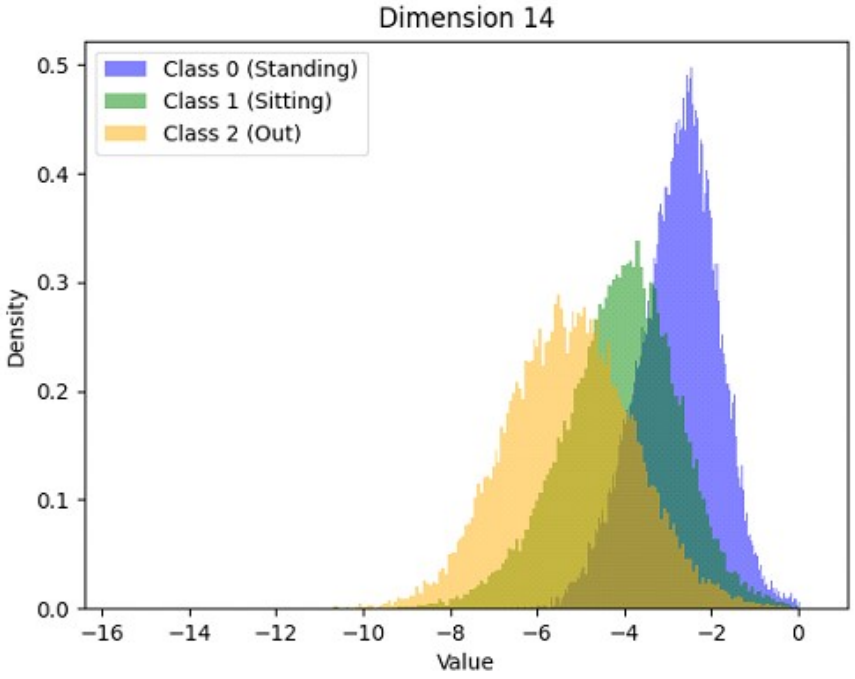

Figure B.15: Histogram of the feature 14.

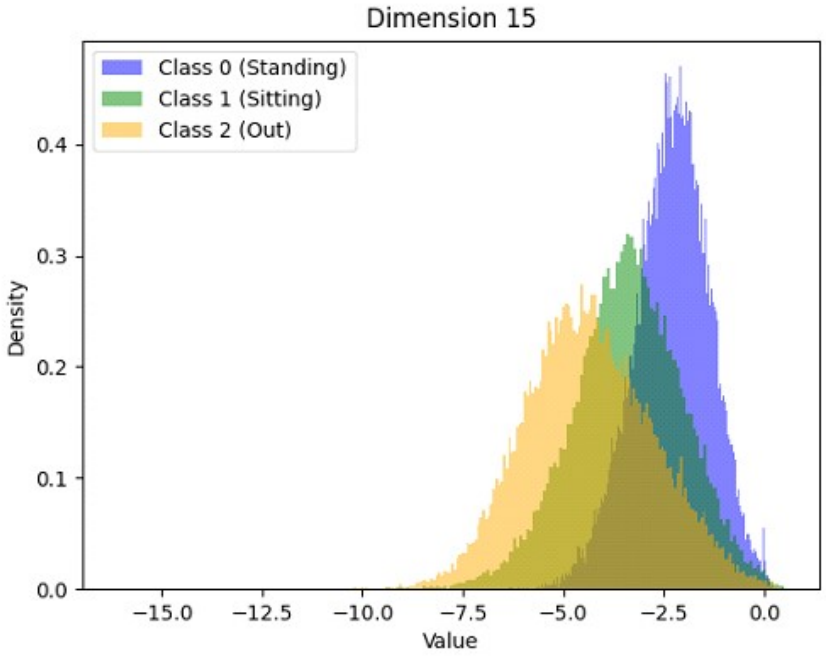

Figure B.16: Histogram of the feature 15.

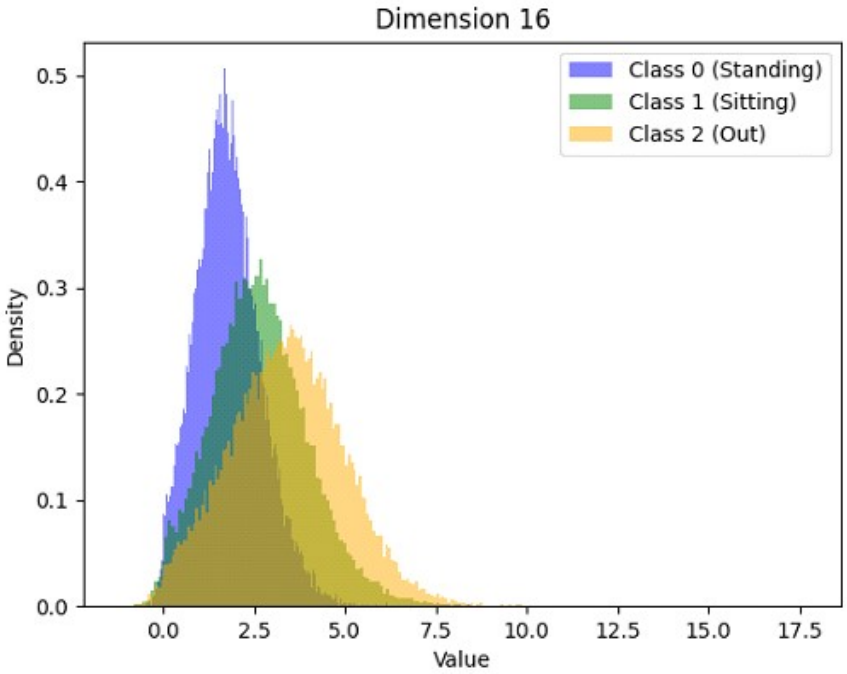

Figure B.17: Histogram of the feature 16.

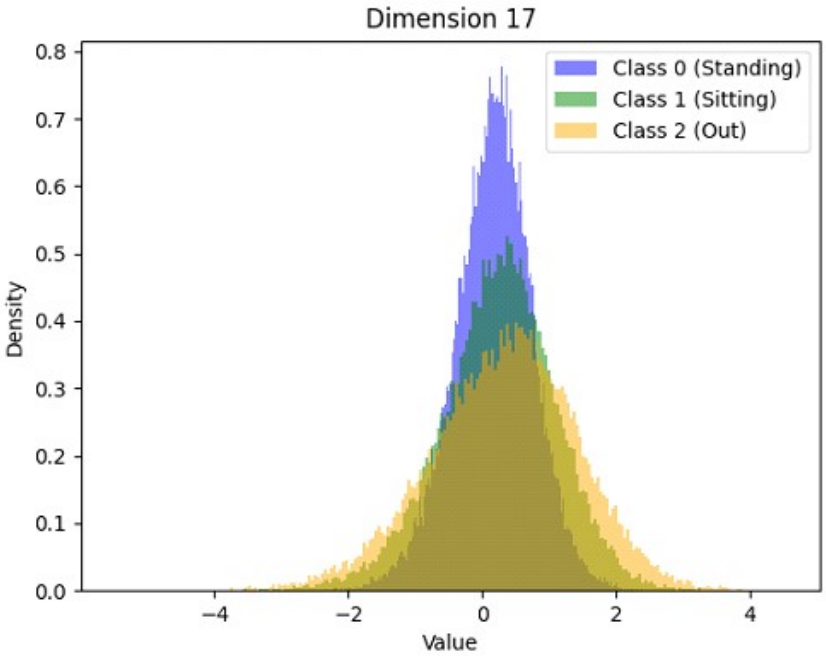

Figure B.18: Histogram of the feature 17.

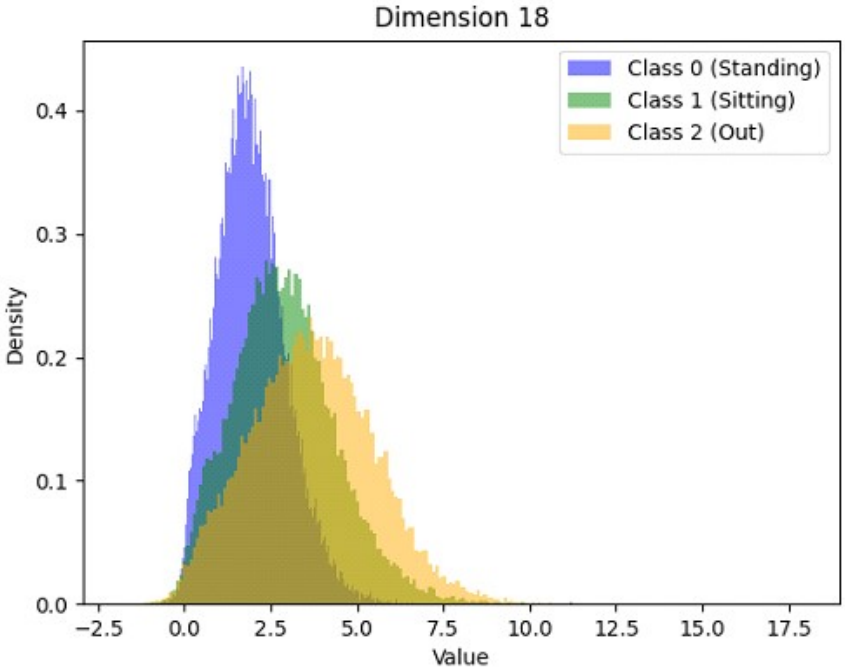

Figure B.19: Histogram of the feature 18.

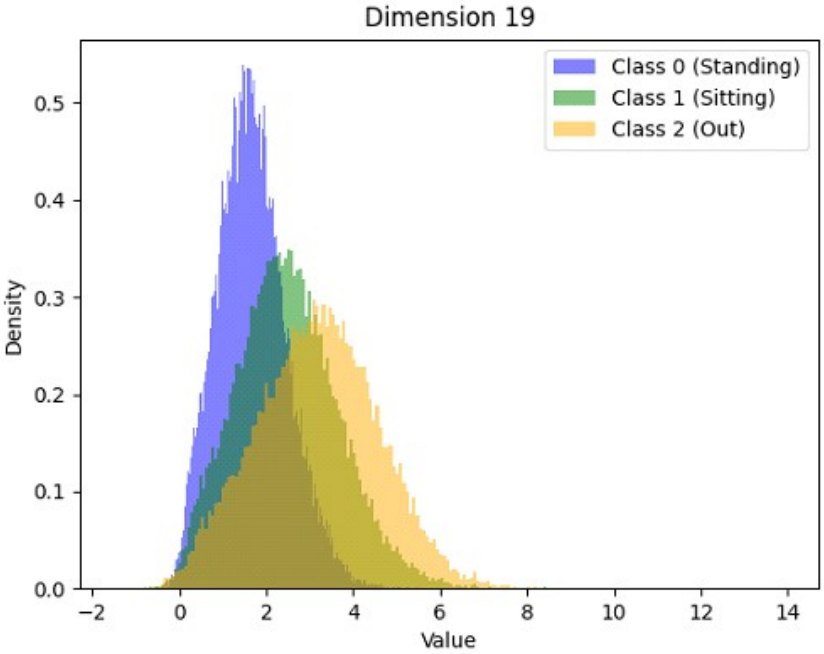

Figure B.20: Histogram of the feature 19.

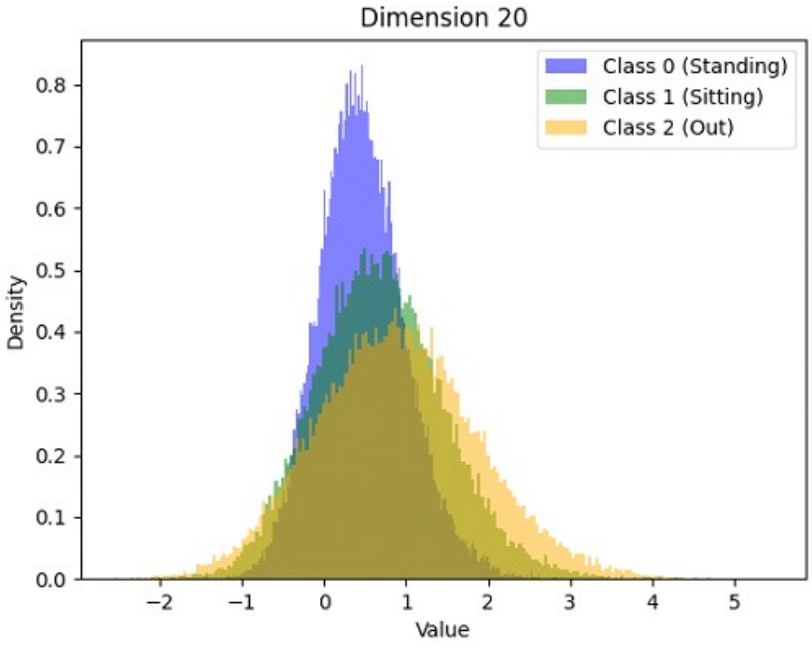

Figure B.21: Histogram of the feature 20.

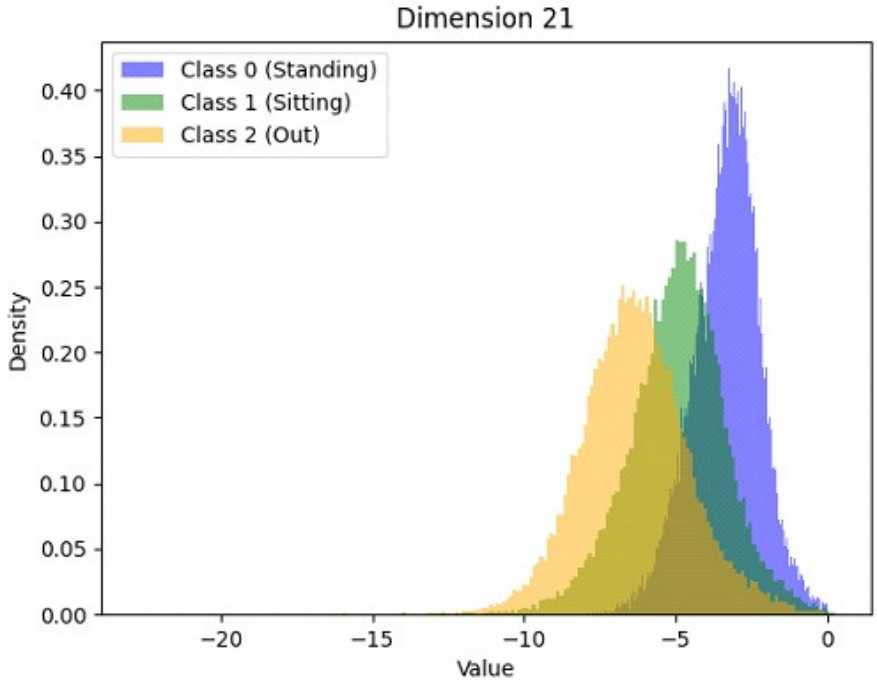

Figure B.22: Histogram of the feature 21.

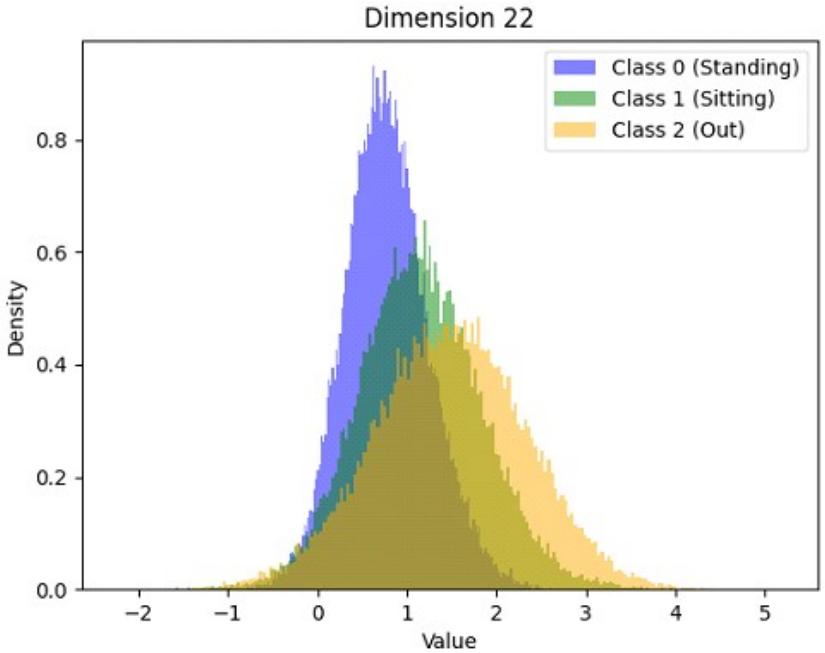

Figure B.23: Histogram of the feature 22.

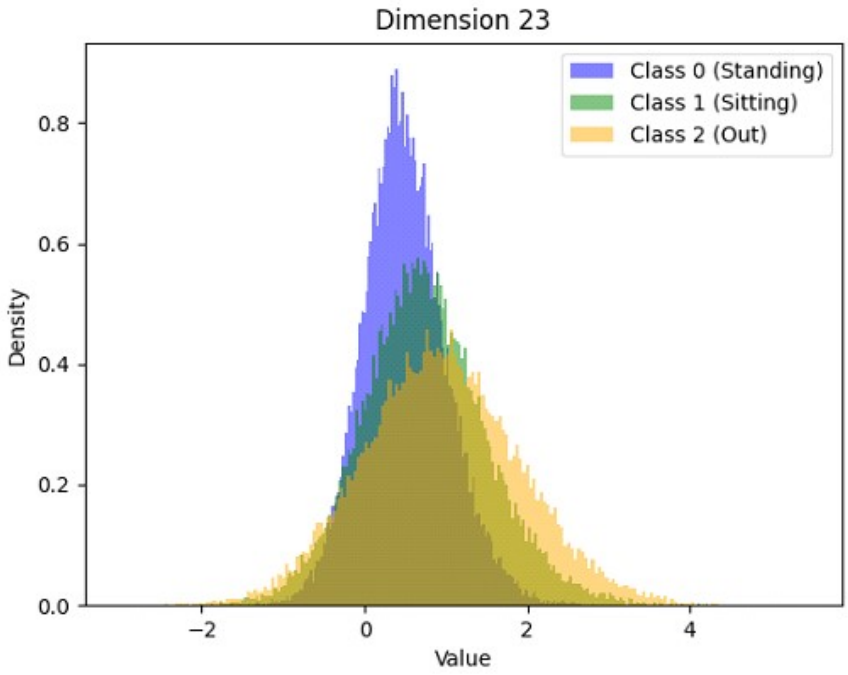

Figure B.24: Histogram of the feature 23.

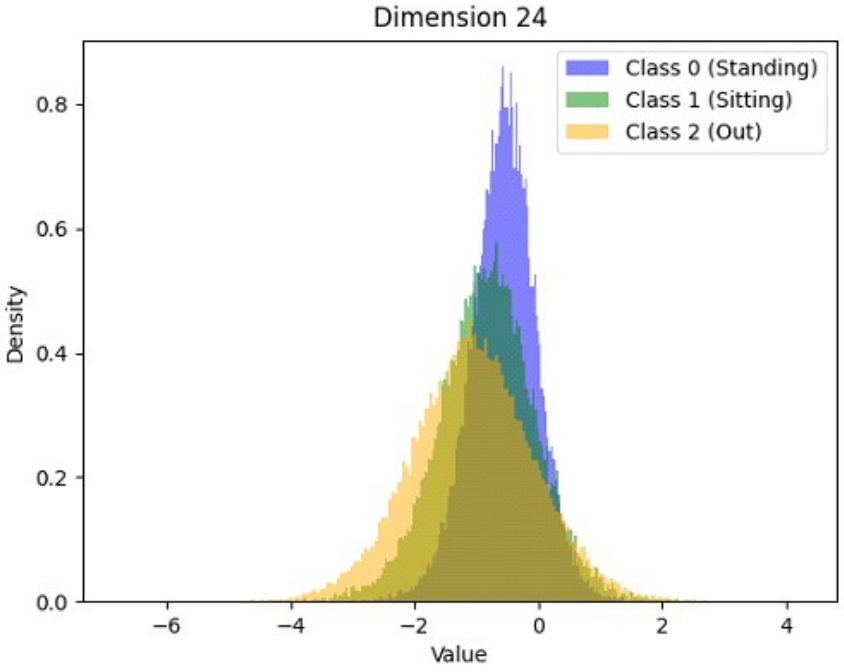

Figure B.25: Histogram of the feature 24.

Appendix 3

In this appendix, we present the histograms for all 50 latent features extracted using the VAE using the complete band.

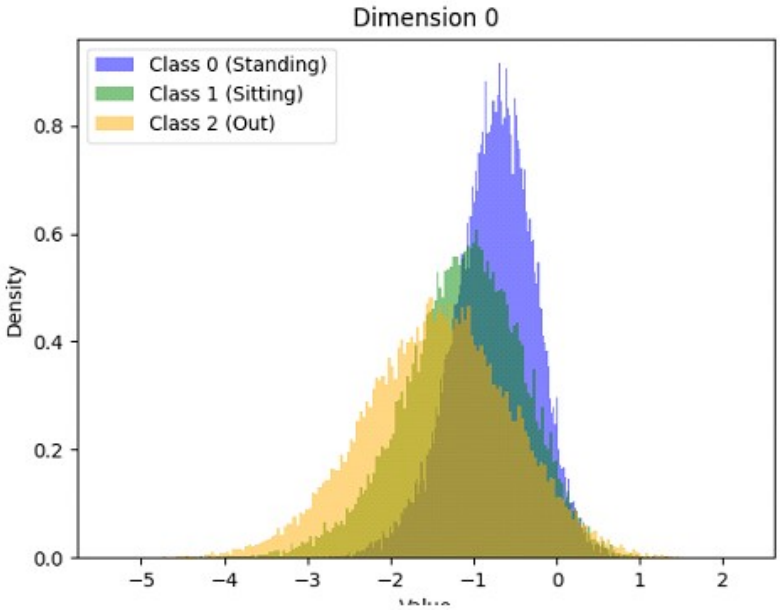

Figure C.1: Histogram of the feature 0.

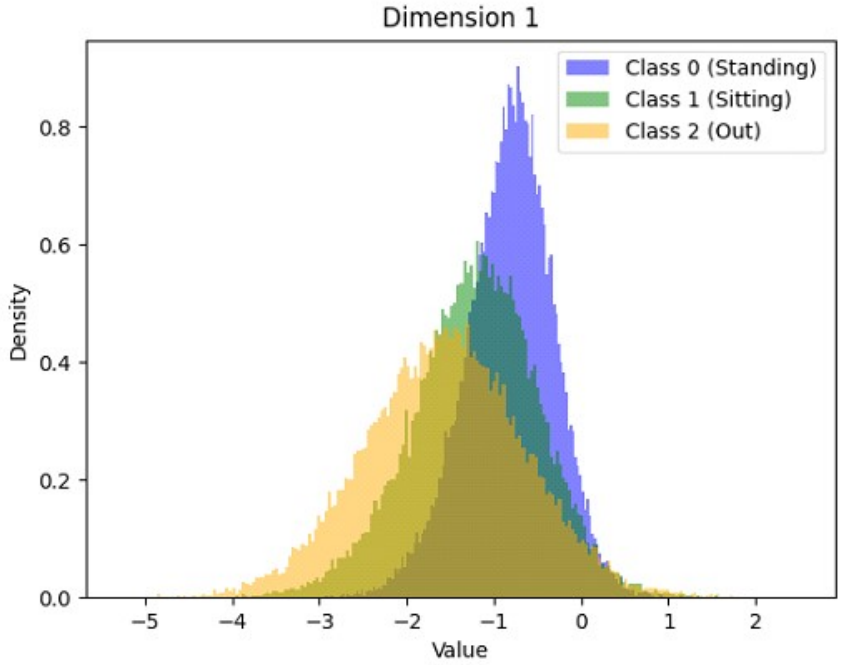

Figure C.2: Histogram of the feature 1.

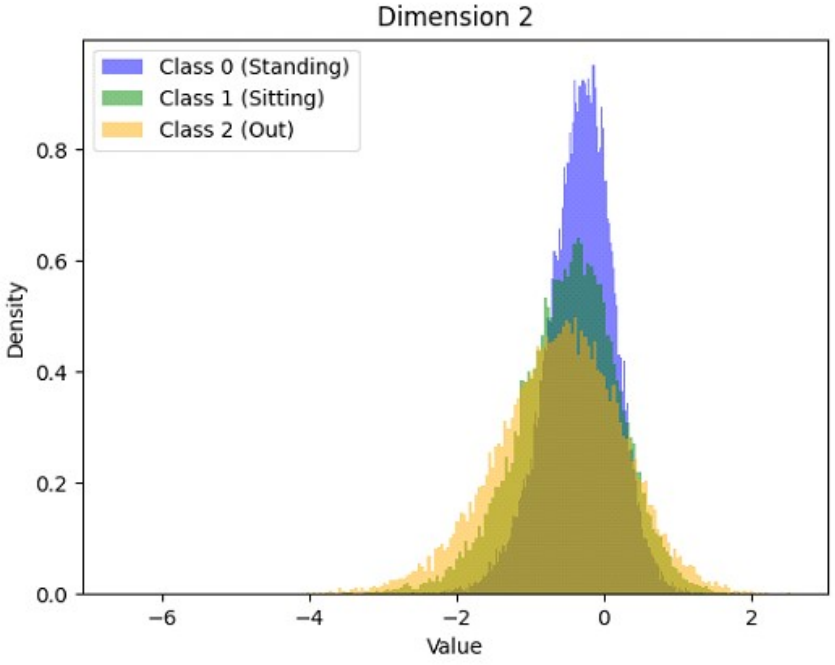

Figure C.3: Histogram of the feature 2.

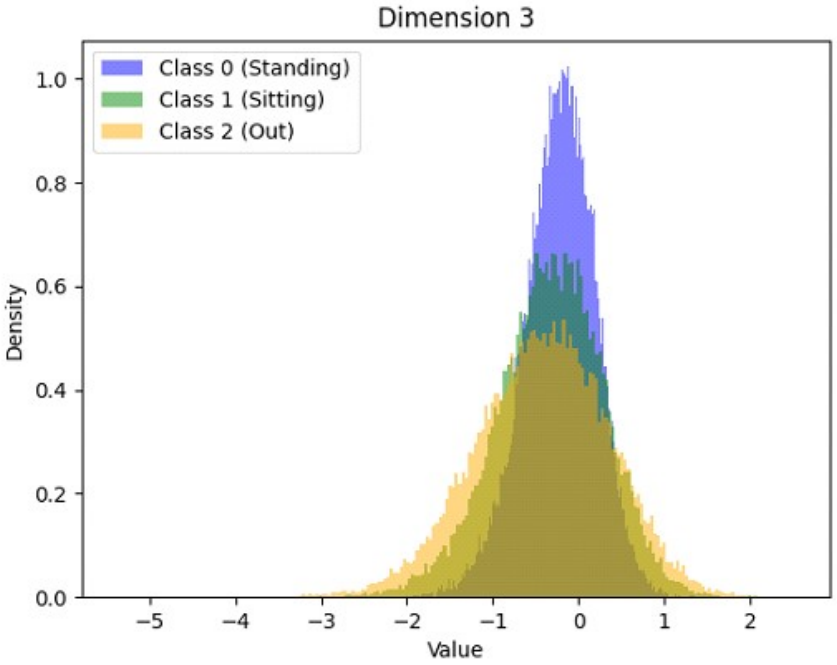

Figure C.4: Histogram of the feature 3.

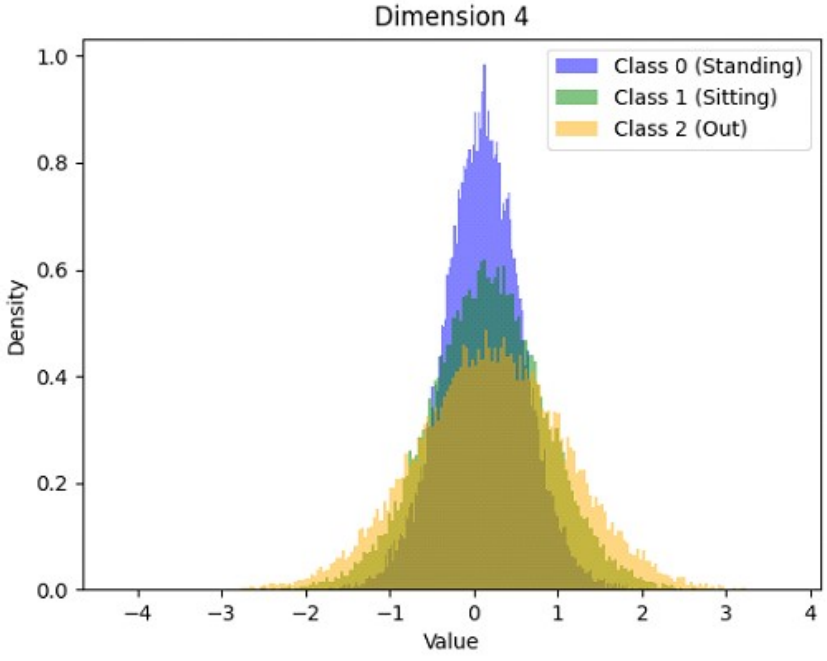

Figure C.5: Histogram of the feature 4.

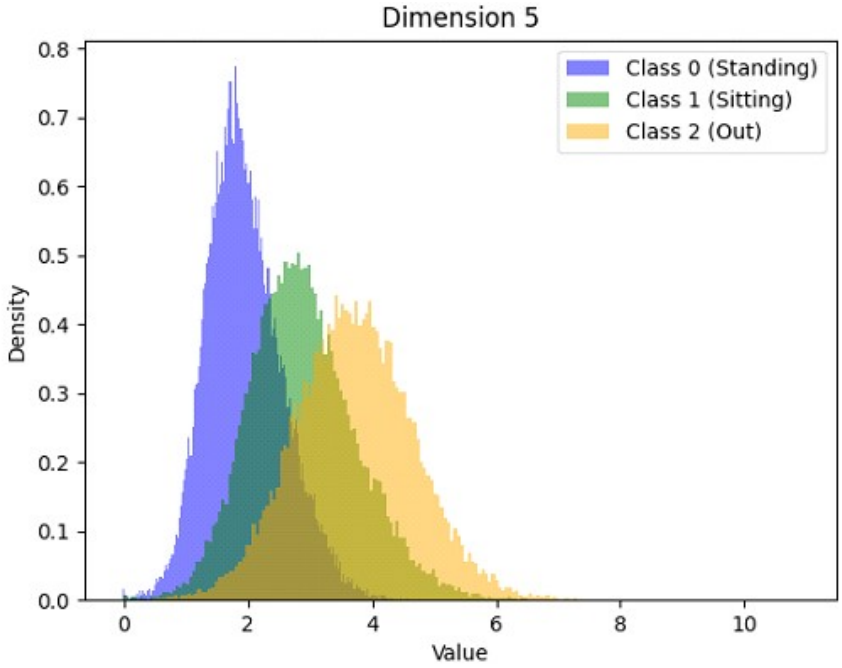

Figure C.6: Histogram of the feature 5.

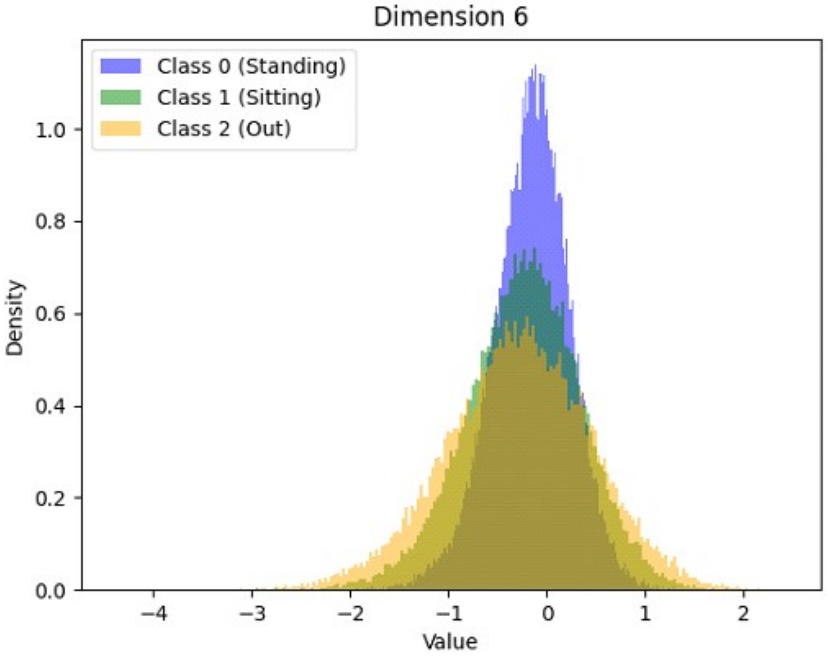

Figure C.7: Histogram of the feature 6.

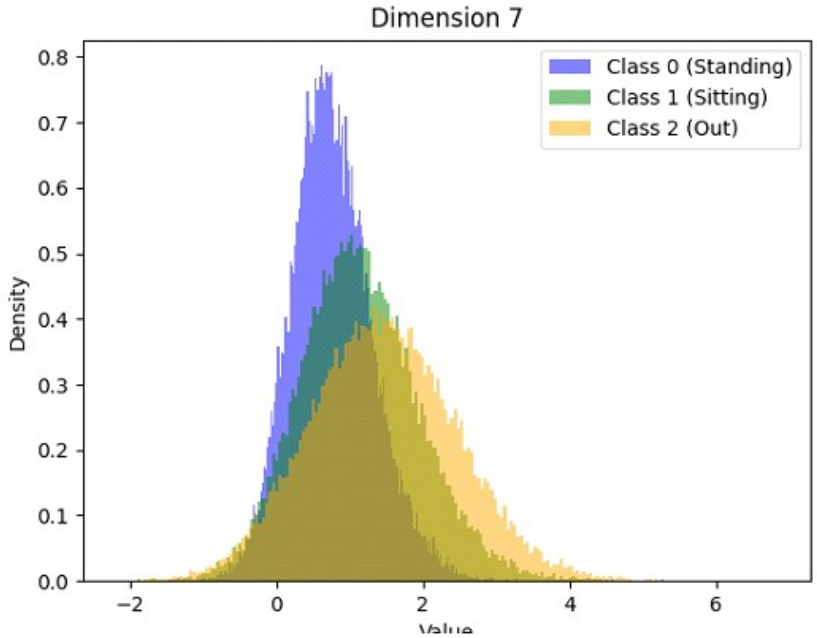

Figure C.8: Histogram of the feature 7.

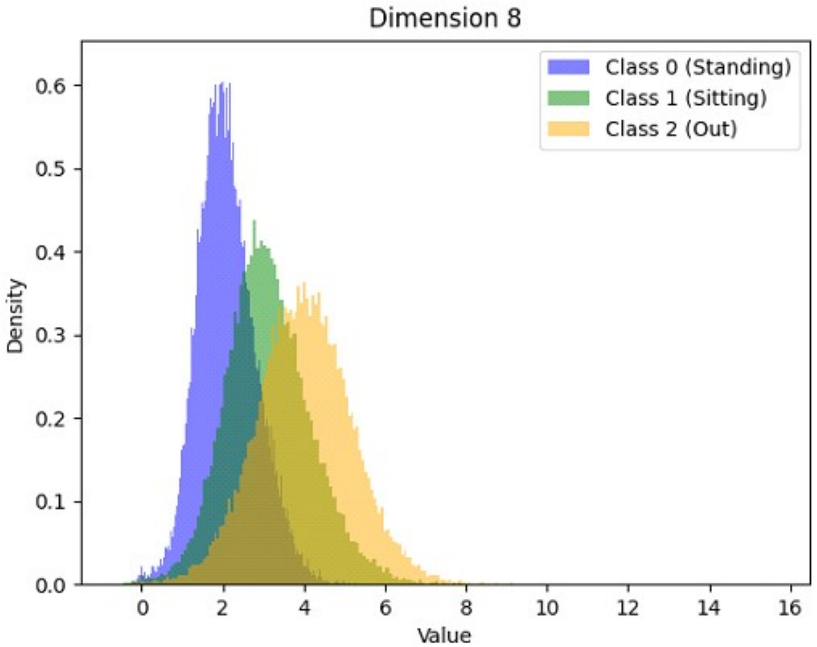

Figure C.9: Histogram of the feature 8.

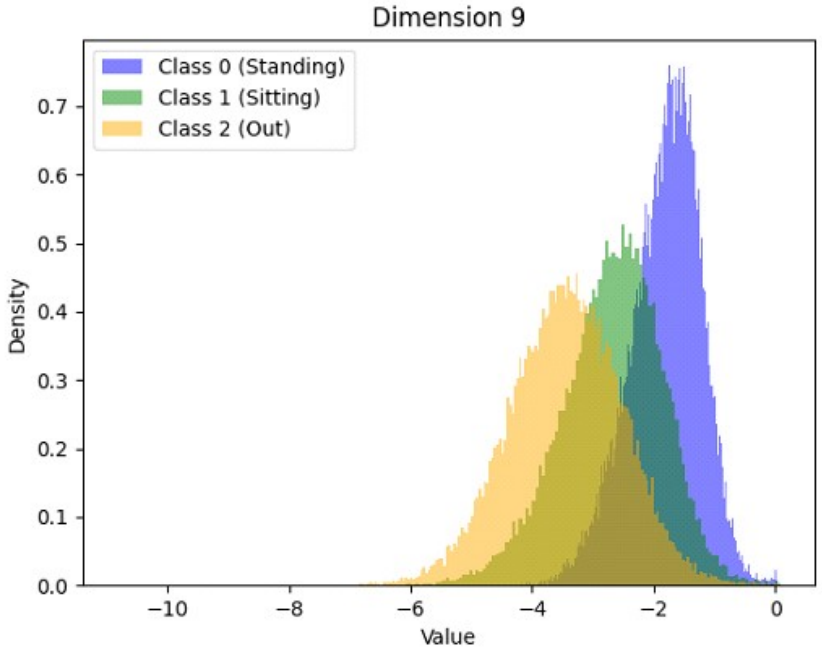

Figure C.10: Histogram of the feature 9.

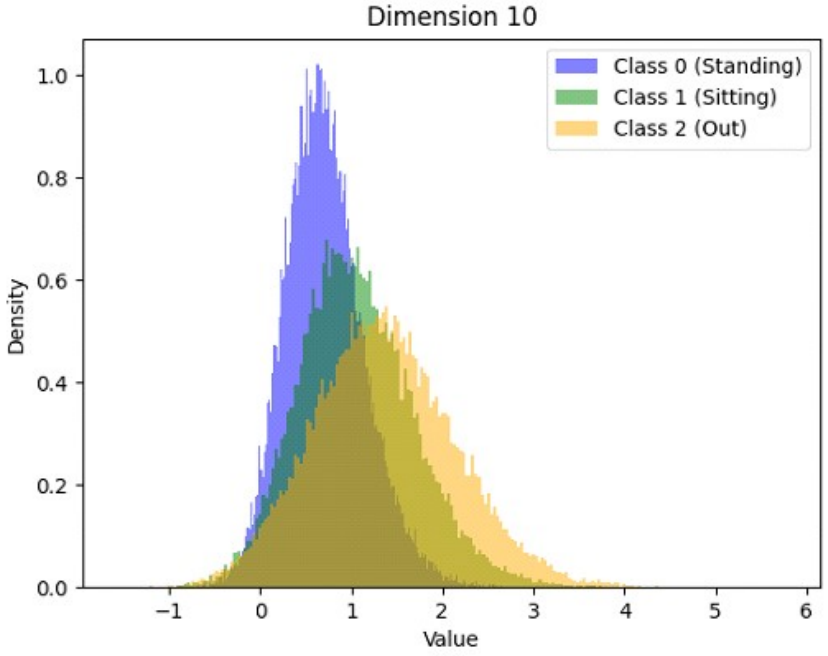

Figure C.11: Histogram of the feature 10.

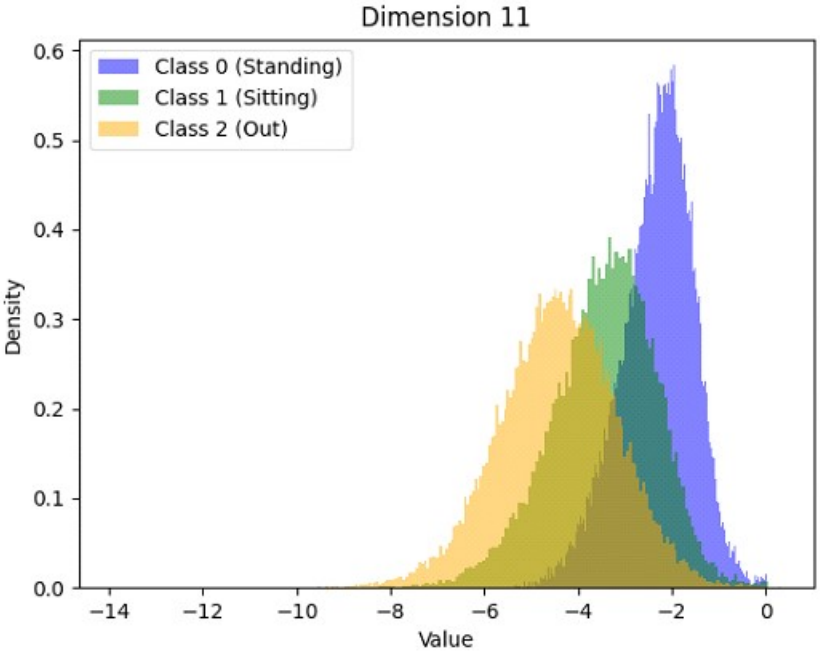

Figure C.12: Histogram of the feature 11.

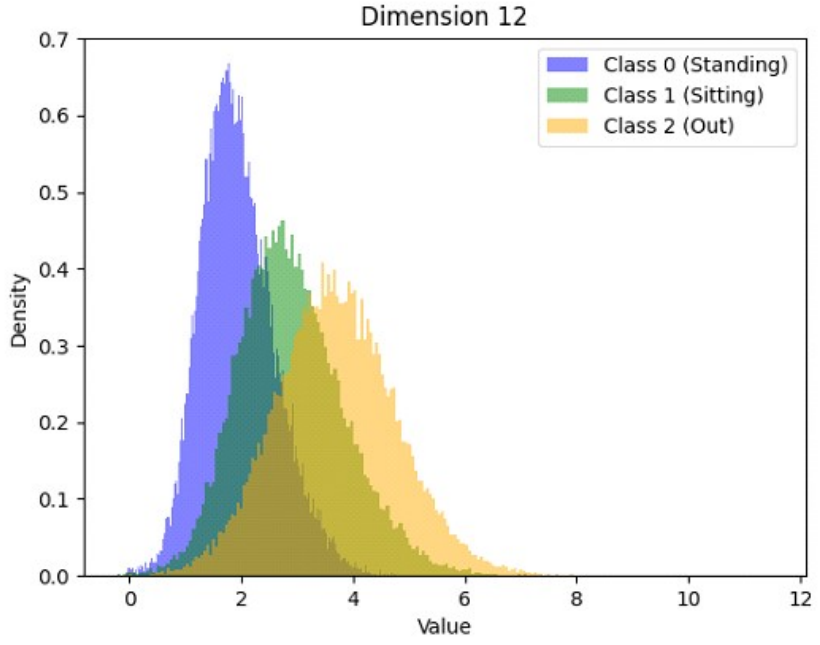

Figure C.13: Histogram of the feature 12.

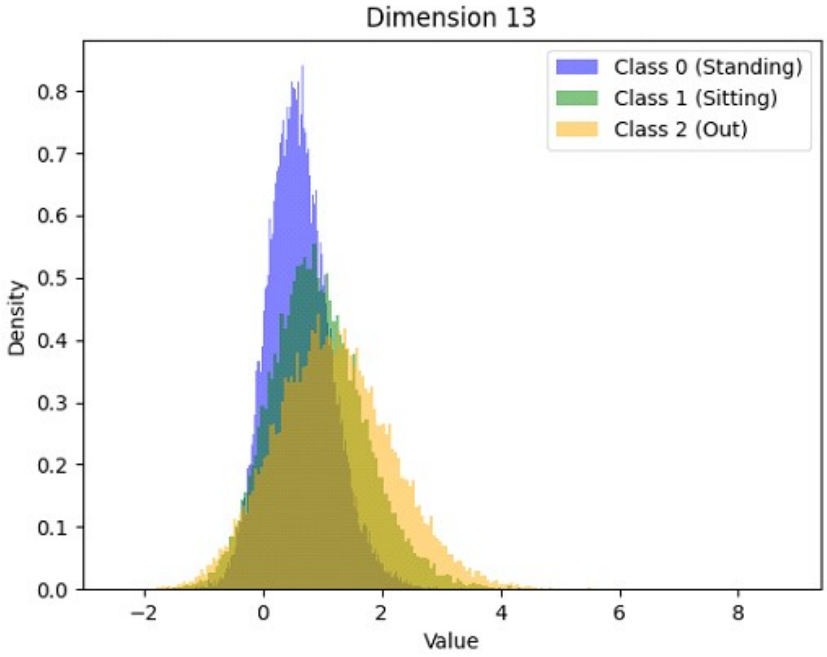

Figure C.14: Histogram of the feature 13.

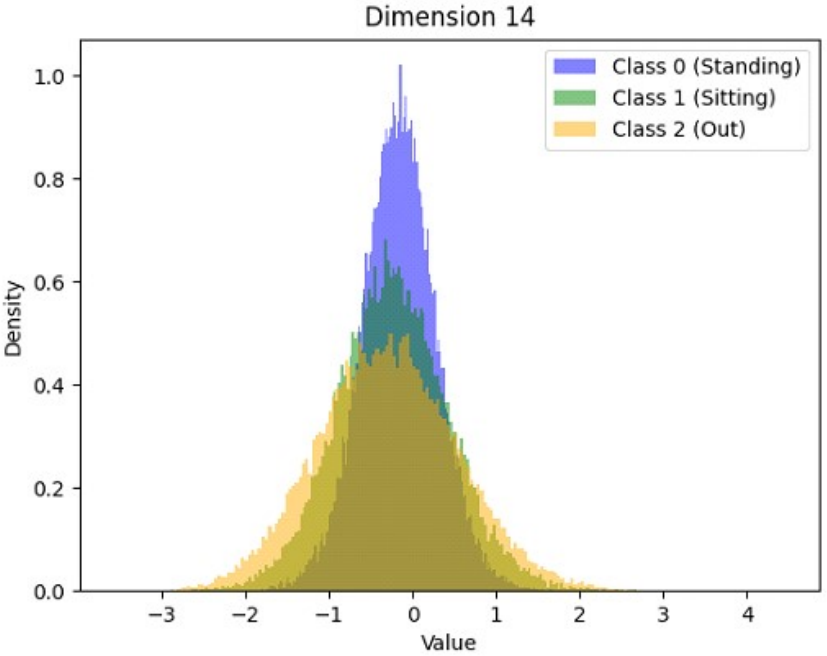

Figure C.15: Histogram of the feature 14.

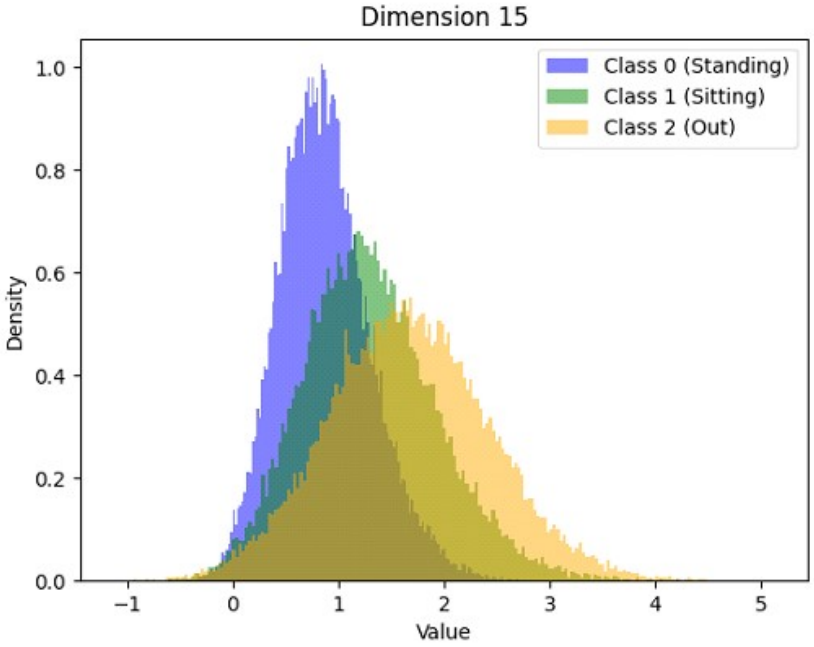

Figure C.16: Histogram of the feature 15.

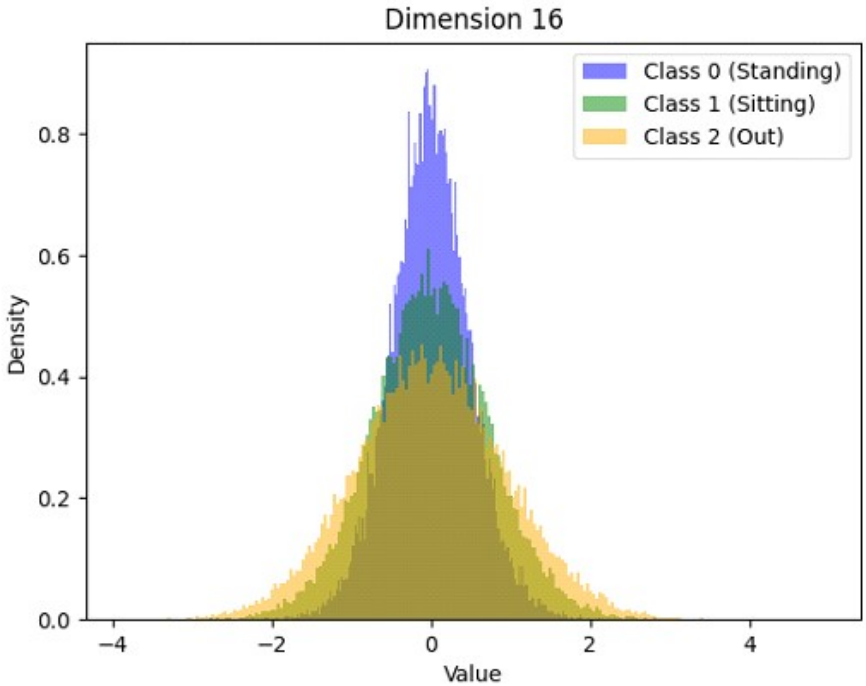

Figure C.17: Histogram of the feature 16.

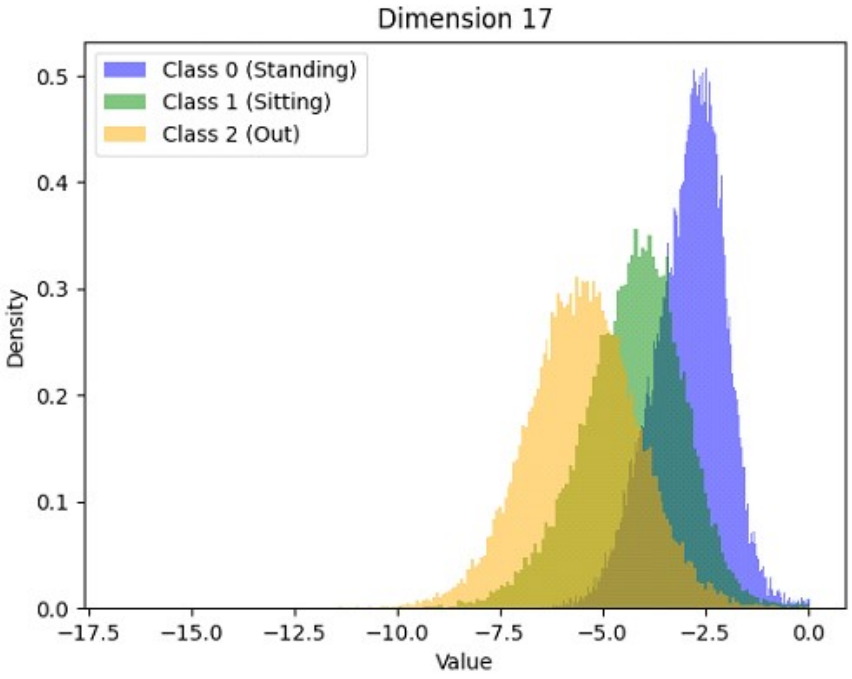

Figure C.18: Histogram of the feature 17.

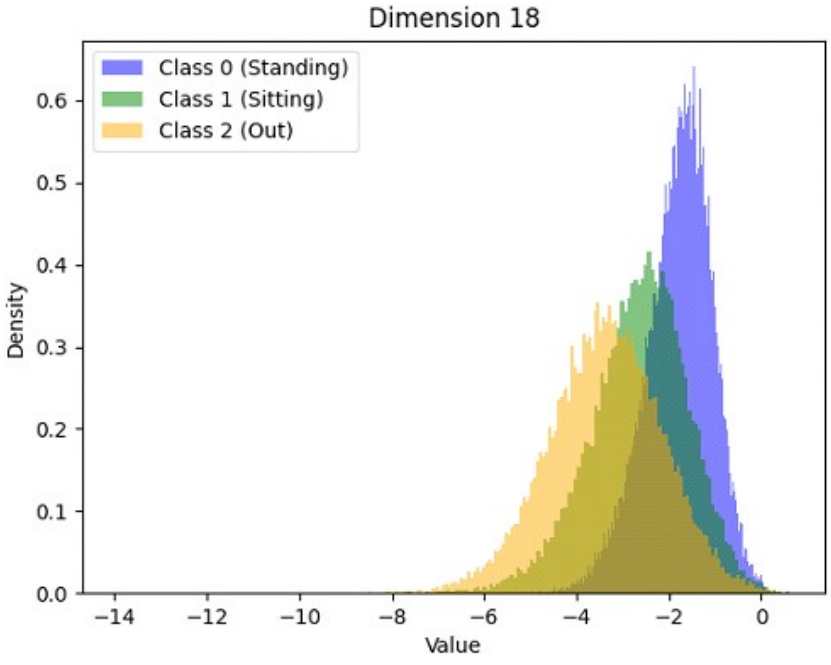

Figure C.19: Histogram of the feature 18.

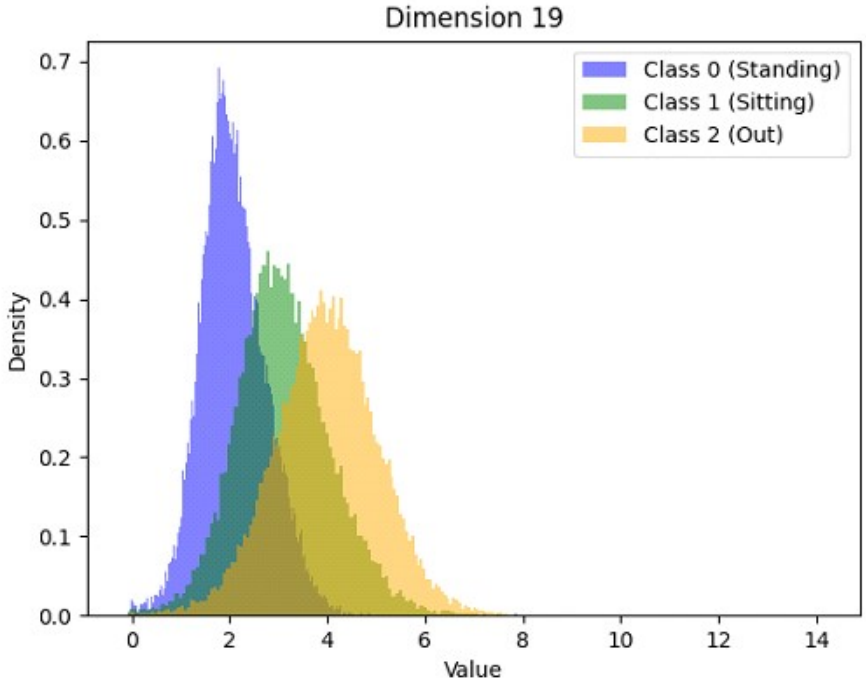

Figure C.20: Histogram of the feature 19.

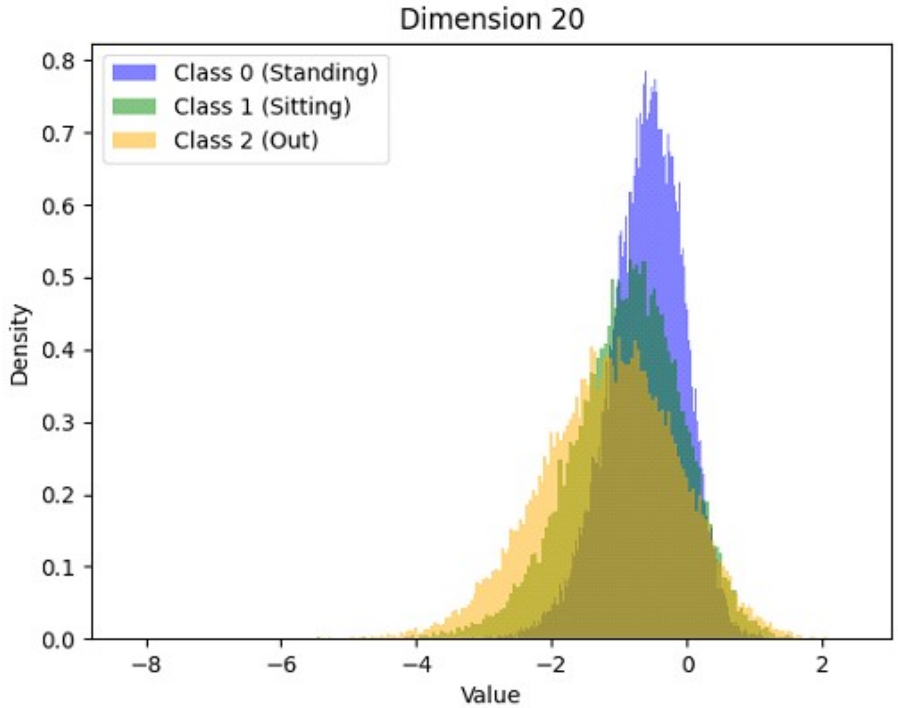

Figure C.21: Histogram of the feature 20.

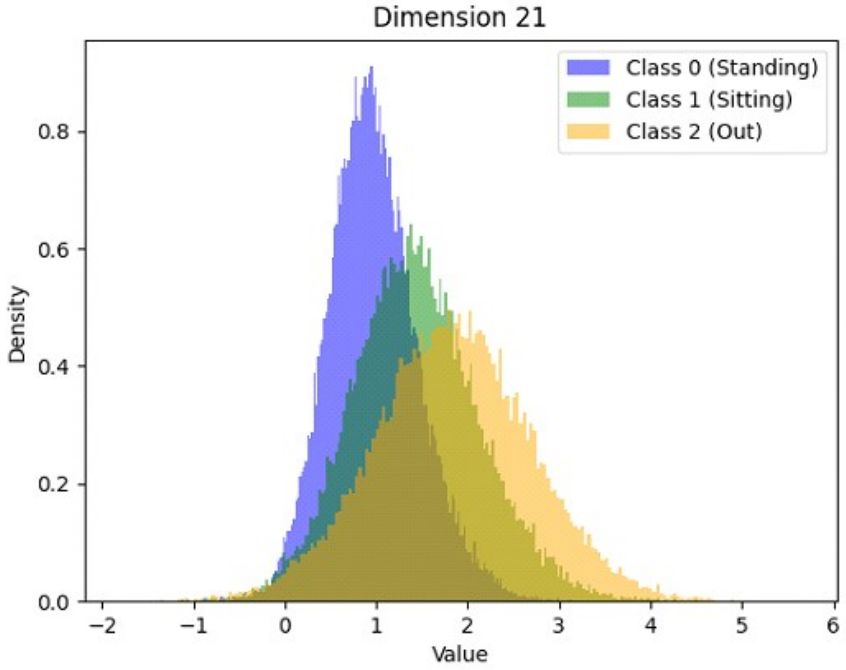

Figure C.22: Histogram of the feature 21.

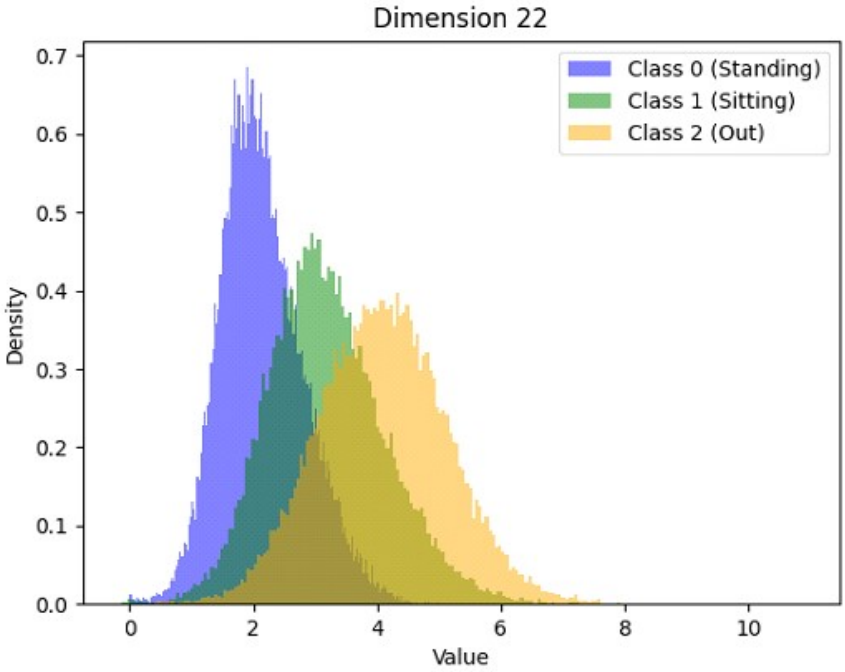

Figure C.23: Histogram of the feature 22.

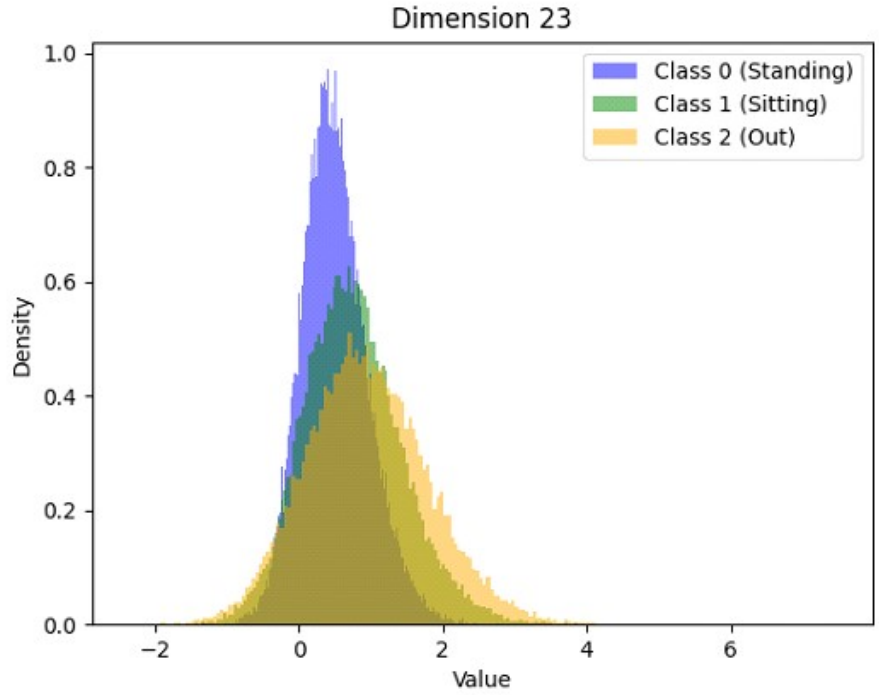

Figure C.24: Histogram of the feature 23.

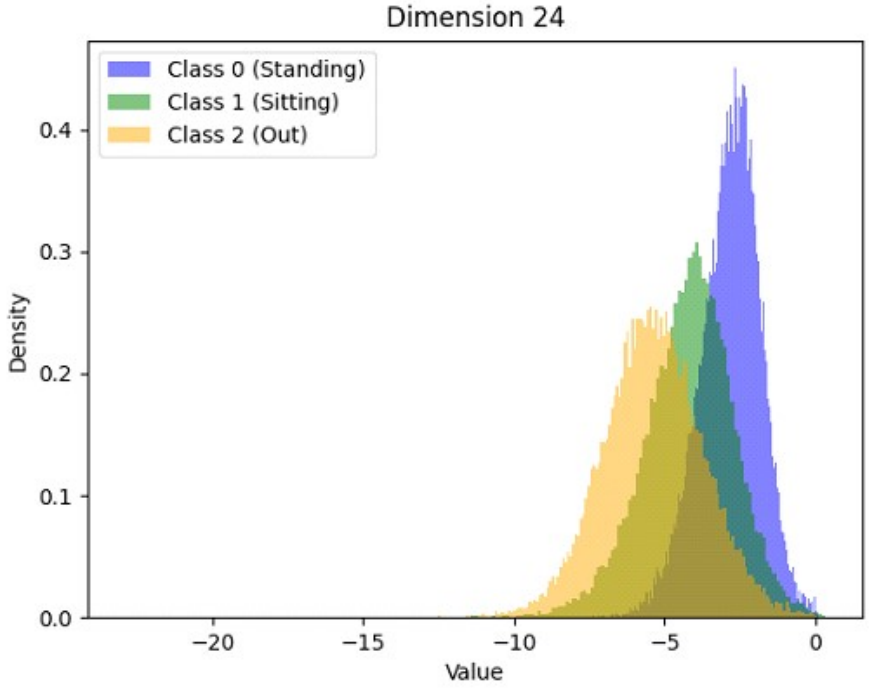

Figure C.25: Histogram of the feature 24.

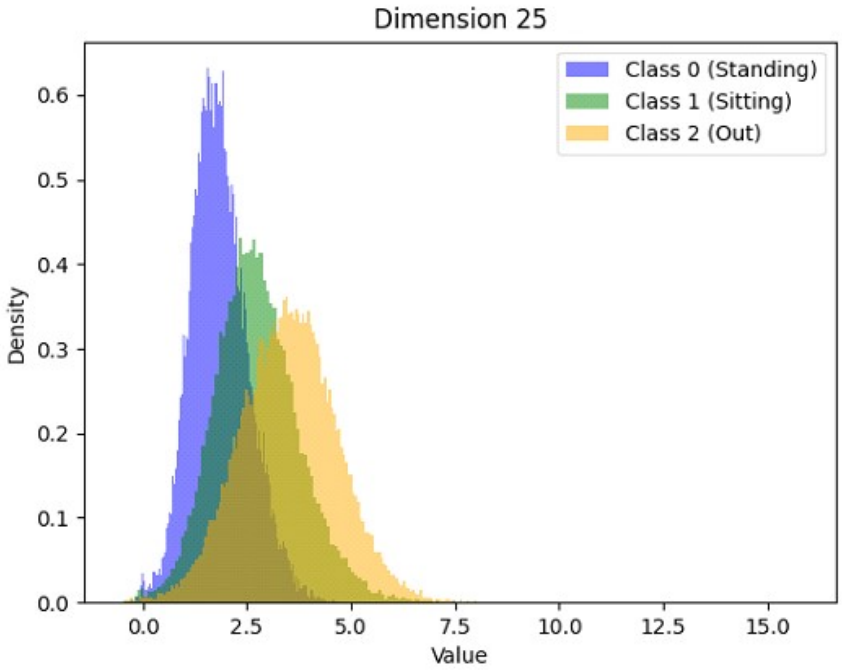

Figure C.26: Histogram of the feature 25.

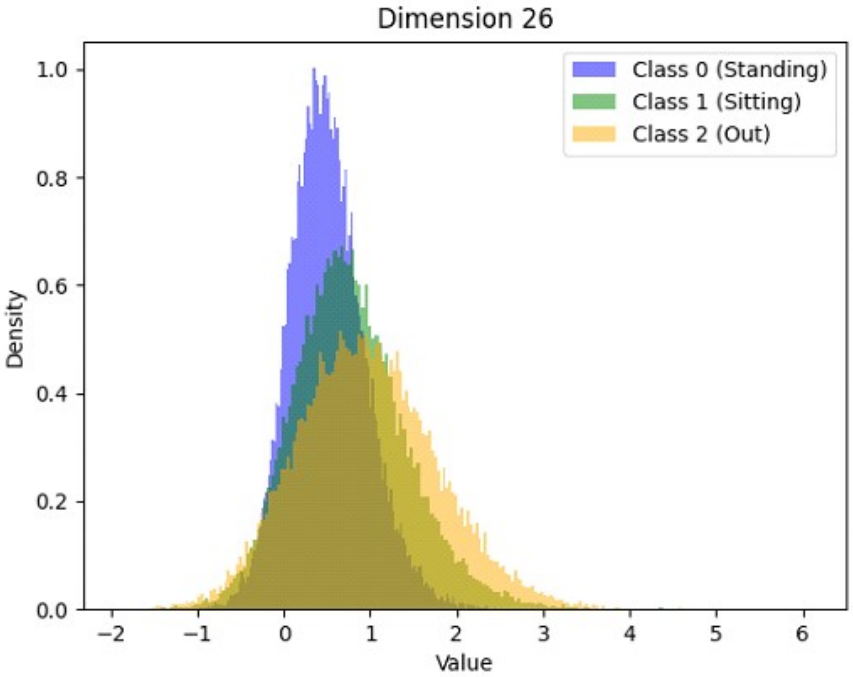

Figure C.27: Histogram of the feature 26.

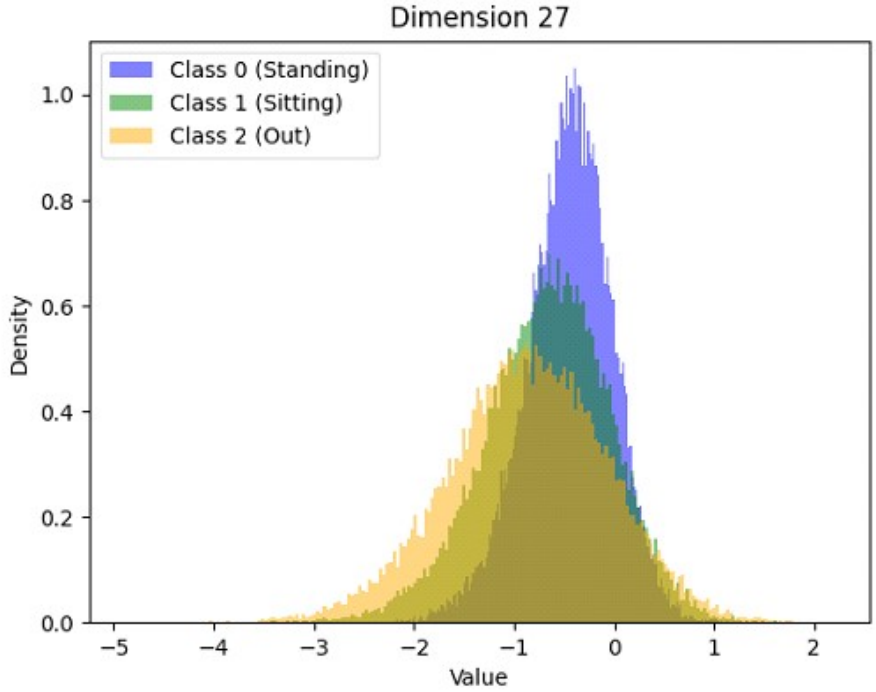

Figure C.28: Histogram of the feature 27.

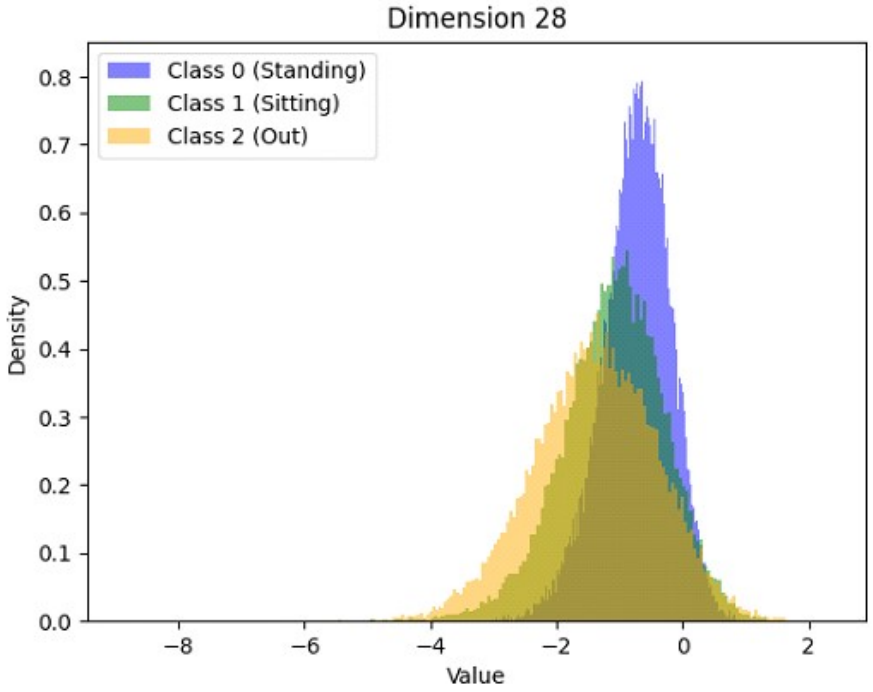

Figure C.29: Histogram of the feature 28.

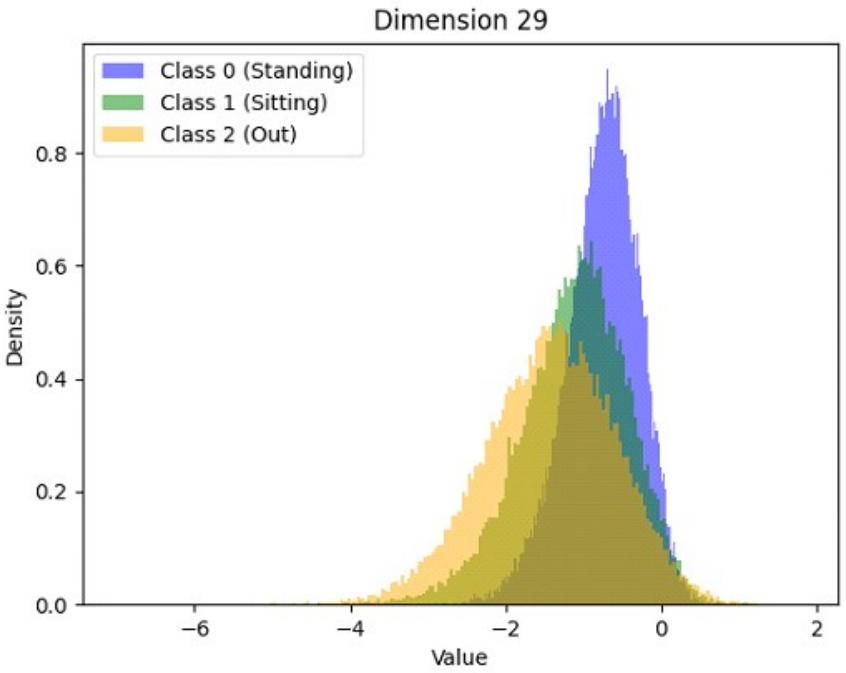

Figure C.30: Histogram of the feature 29.

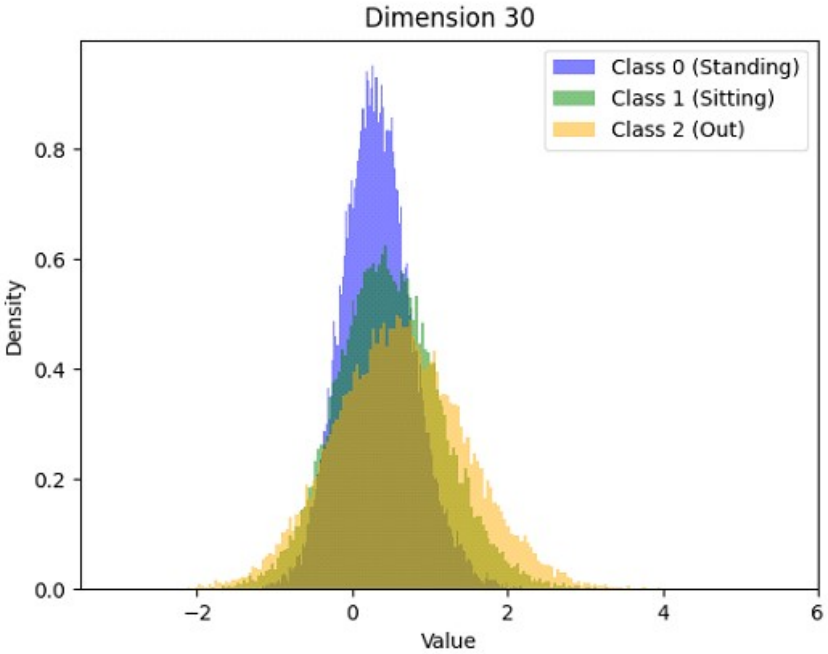

Figure C.31: Histogram of the feature 30.

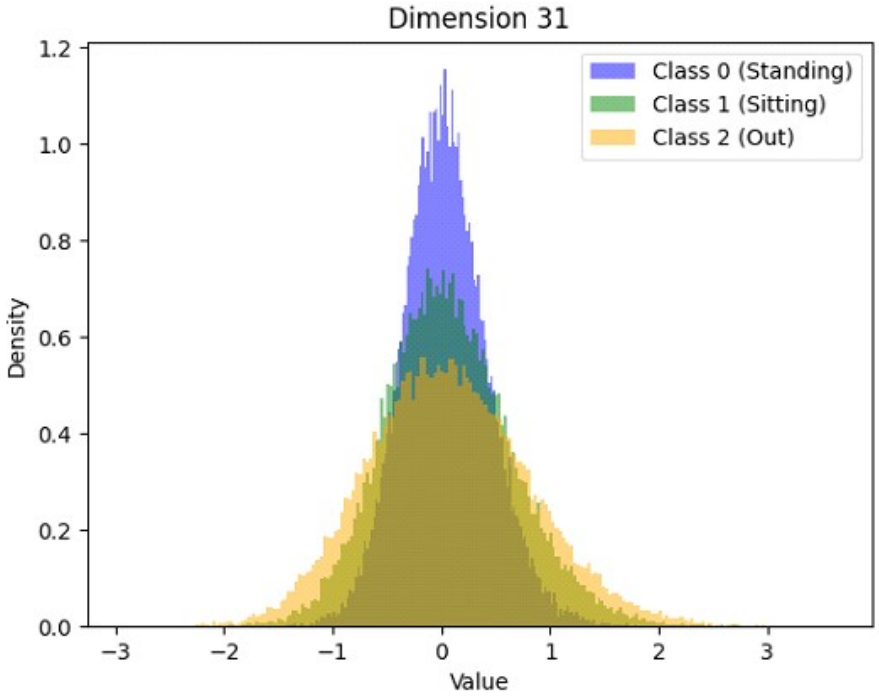

Figure C.32: Histogram of the feature 31.

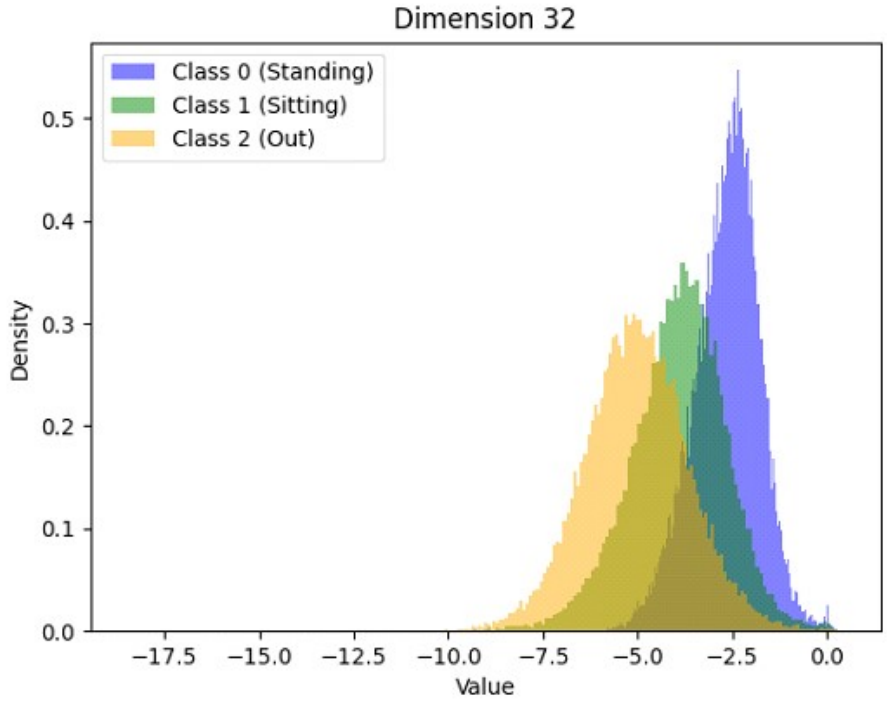

Figure C.33: Histogram of the feature 32.

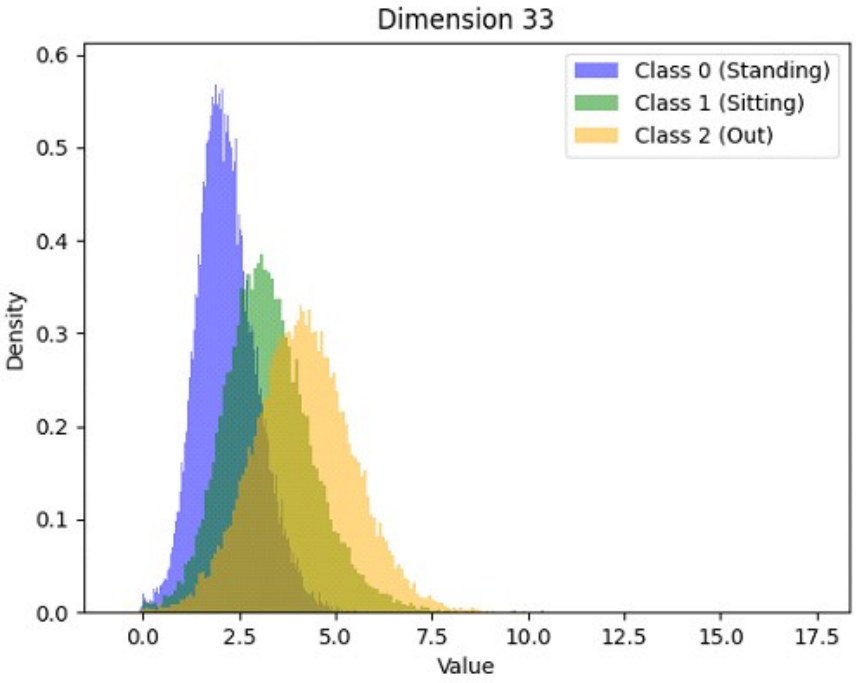

Figure C.34: Histogram of the feature 33.

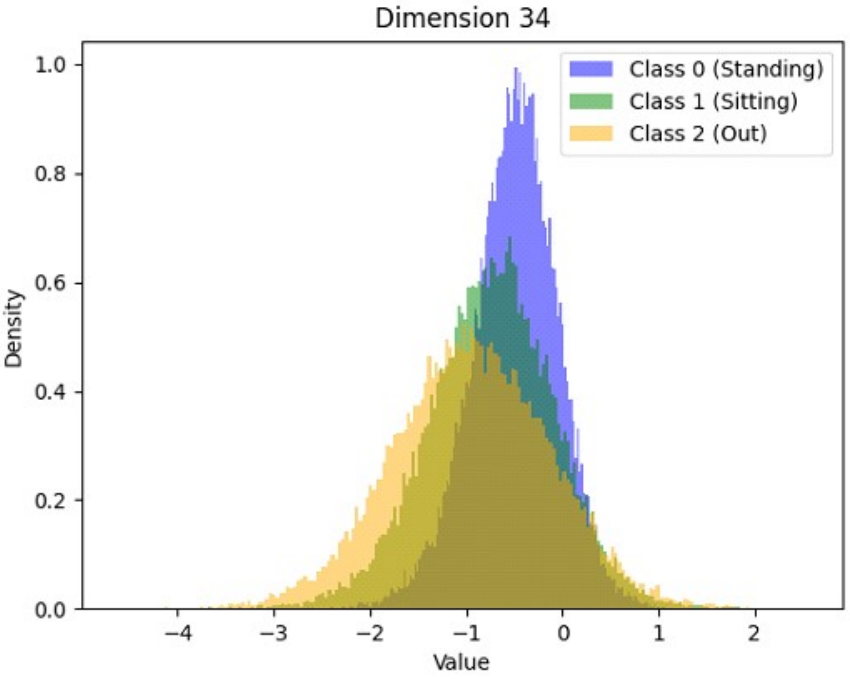

Figure C.35: Histogram of the feature 34.

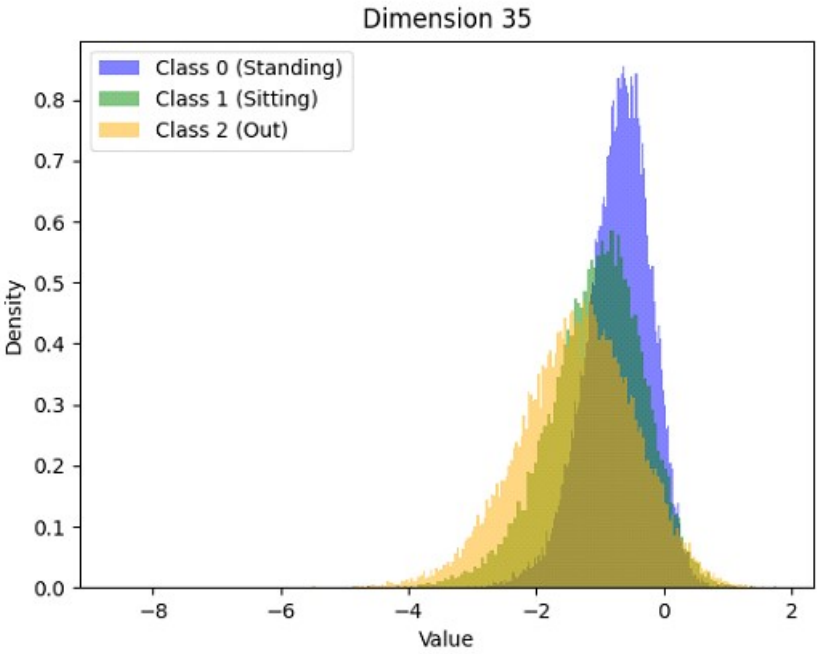

Figure C.36: Histogram of the feature 35.

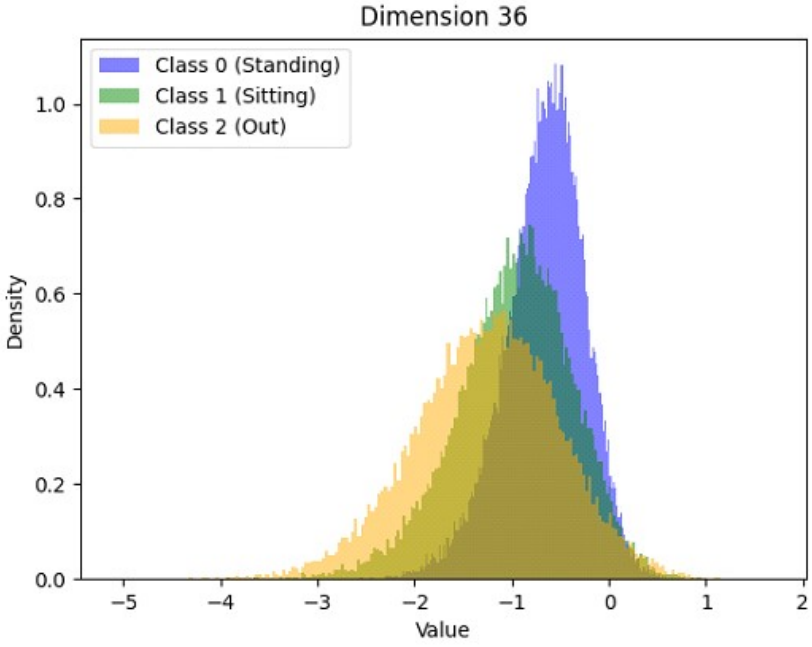

Figure C.37: Histogram of the feature 36.

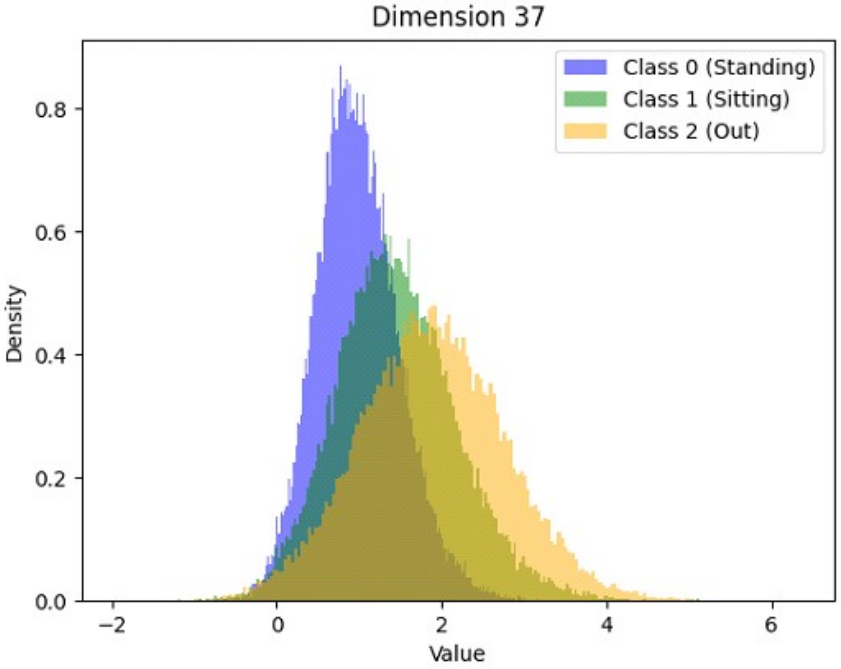

Figure C.38: Histogram of the feature 37.

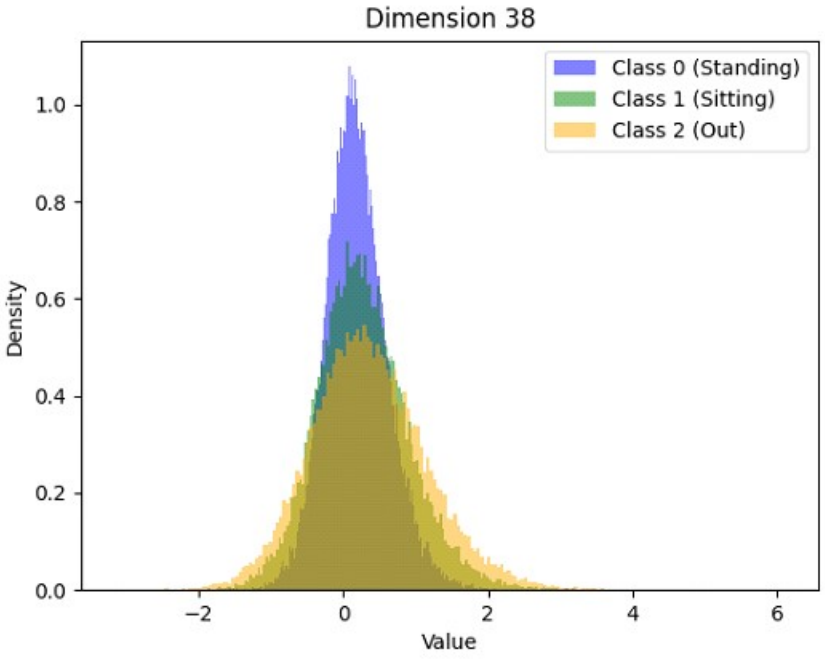

Figure C.39: Histogram of the feature 38.

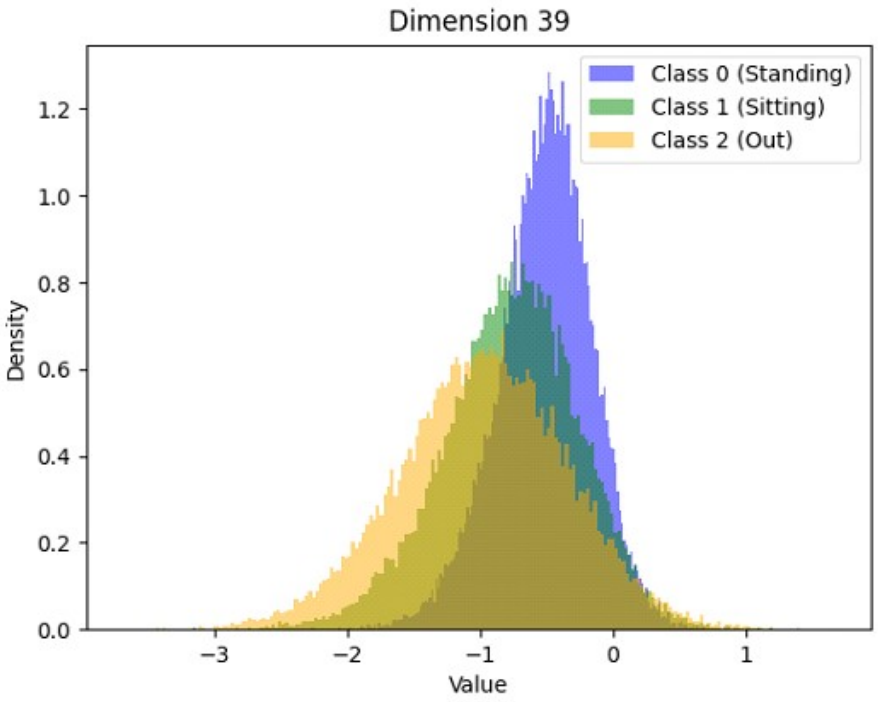

Figure C.40: Histogram of the feature 39.

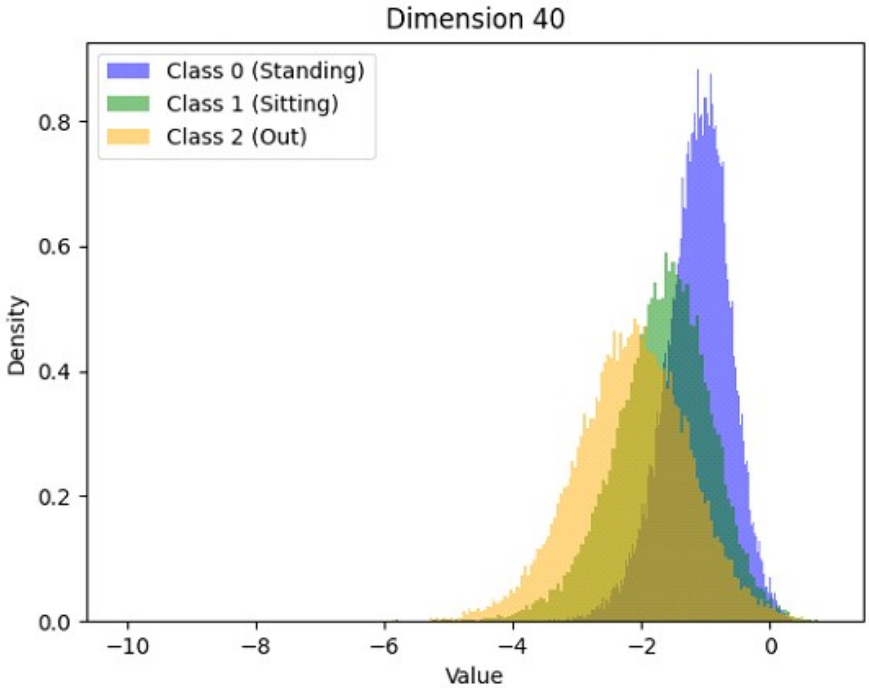

Figure C.41: Histogram of the feature 40.

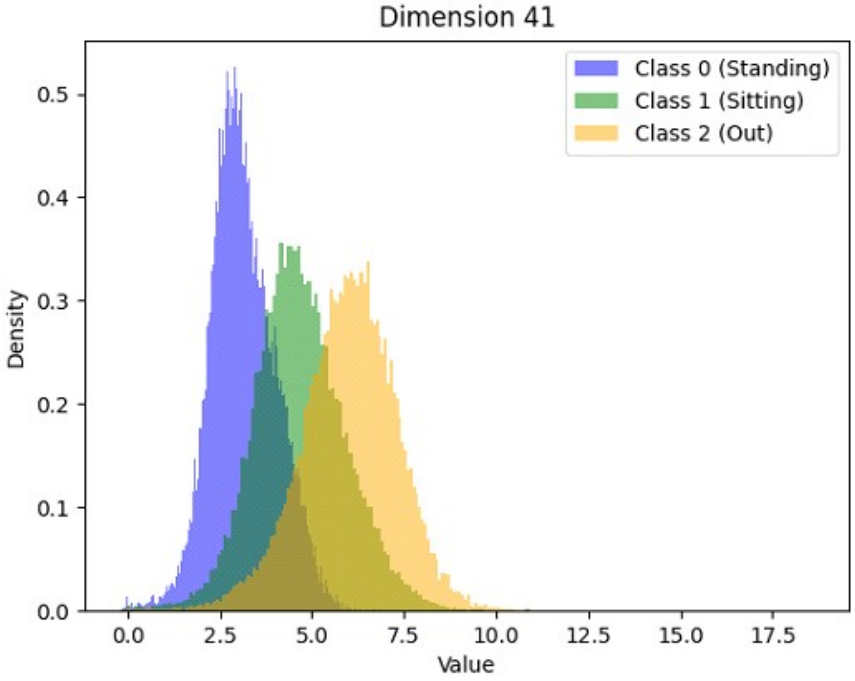

Figure C.42: Histogram of the feature 41.

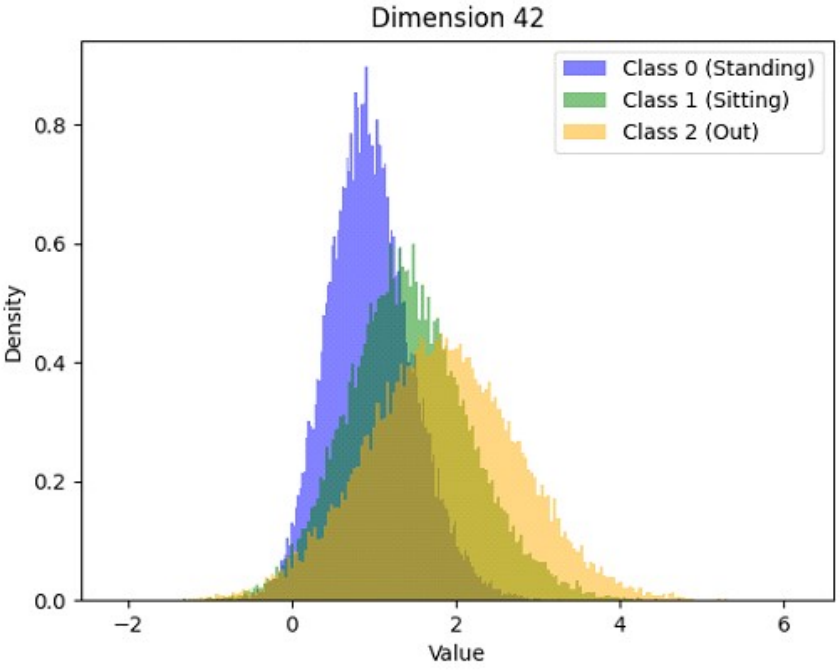

Figure C.43: Histogram of the feature 42.

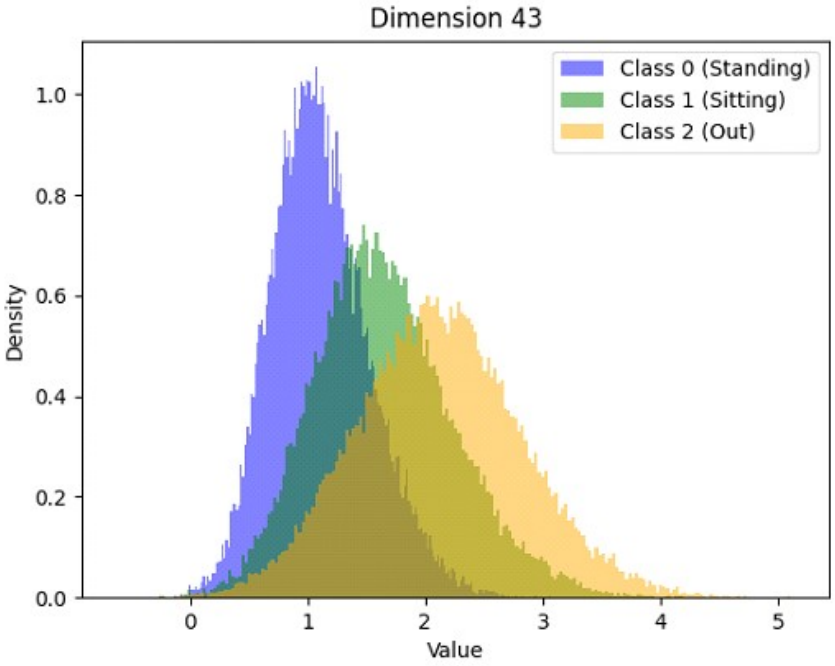

Figure C.44: Histogram of the feature 43.

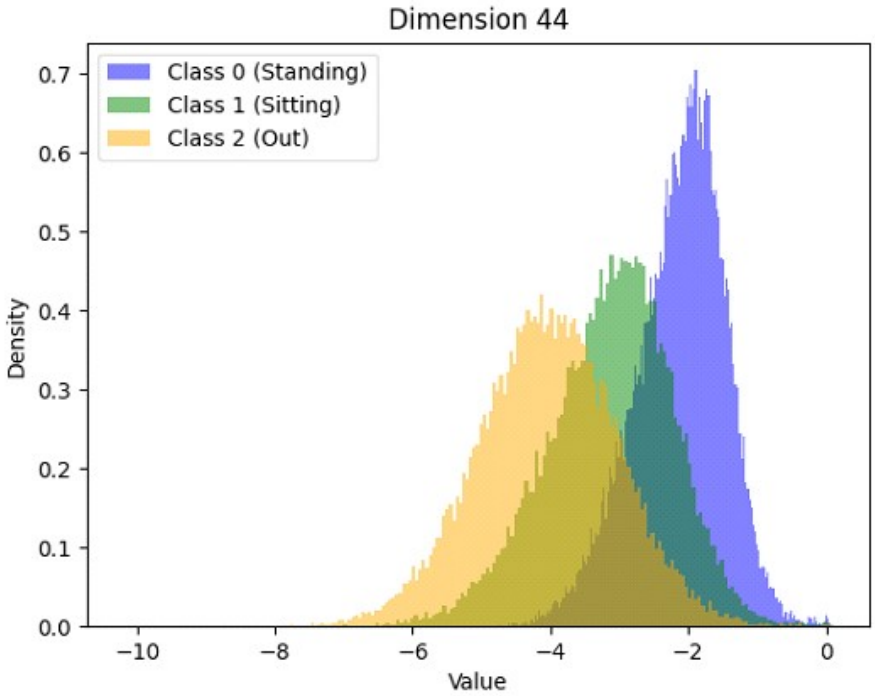

Figure C.45: Histogram of the feature 44.

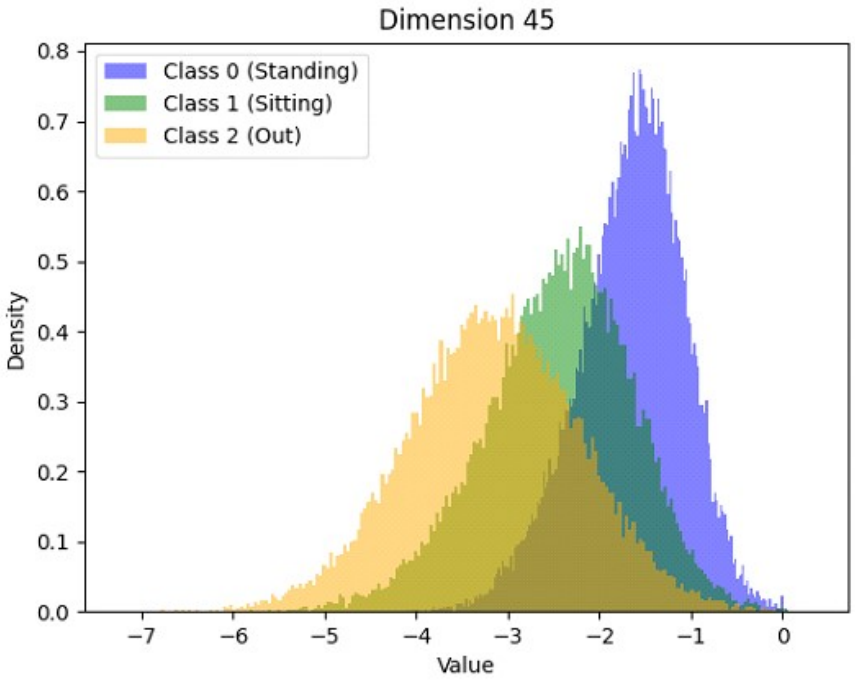

Figure C.46: Histogram of the feature 45.

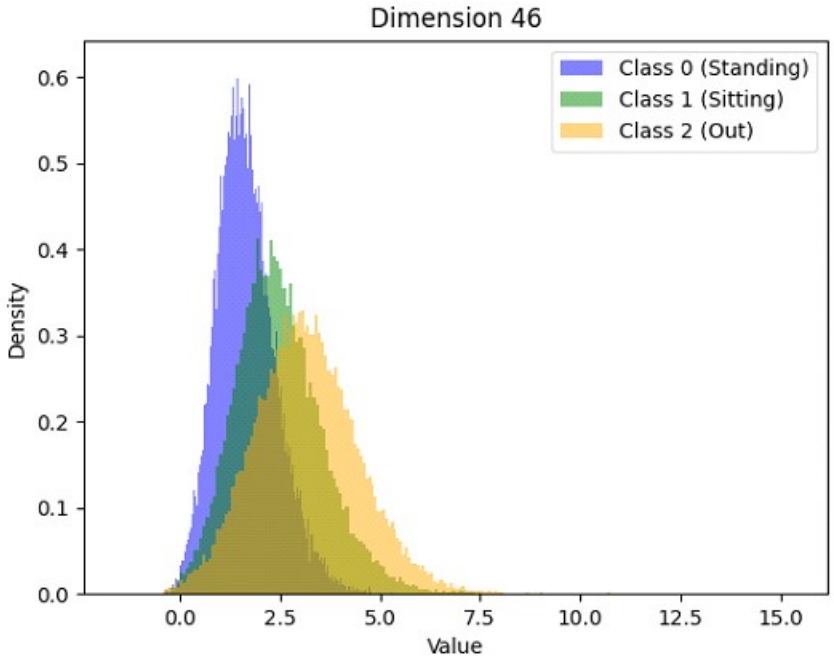

Figure C.47: Histogram of the feature 46.

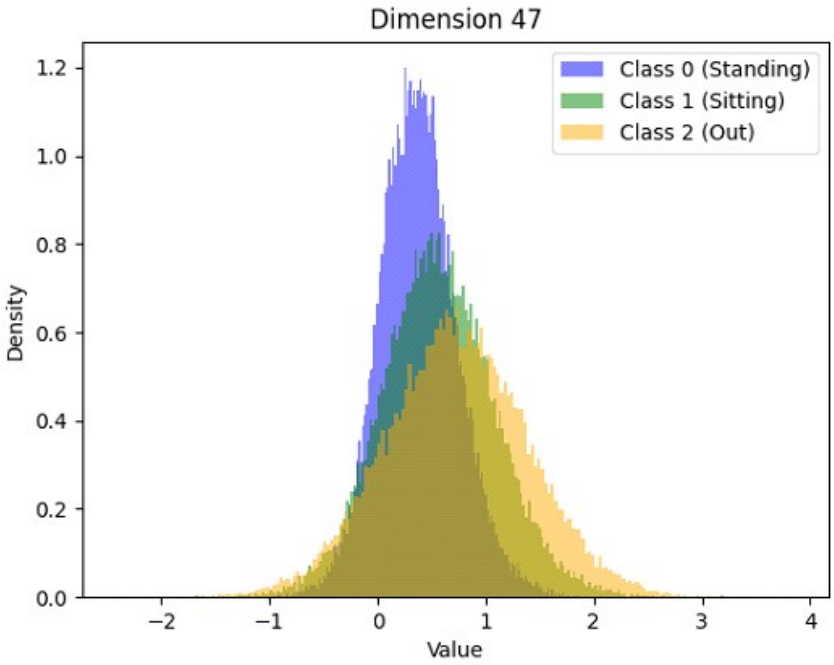

Figure C.48: Histogram of the feature 47.

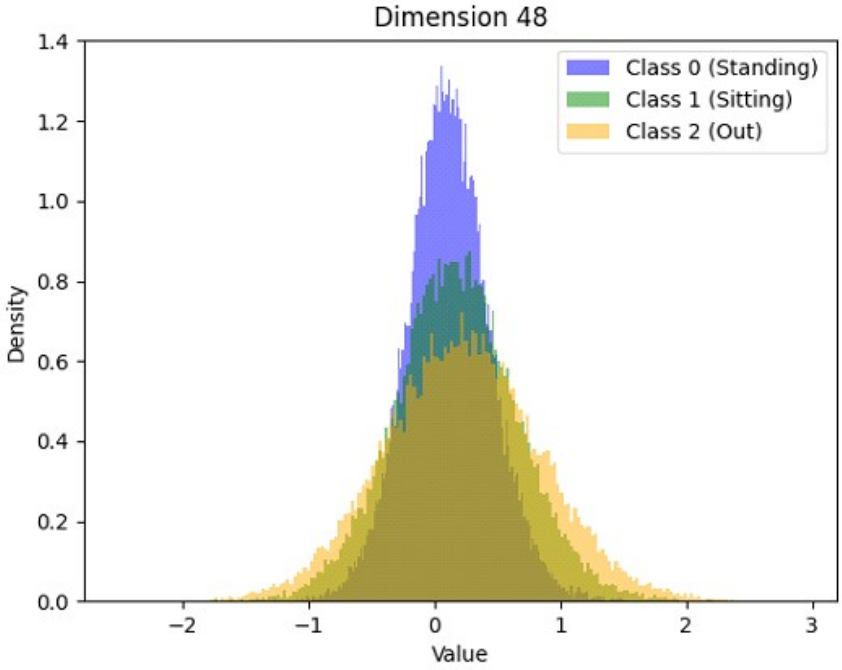

Figure C.49: Histogram of the feature 48.

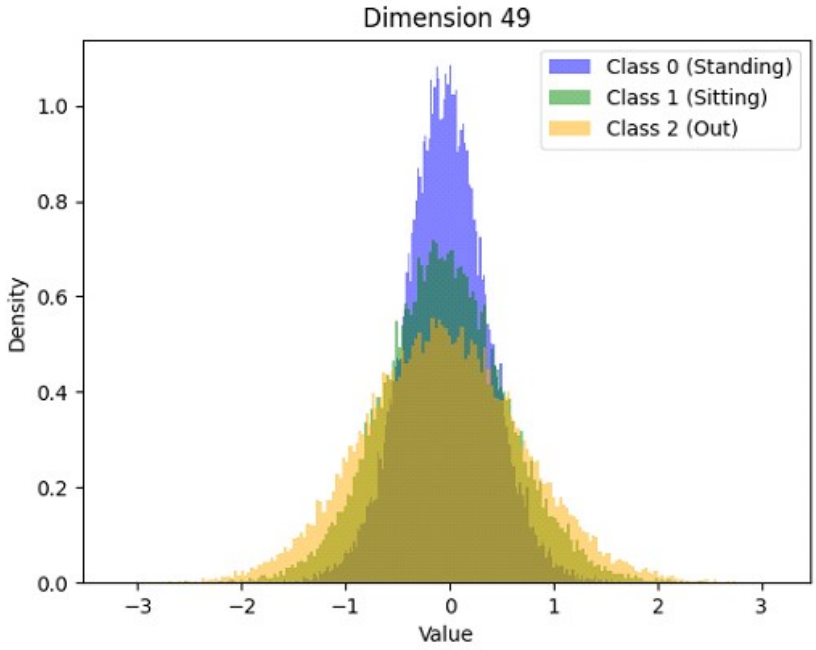

Figure C.50: Histogram of the feature 49.

Appendix 4

In this appendix, we present the histograms for all 10 latent features extracted using the VAE with a frequency of 19 kHz.

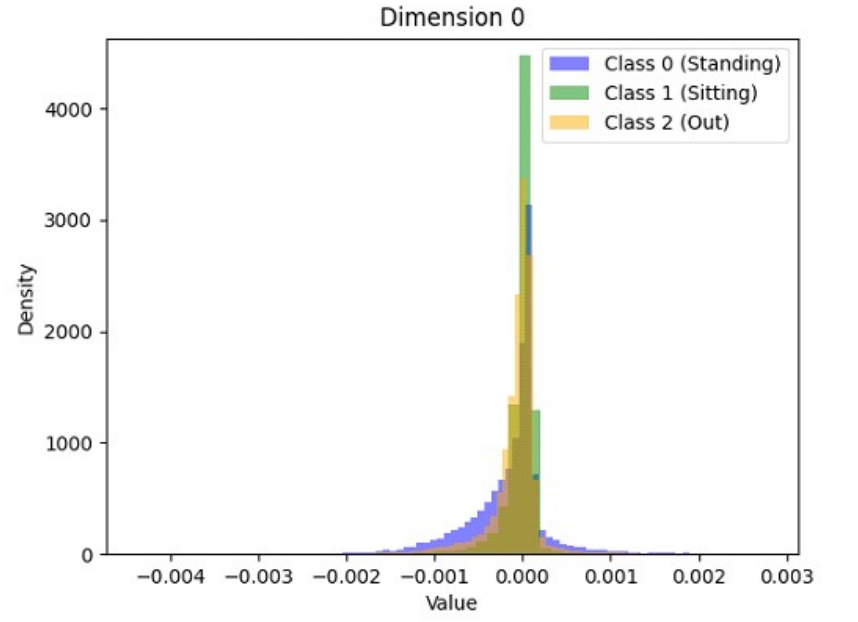

Figure D.1: Histogram of the feature 0.

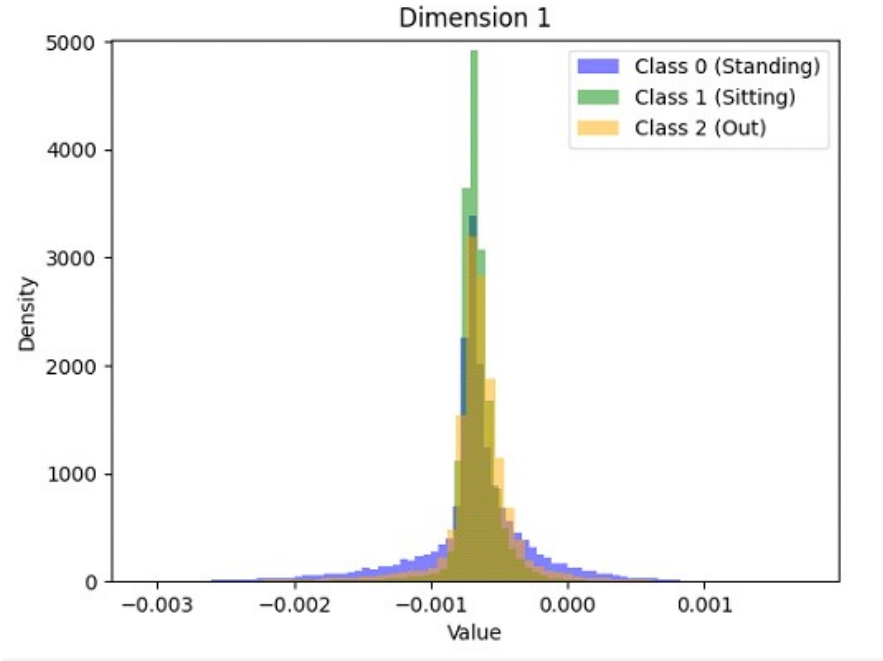

Figure D.2: Histogram of the feature 1.

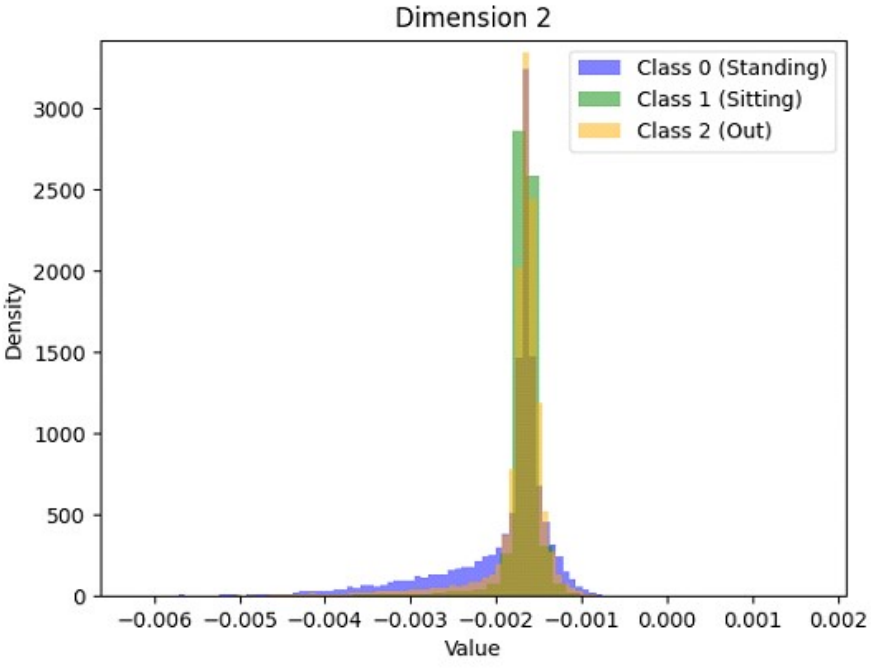

Figure D.3: Histogram of the feature 2.

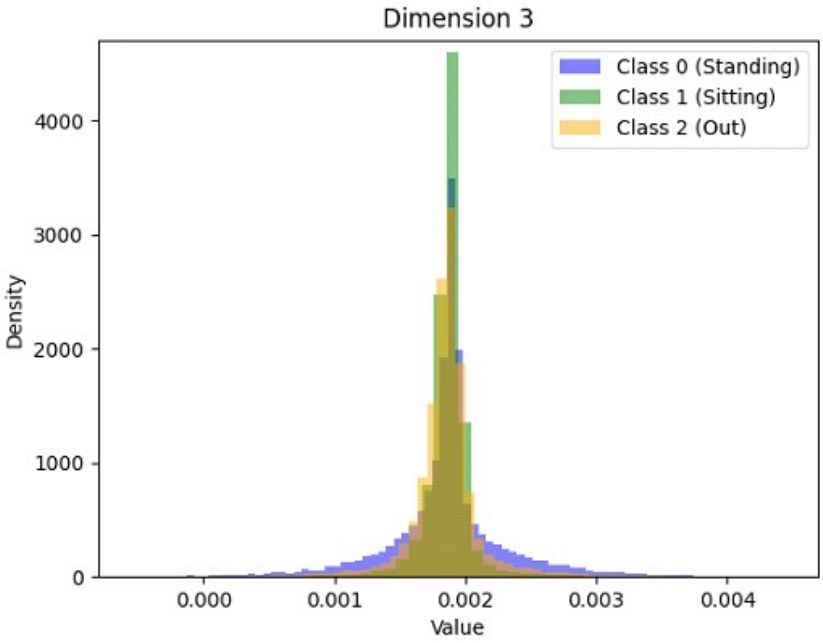

Figure D.4: Histogram of the feature 3.

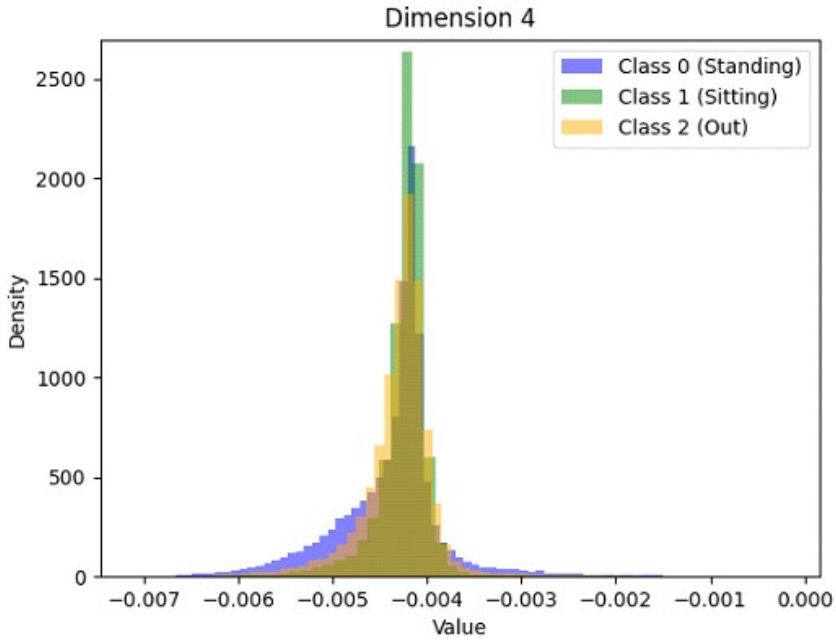

Figure D.5: Histogram of the feature 4.

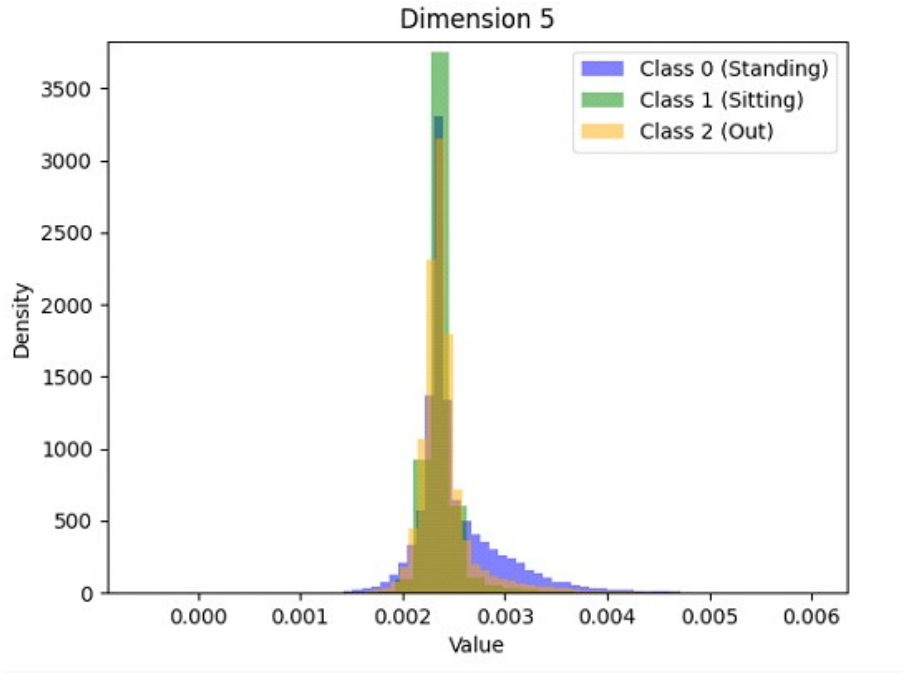

Figure D.6: Histogram of the feature 5.

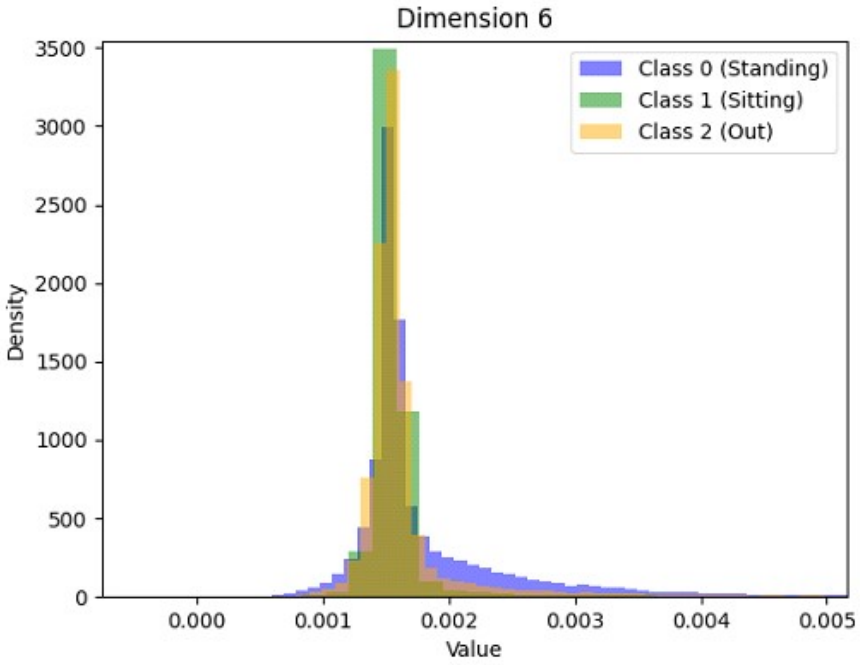

Figure D.7: Histogram of the feature 6.

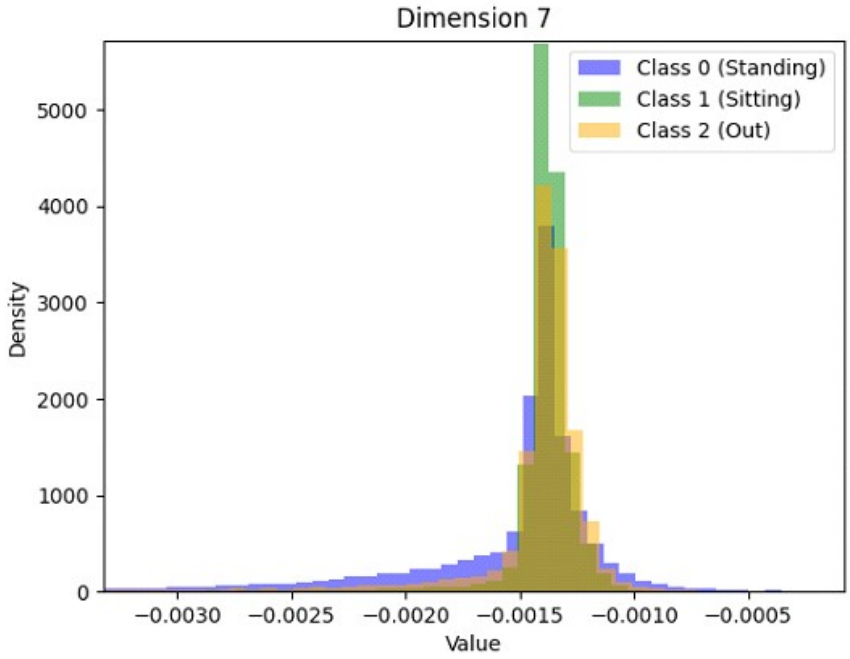

Figure D.8: Histogram of the feature 7.

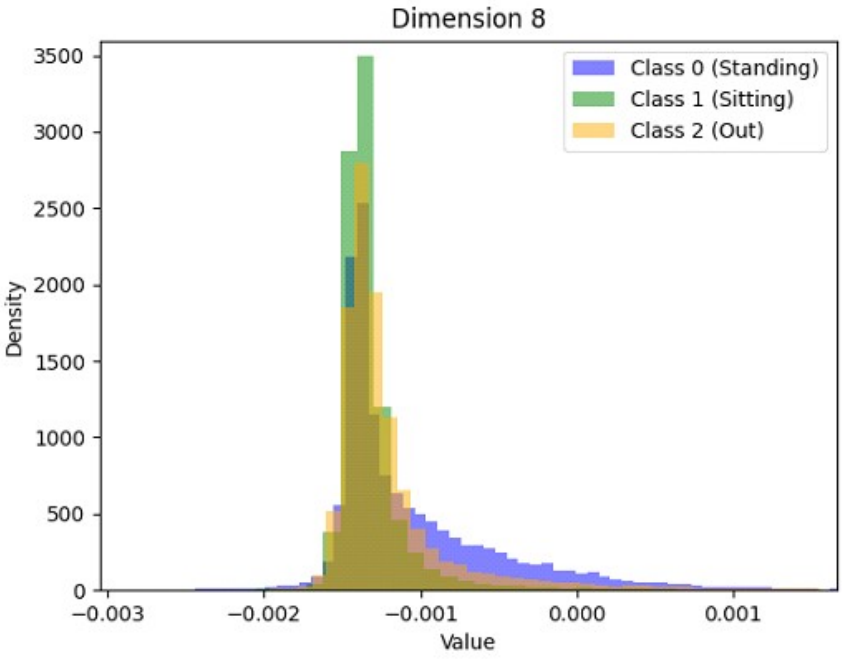

Figure D.9: Histogram of the feature 8.

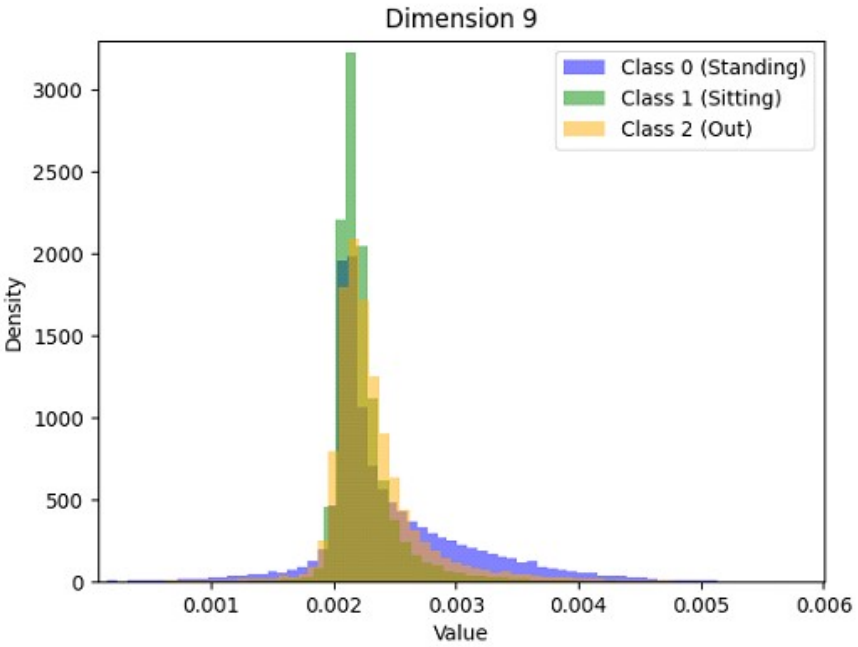

Figure D.10: Histogram of the feature 9.

Appendix 5

In this appendix, we present the histograms for all 10 latent features extracted using the VAE using the complete band.

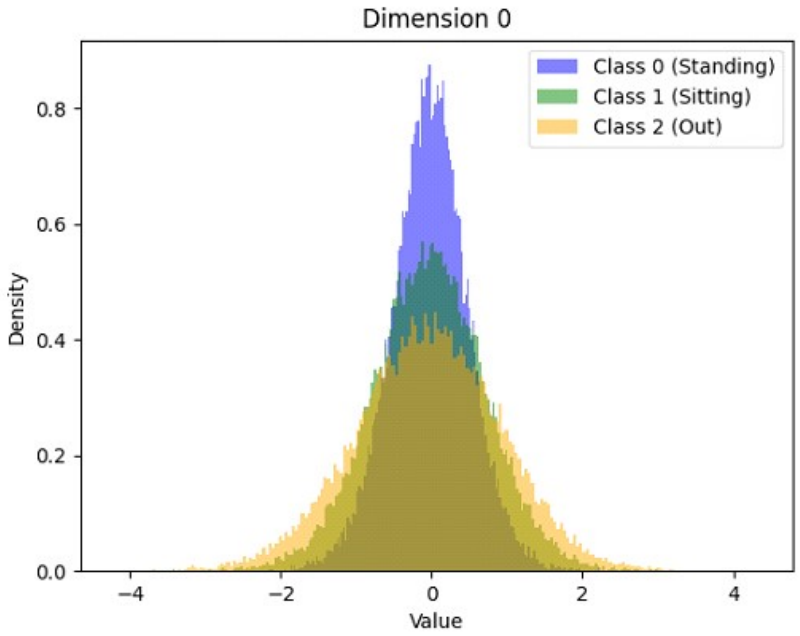

Figure E.1: Histogram of the feature 0.

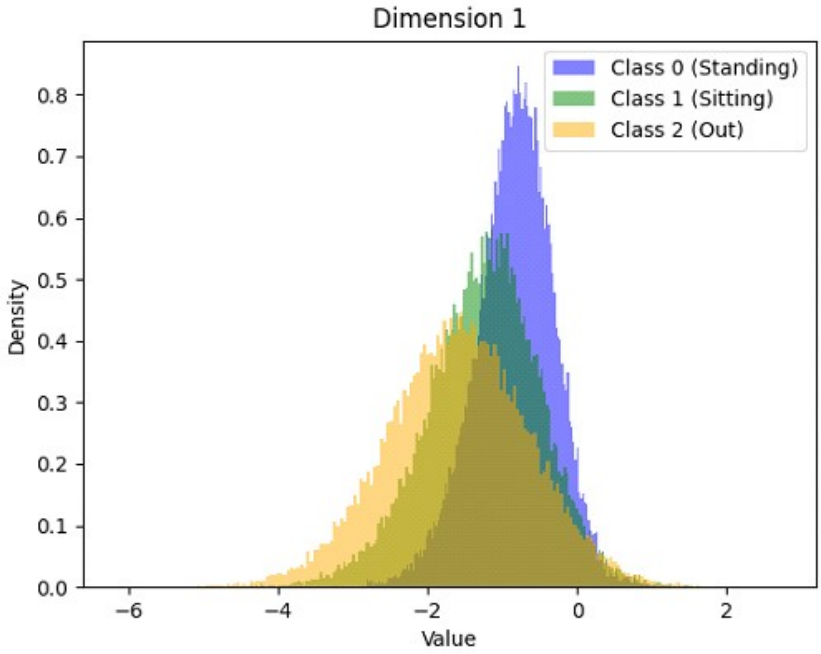

Figure E.2: Histogram of the feature 1.

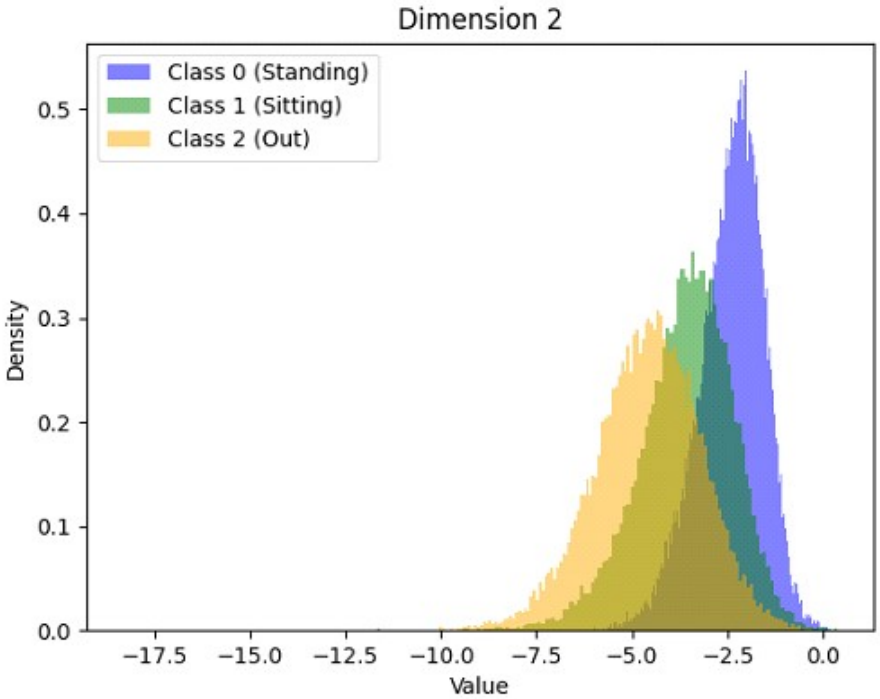

Figure E.3: Histogram of the feature 2.

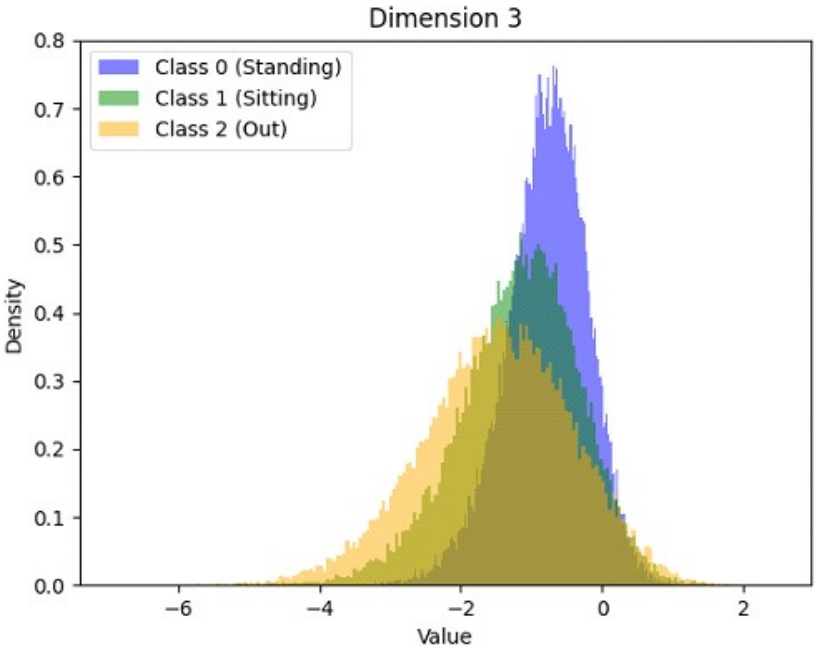

Figure E.4: Histogram of the feature 3.

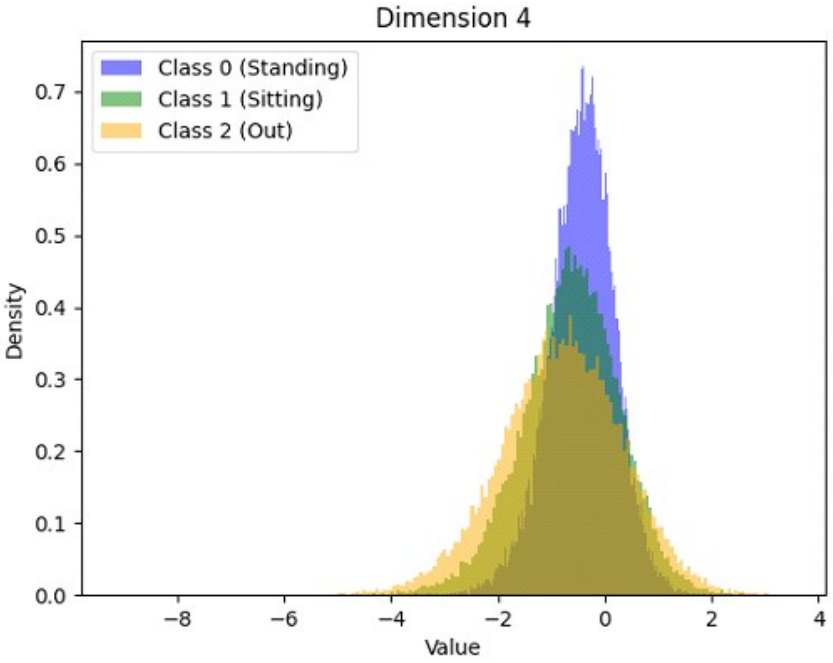

Figure E.5: Histogram of the feature 4.

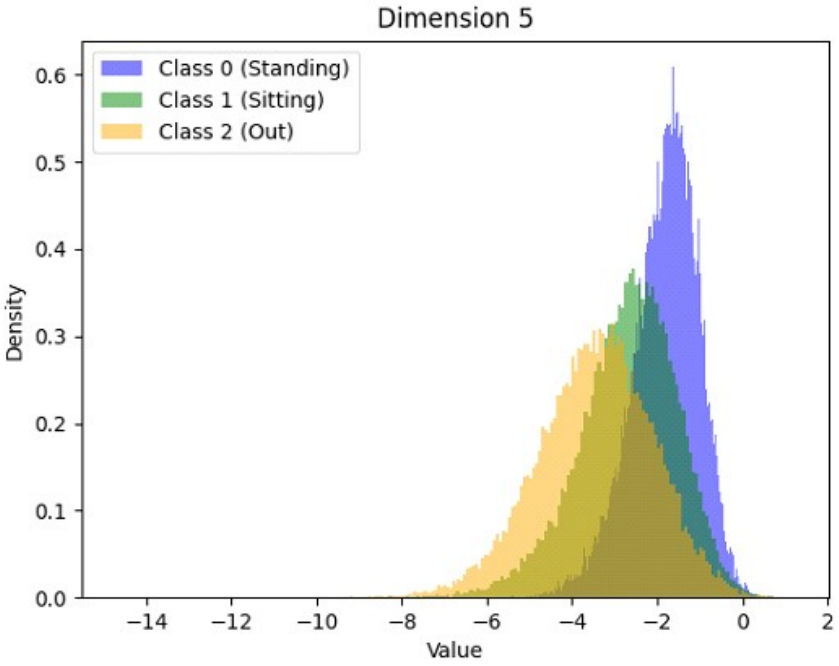

Figure E.6: Histogram of the feature 5.

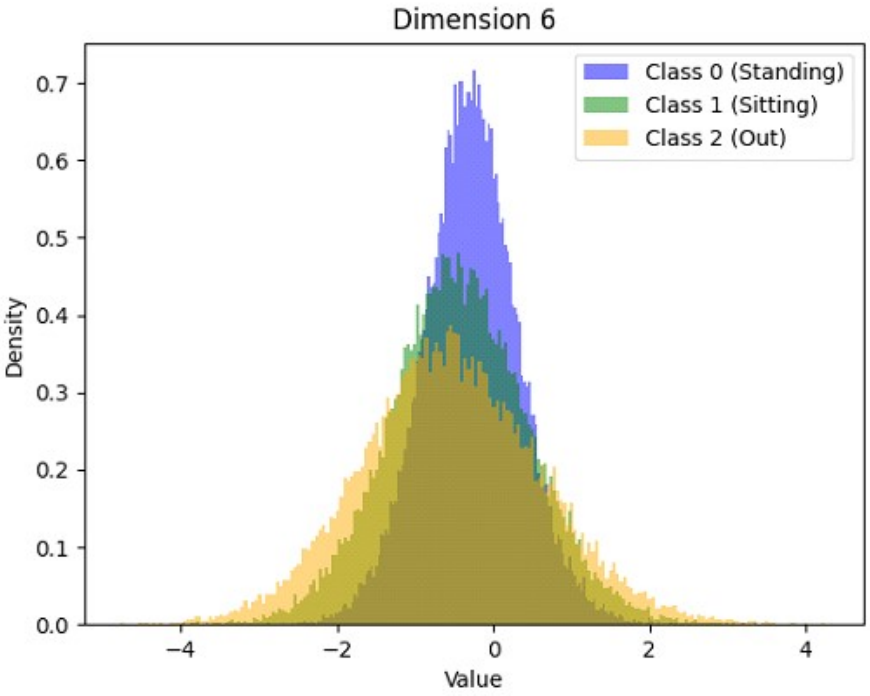

Figure E.7: Histogram of the feature 6.

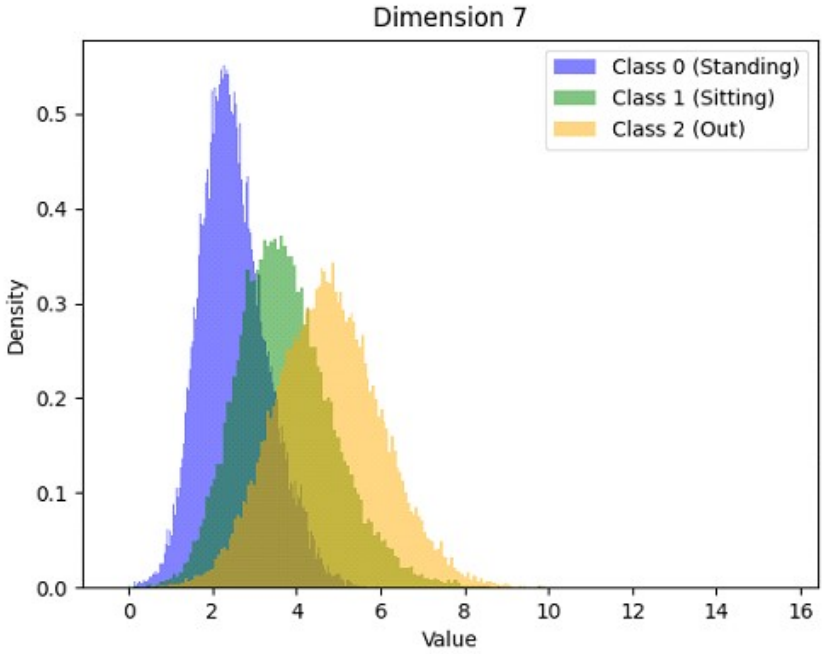

Figure E.8: Histogram of the feature 7.

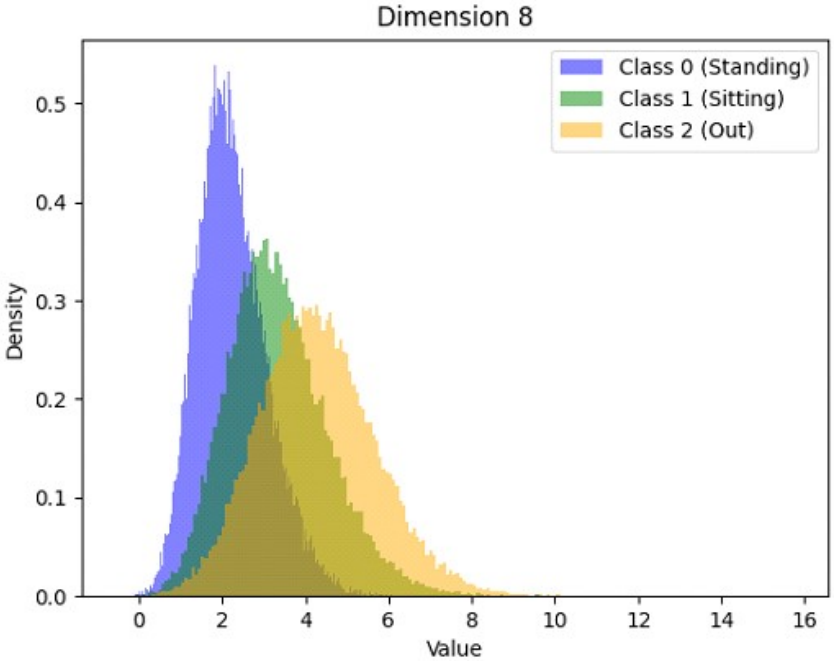

Figure E.9: Histogram of the feature 8.

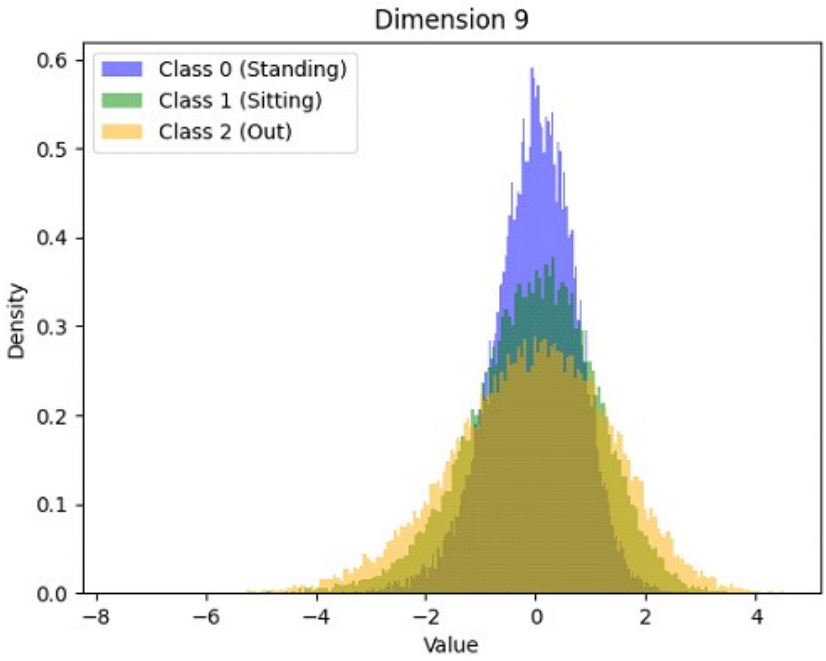

Figure E.10: Histogram of the feature 9.

Appendix 6

In this appendix, we present the histograms for all 50 latent features extracted using the VAE with a frequency of 19 kHz.

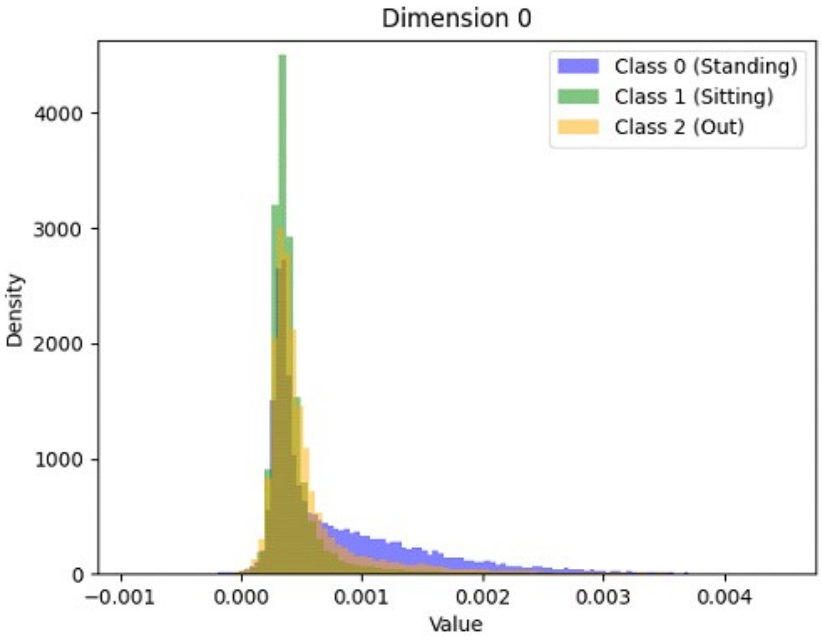

Figure F.1: Histogram of the feature 0.

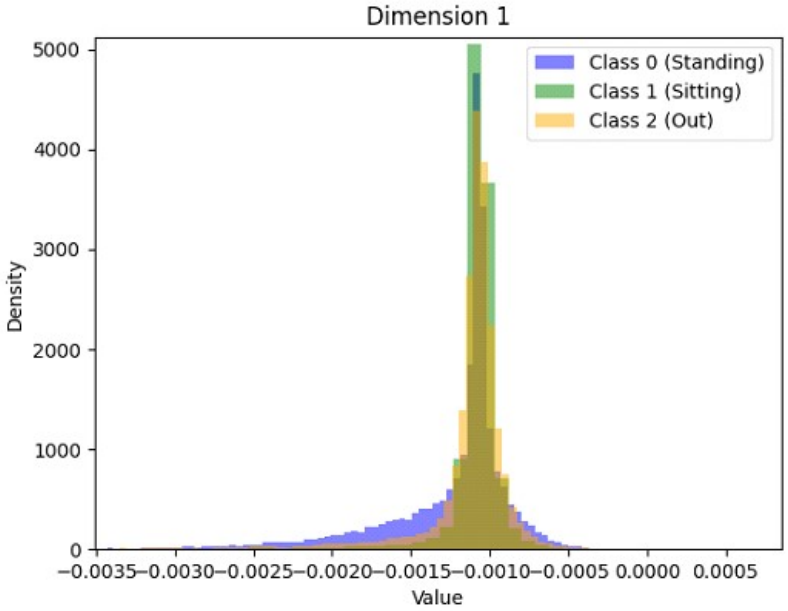

Figure F.2: Histogram of the feature 1.

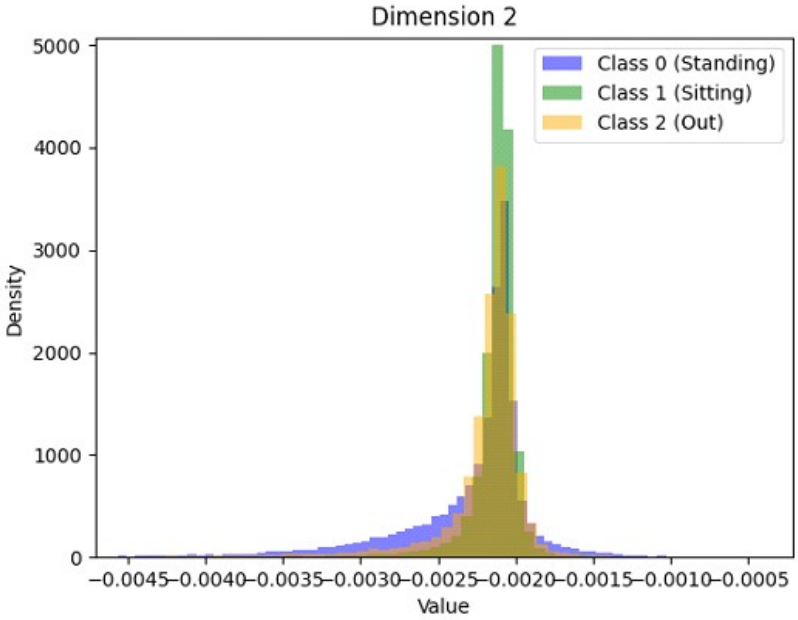

Figure F.3: Histogram of the feature 2.

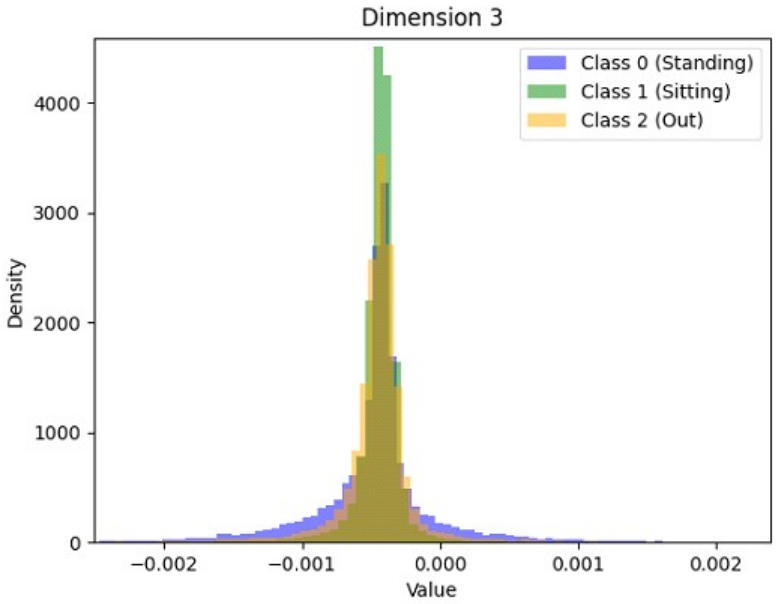

Figure F.4: Histogram of the feature 3.

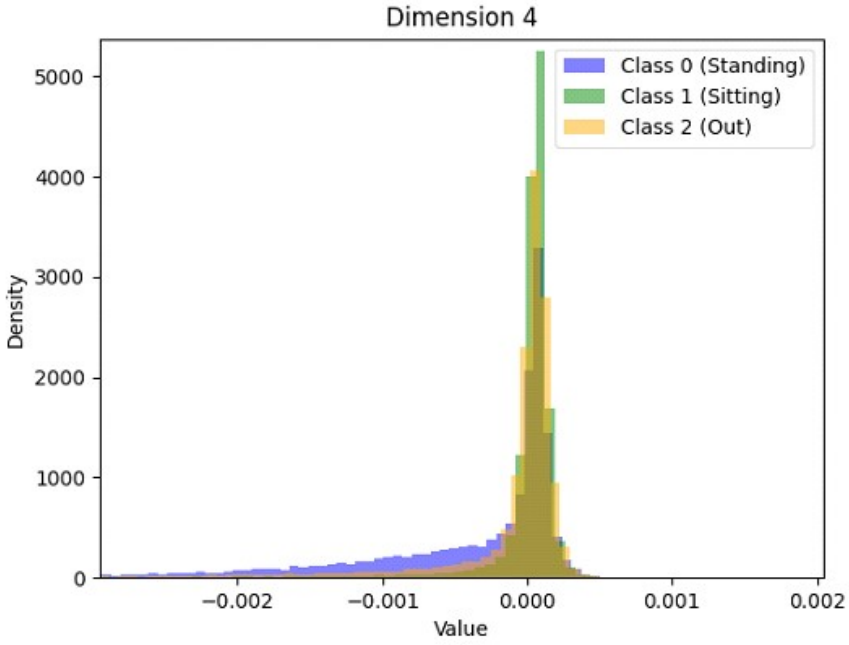

Figure F.5: Histogram of the feature 4.

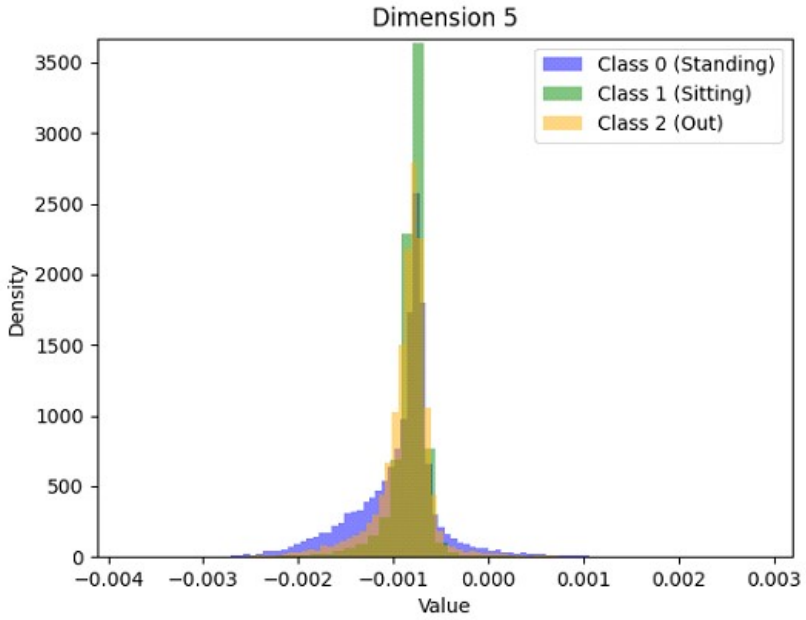

Figure F.6: Histogram of the feature 5.

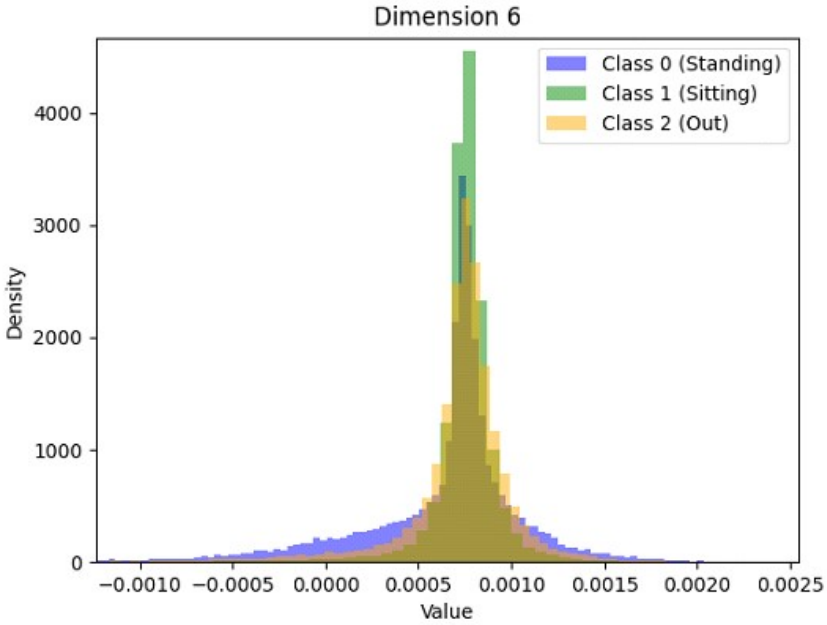

Figure F.7: Histogram of the feature 6.

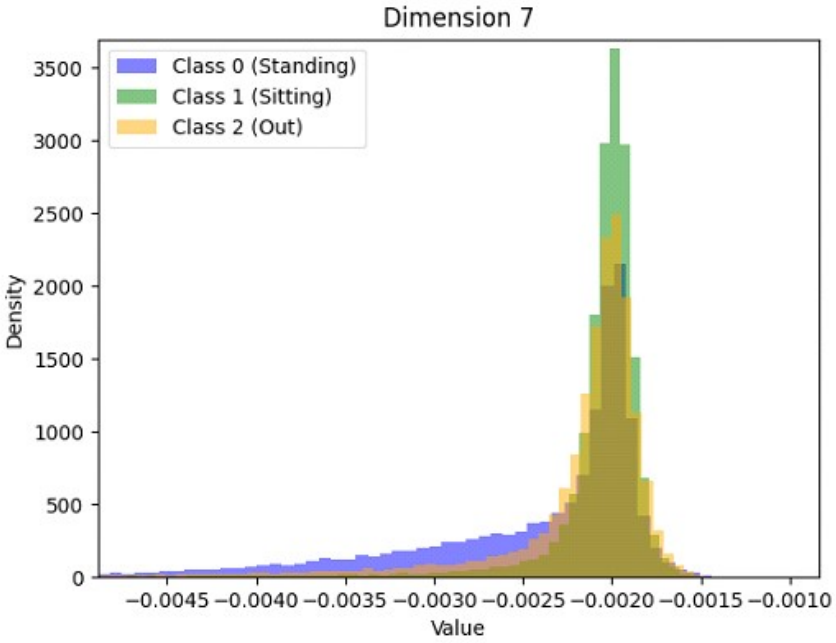

Figure F.8: Histogram of the feature 7.

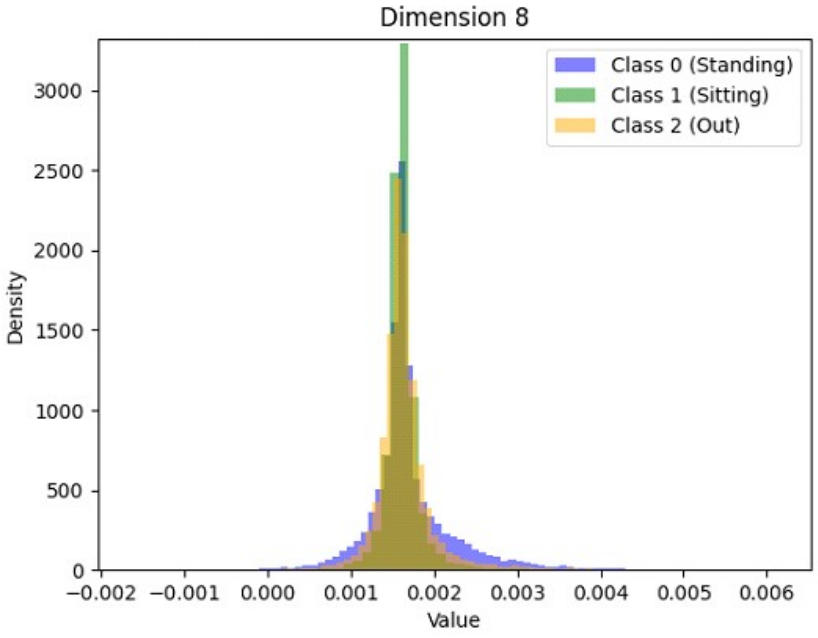

Figure F.9: Histogram of the feature 8.

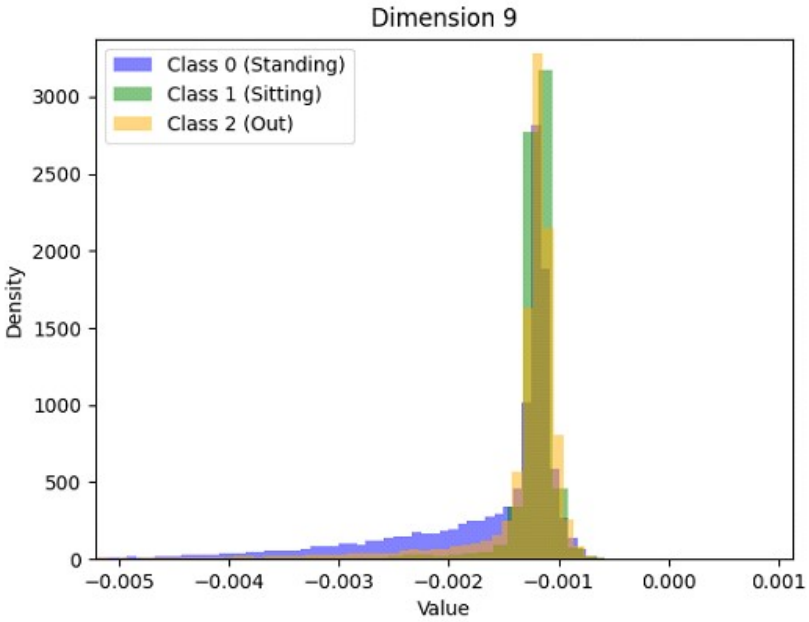

Figure F.10: Histogram of the feature 9.

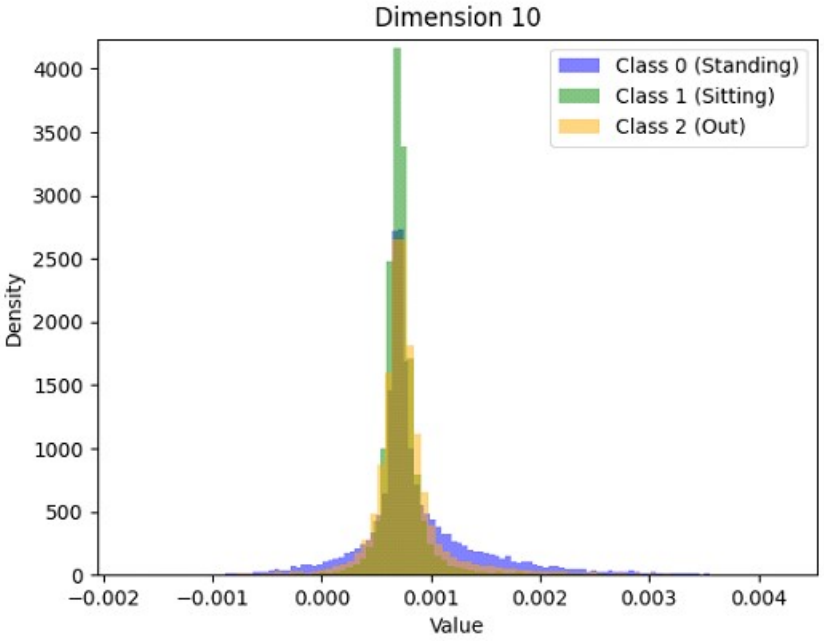

Figure F.11: Histogram of the feature 10.

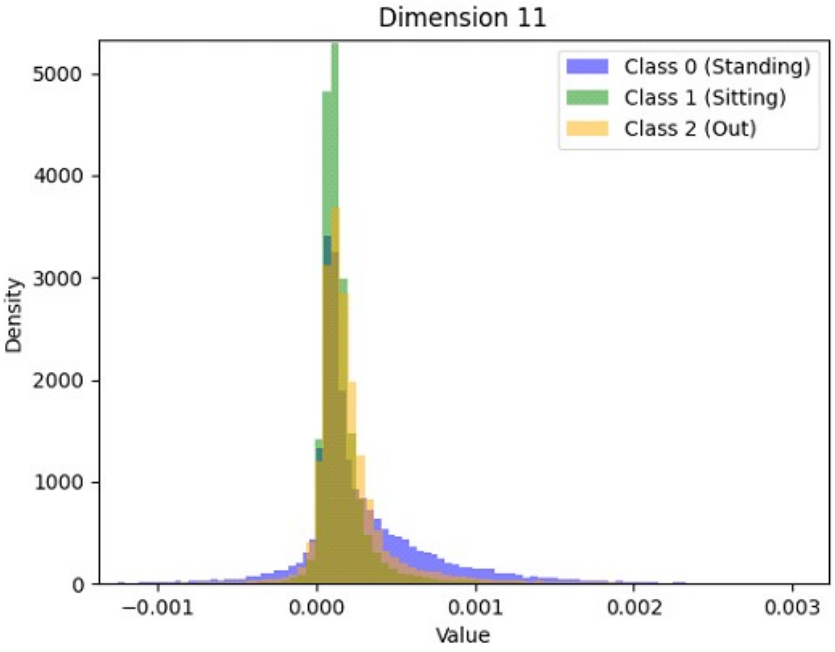

Figure F.12: Histogram of the feature 11.

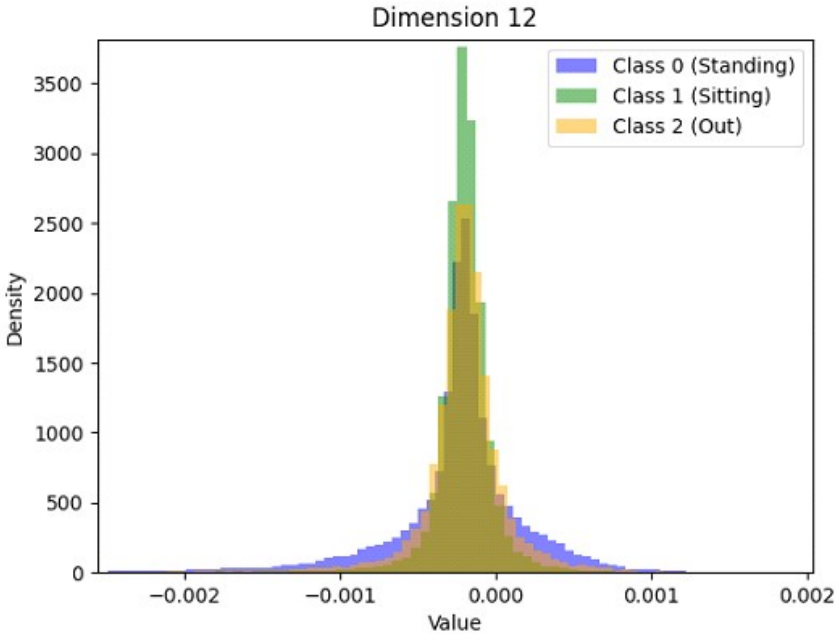

Figure F.13: Histogram of the feature 12.

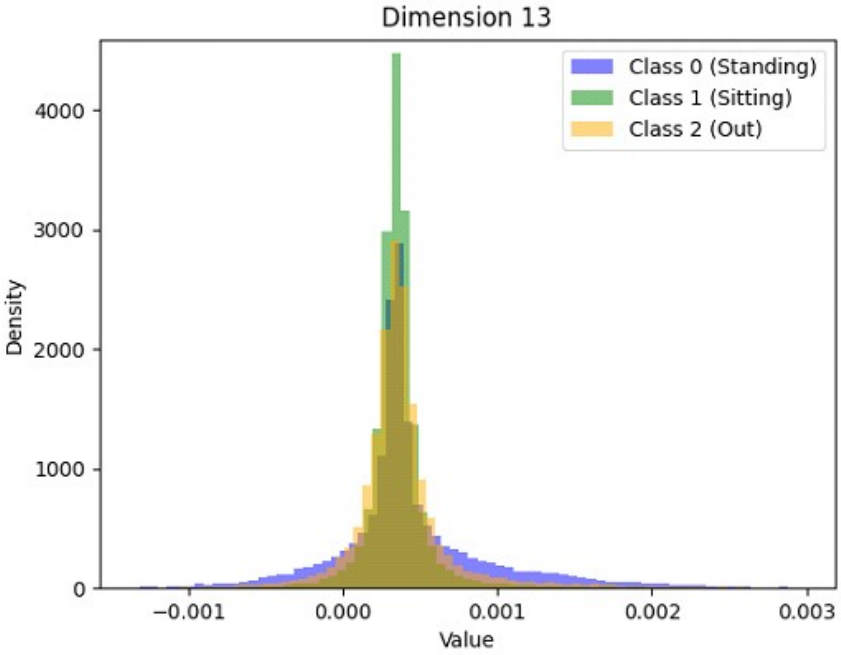

Figure F.14: Histogram of the feature 13.

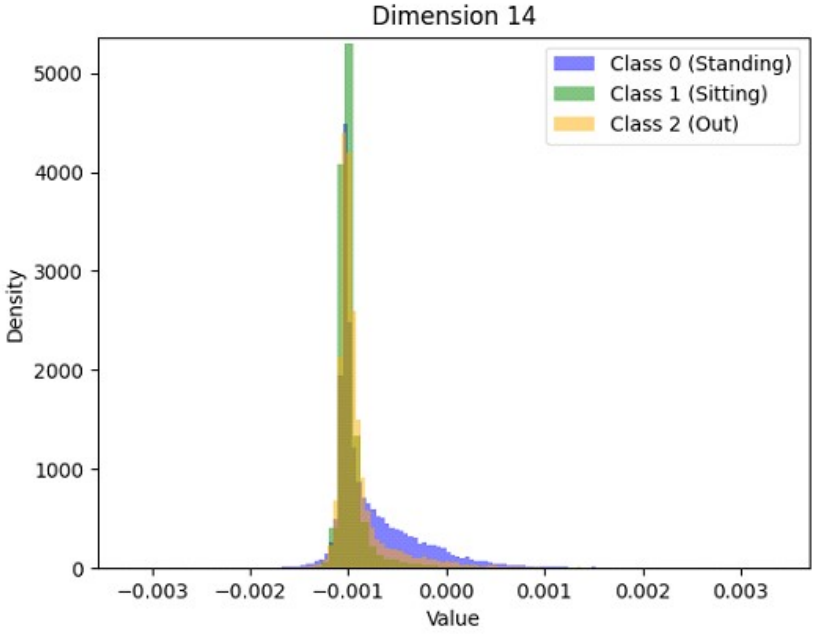

Figure F.15: Histogram of the feature 14.

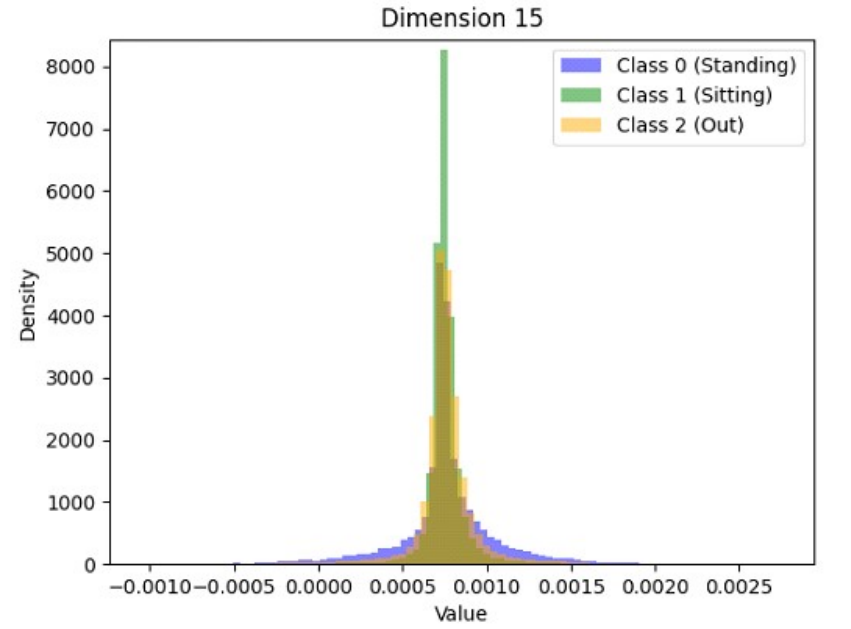

Figure F.16: Histogram of the feature 15.

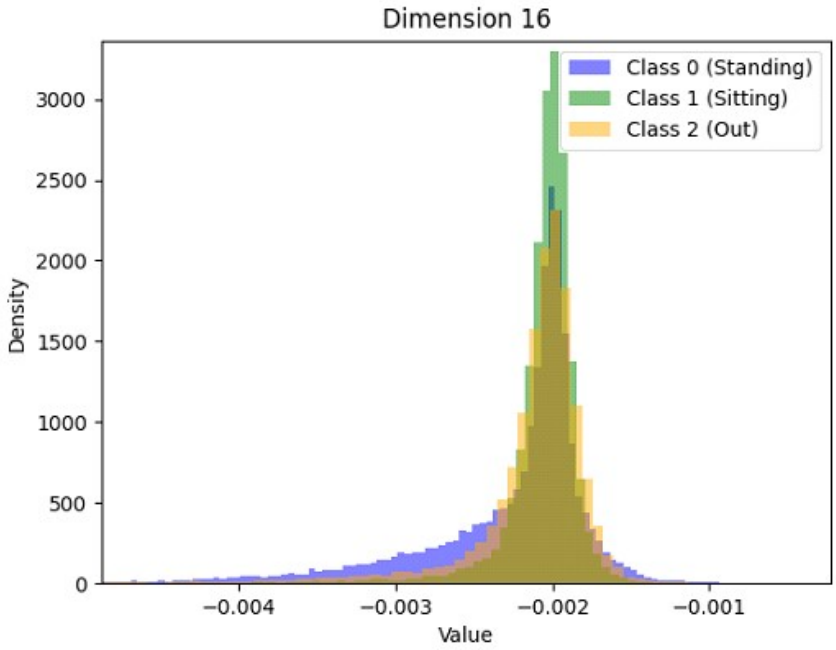

Figure F.17: Histogram of the feature 16.

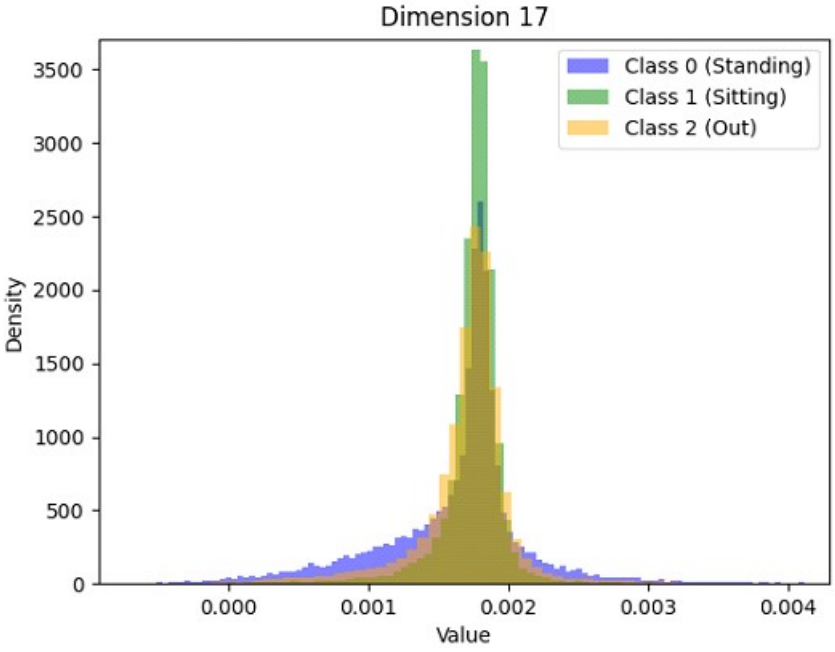

Figure F.18: Histogram of the feature 17.

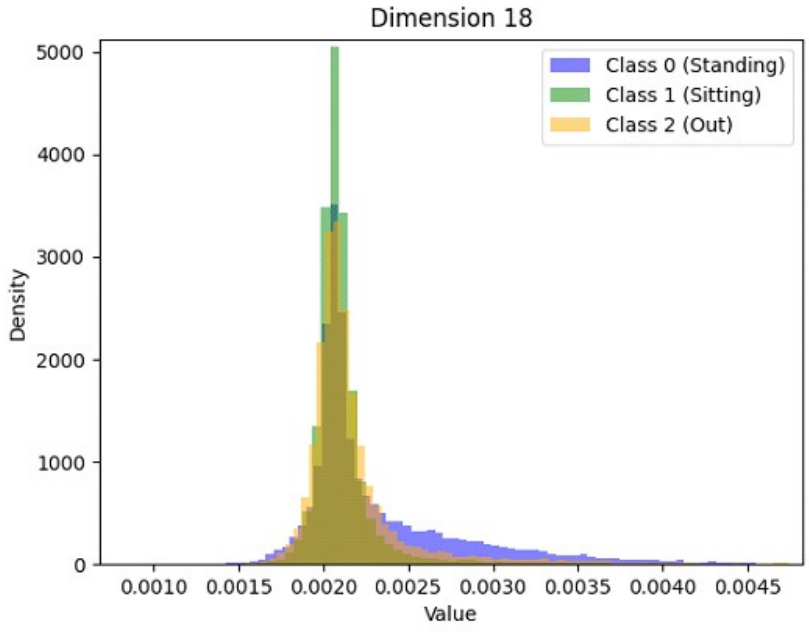

Figure F.19: Histogram of the feature 18.

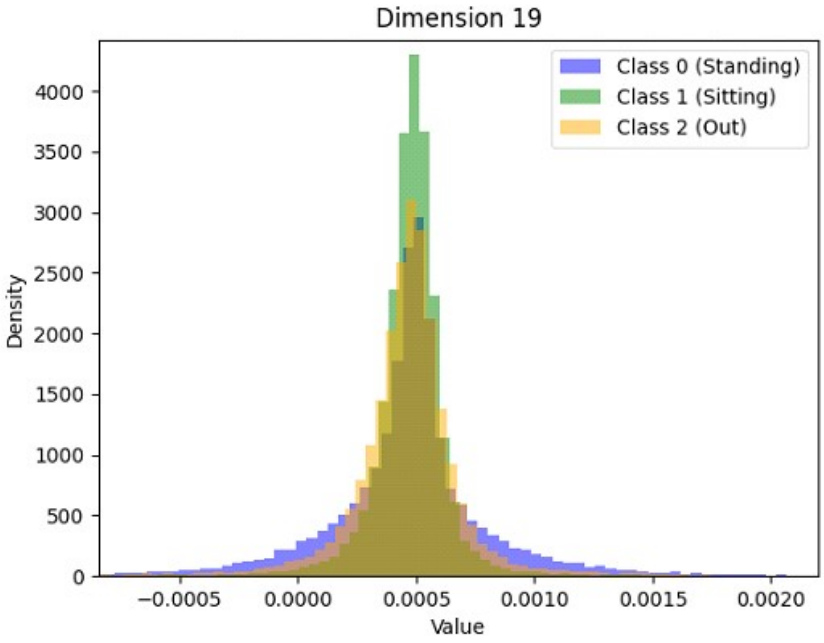

Figure F.20: Histogram of the feature 19.

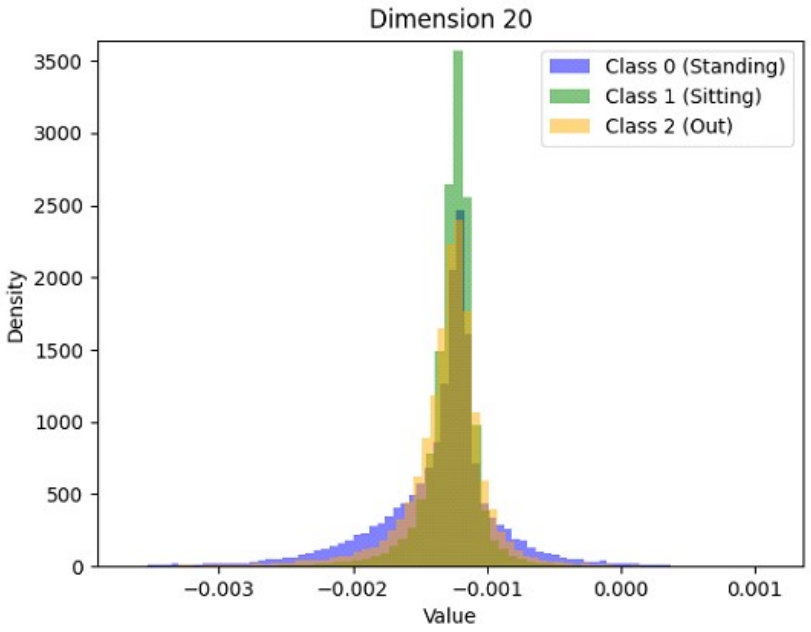

Figure F.21: Histogram of the feature 20.

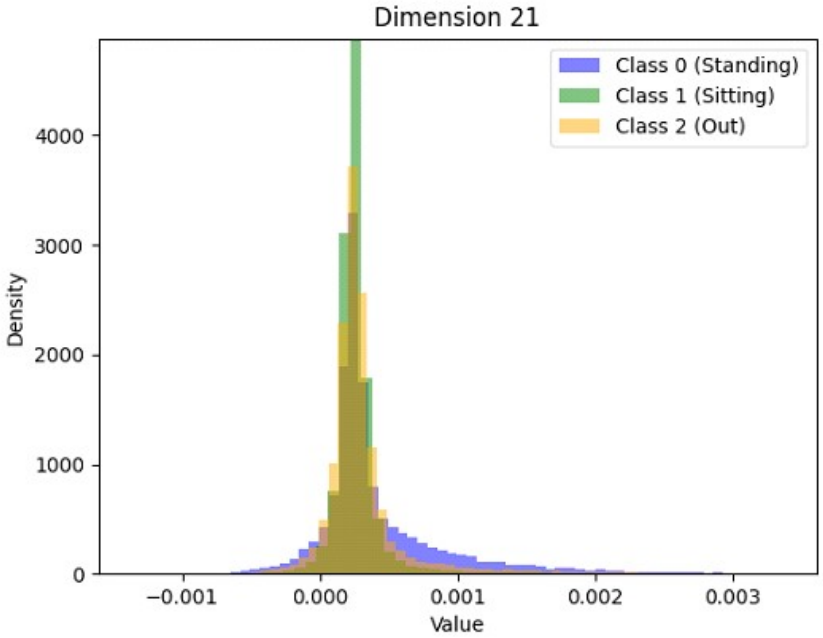

Figure F.22: Histogram of the feature 21.

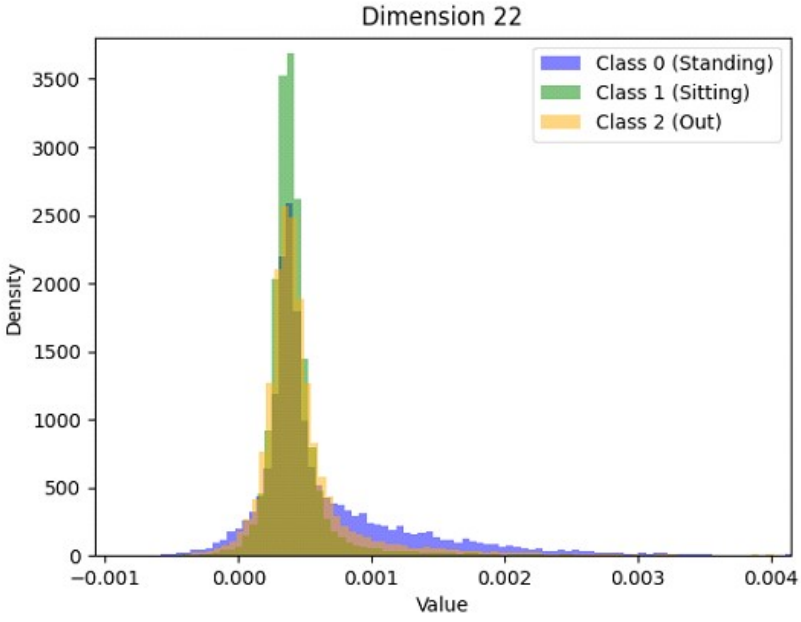

Figure F.23: Histogram of the feature 22.

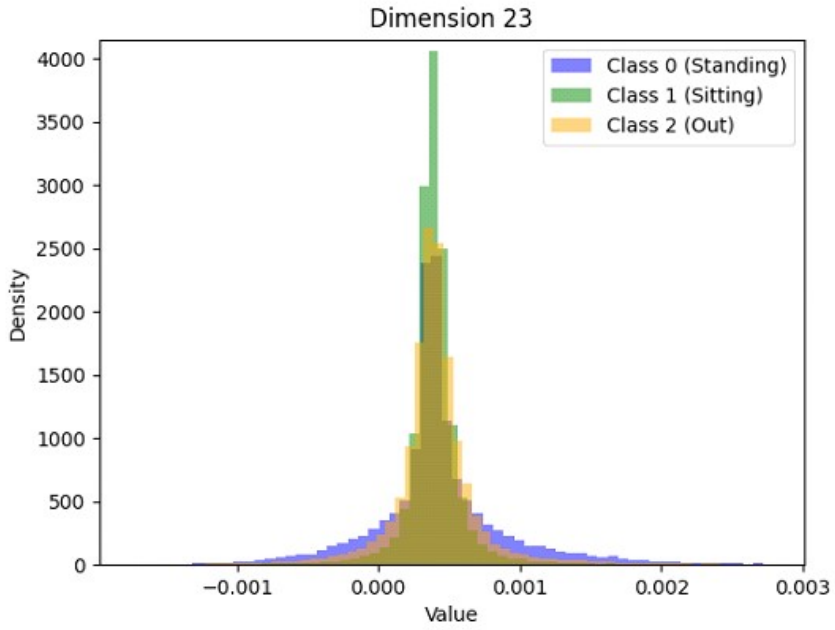

Figure F.24: Histogram of the feature 23.

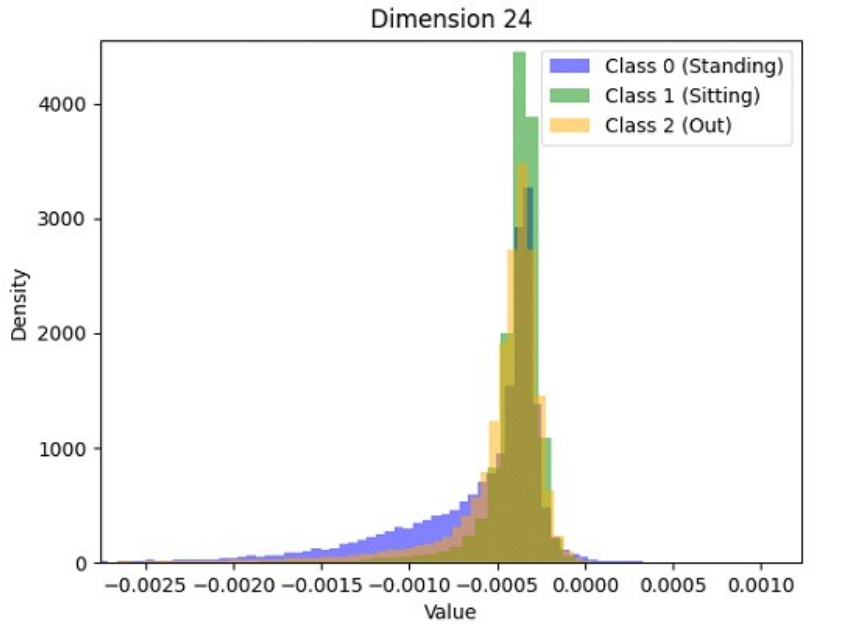

Figure F.25: Histogram of the feature 24.

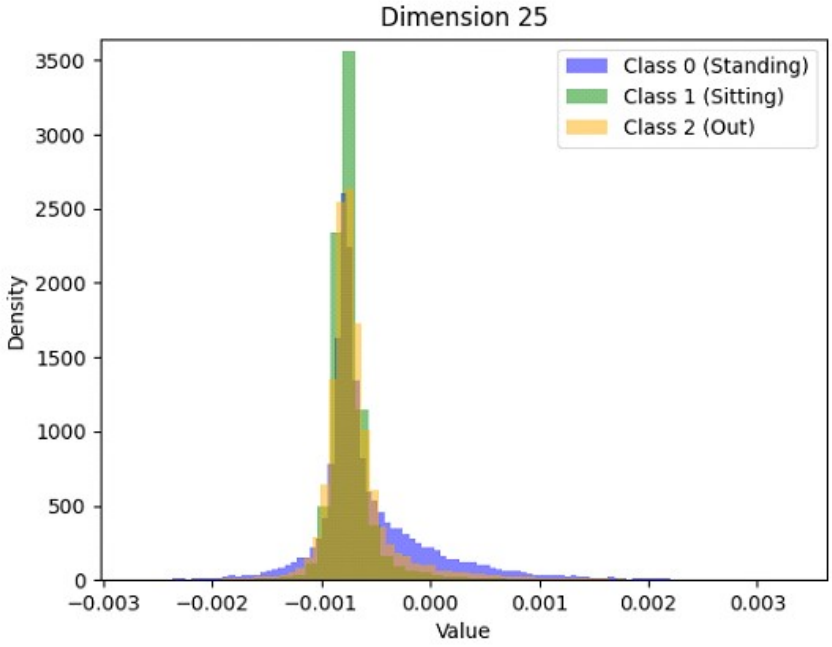

Figure F.26: Histogram of the feature 25.

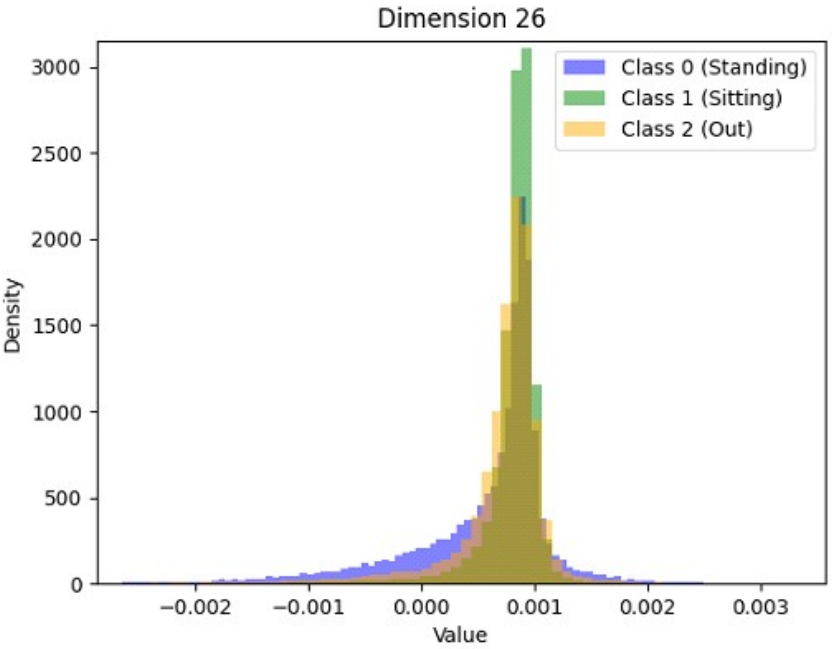

Figure F.27: Histogram of the feature 26.

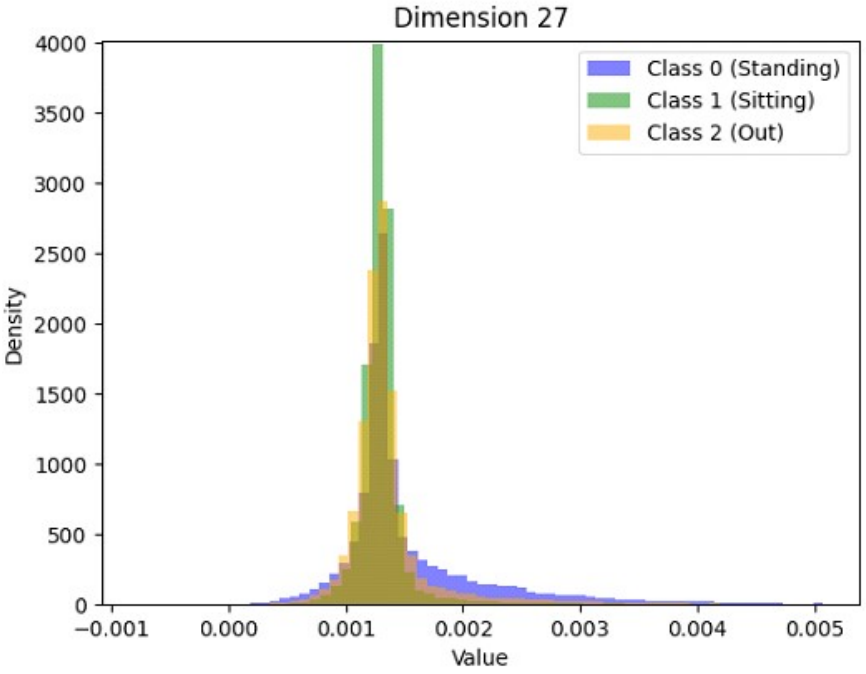

Figure F.28: Histogram of the feature 27.

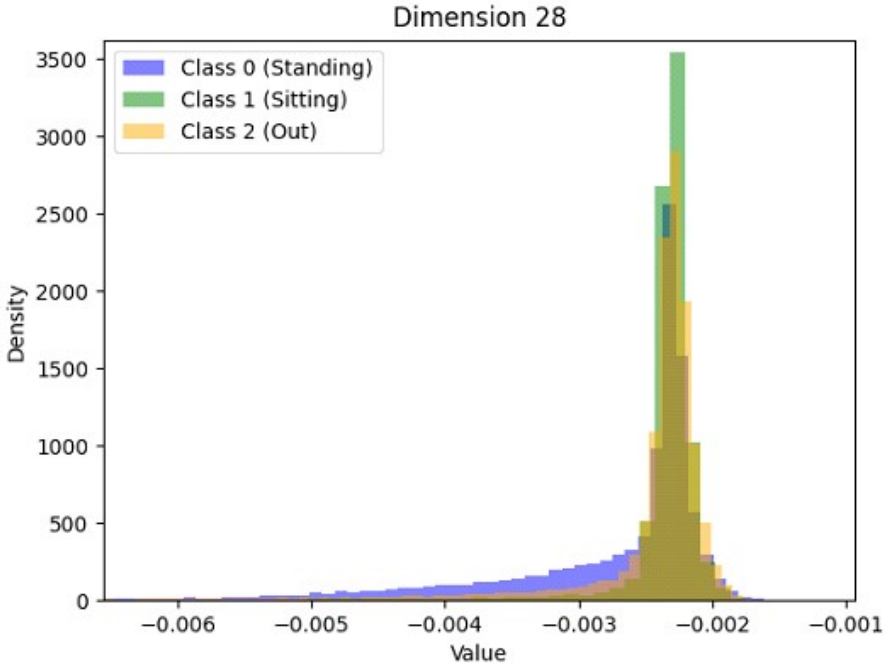

Figure F.29: Histogram of the feature 28.

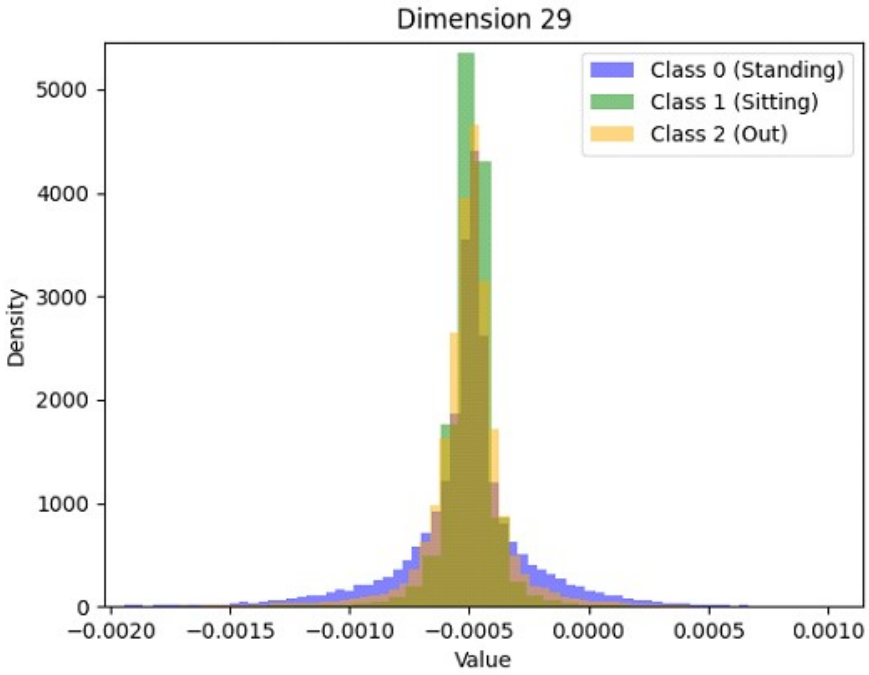

Figure F.30: Histogram of the feature 29.

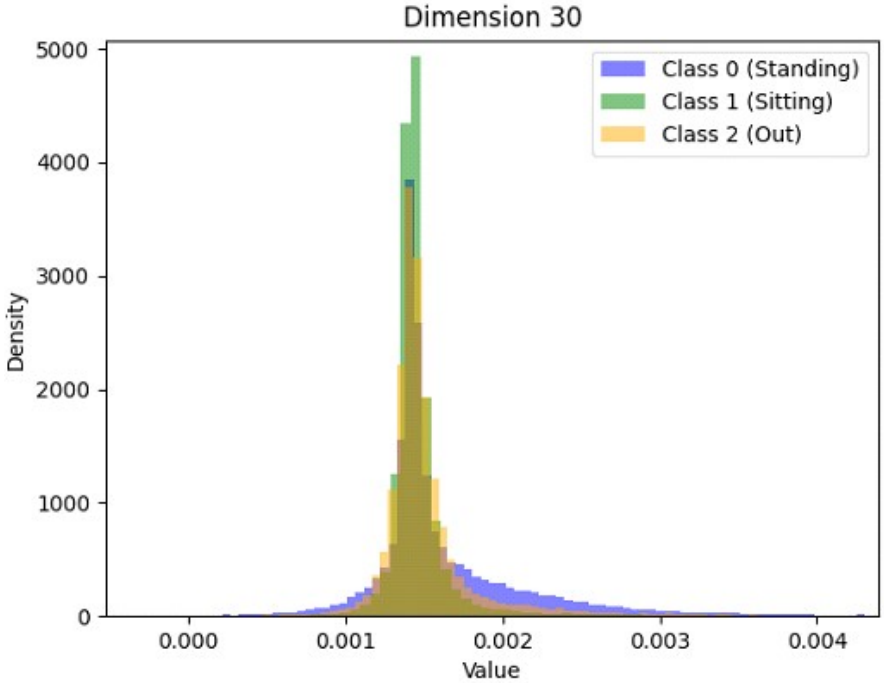

Figure F.31: Histogram of the feature 30.

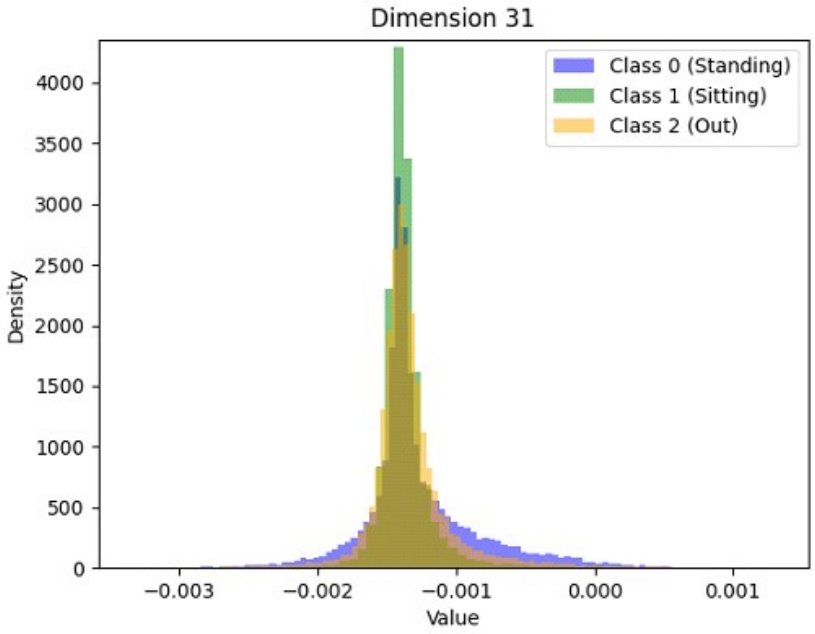

Figure F.32: Histogram of the feature 31.

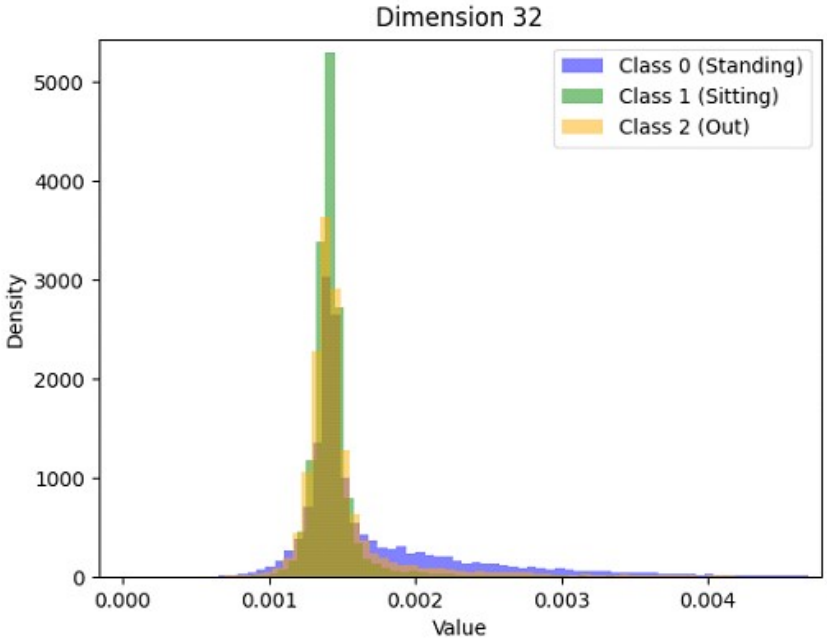

Figure F.33: Histogram of the feature 32.

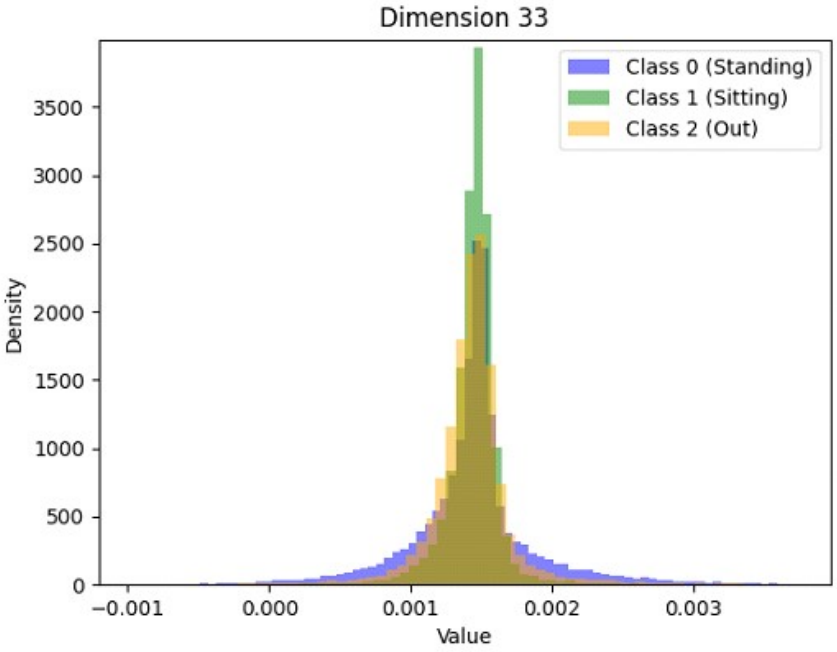

Figure F.34: Histogram of the feature 33.

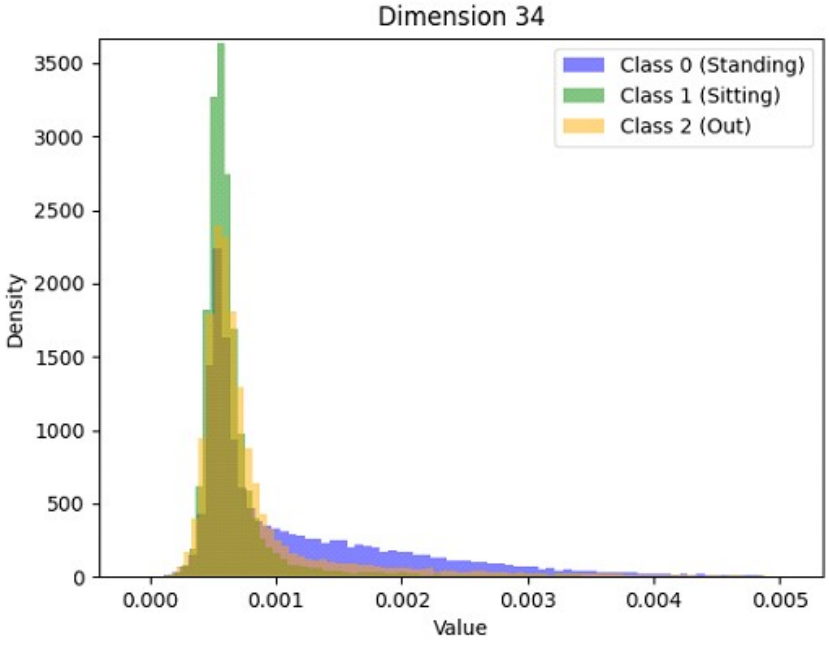

Figure F.35: Histogram of the feature 34.

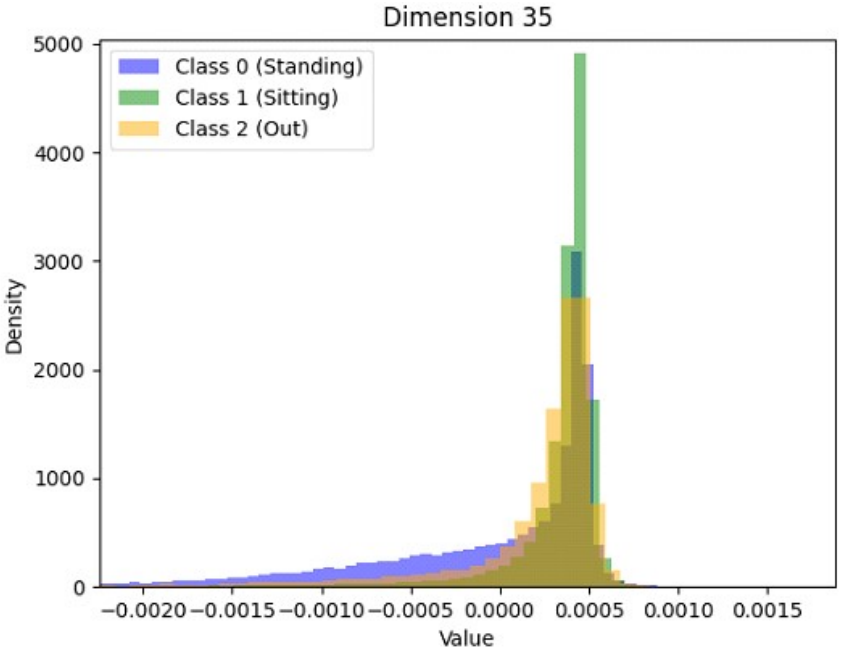

Figure F.36: Histogram of the feature 35.

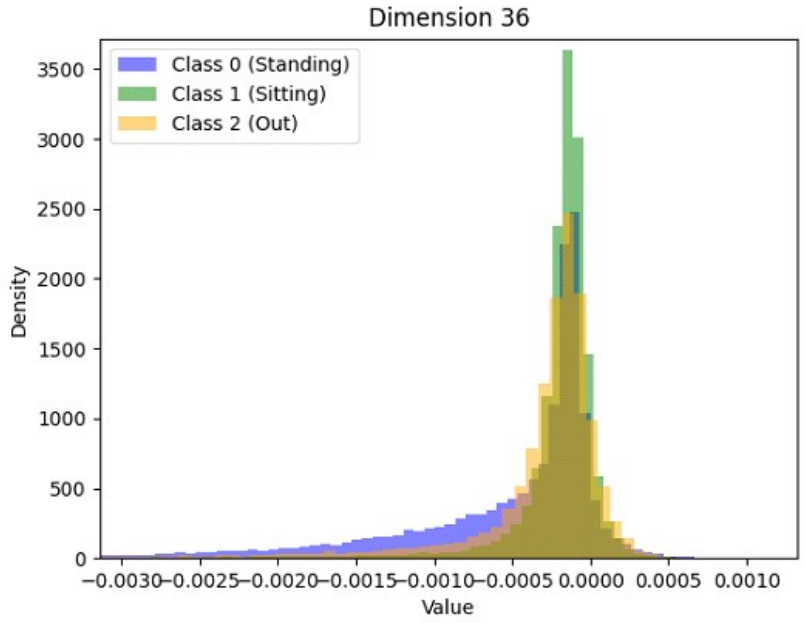

Figure F.37: Histogram of the feature 36.

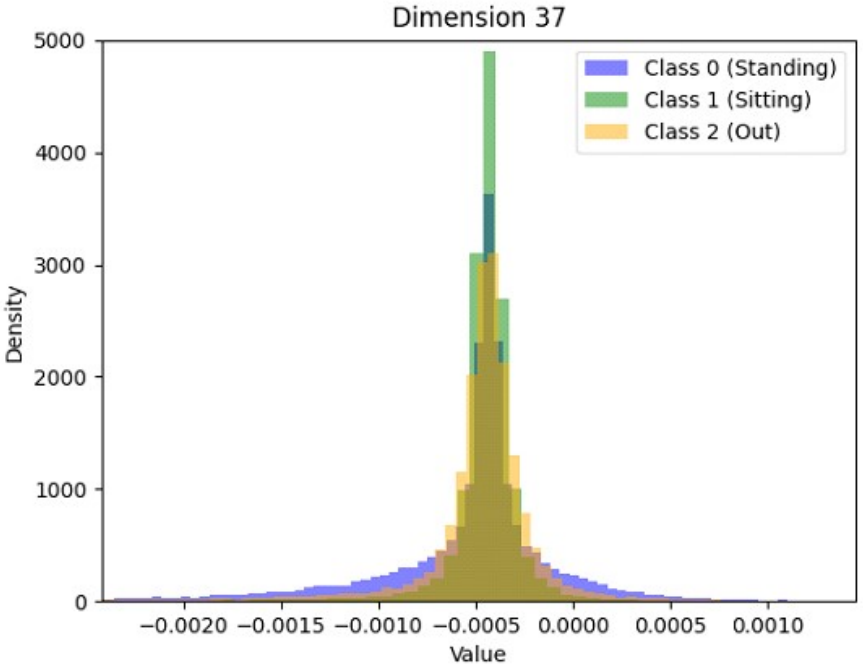

Figure F.38: Histogram of the feature 37.

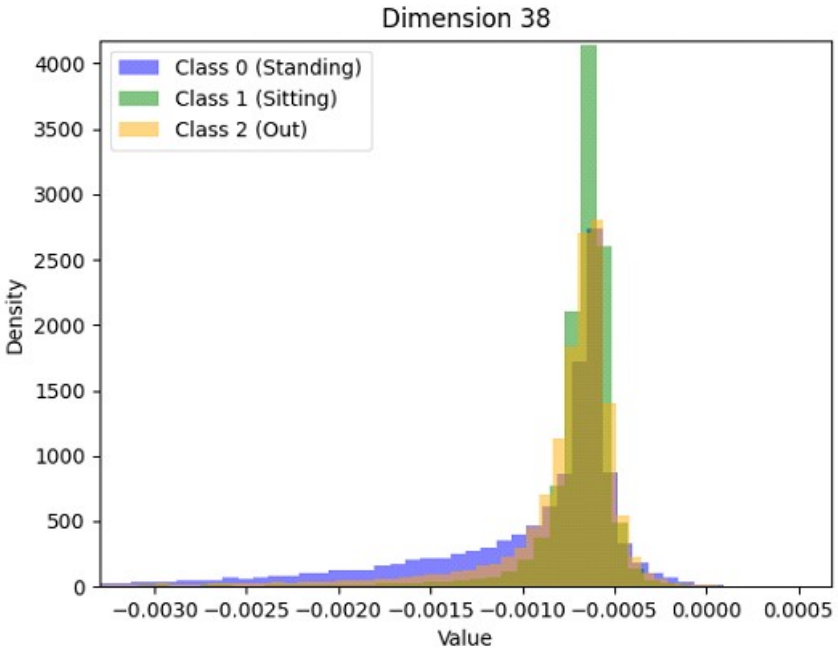

Figure F.39: Histogram of the feature 38.

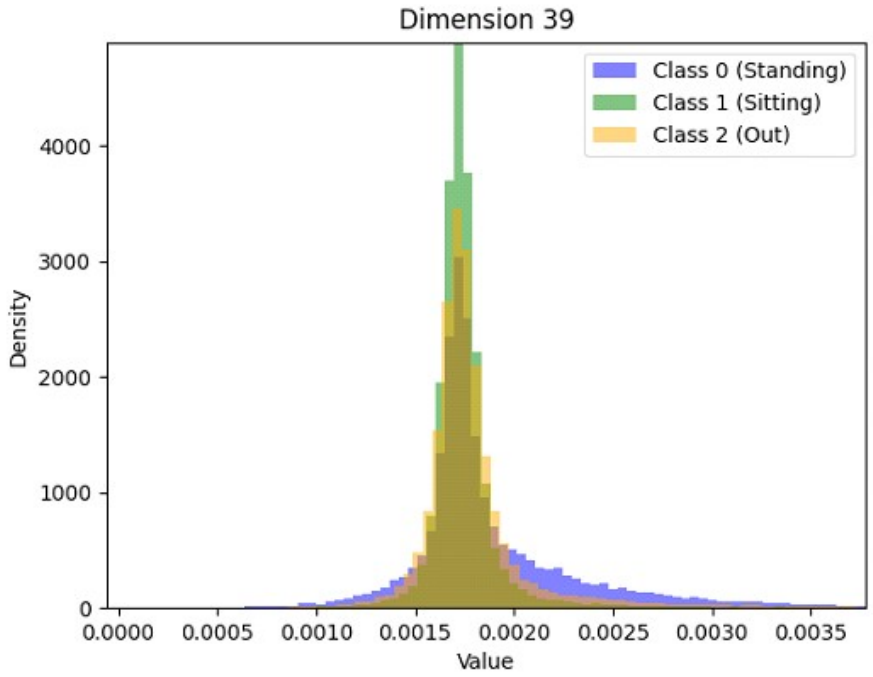

Figure F.40: Histogram of the feature 39.

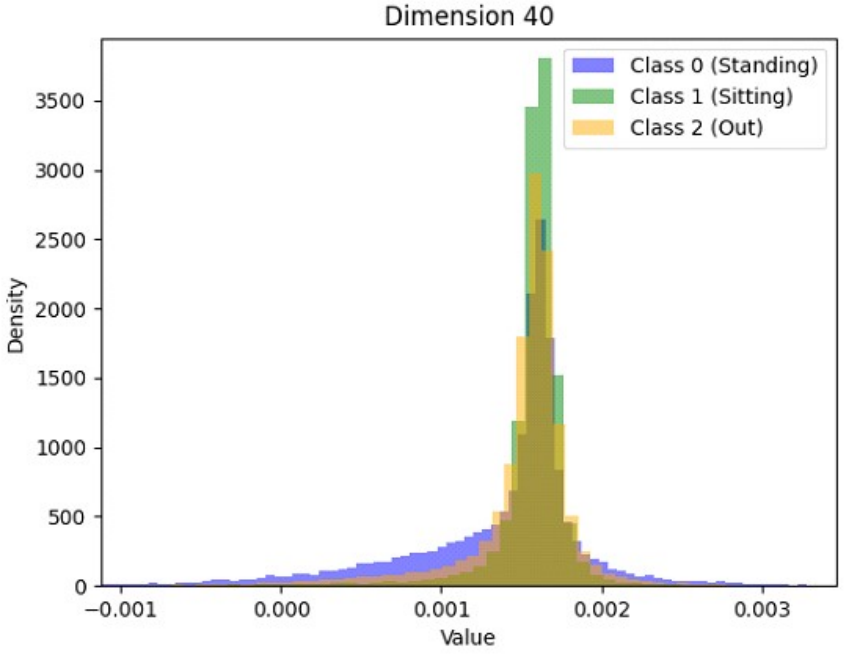

Figure F.41: Histogram of the feature 40.

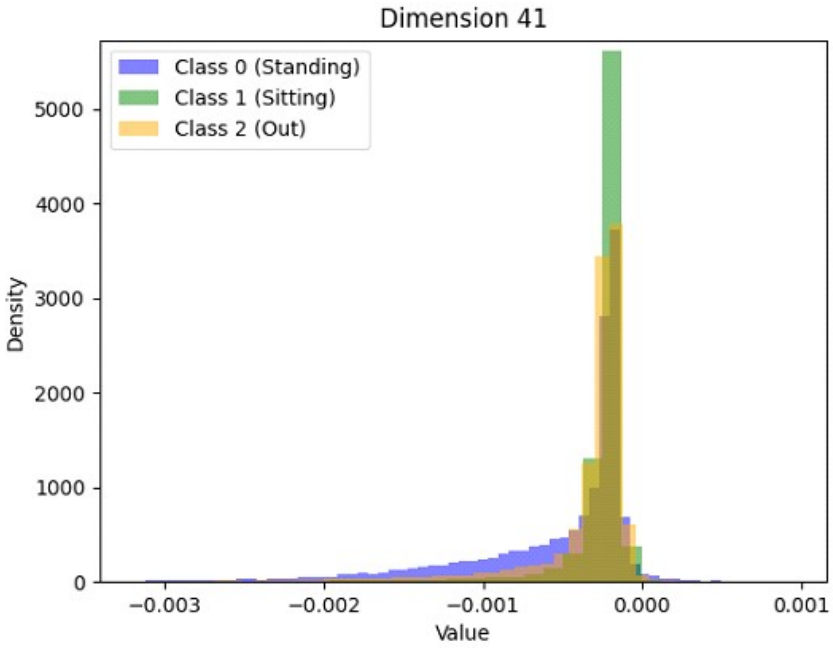

Figure F.42: Histogram of the feature 41.

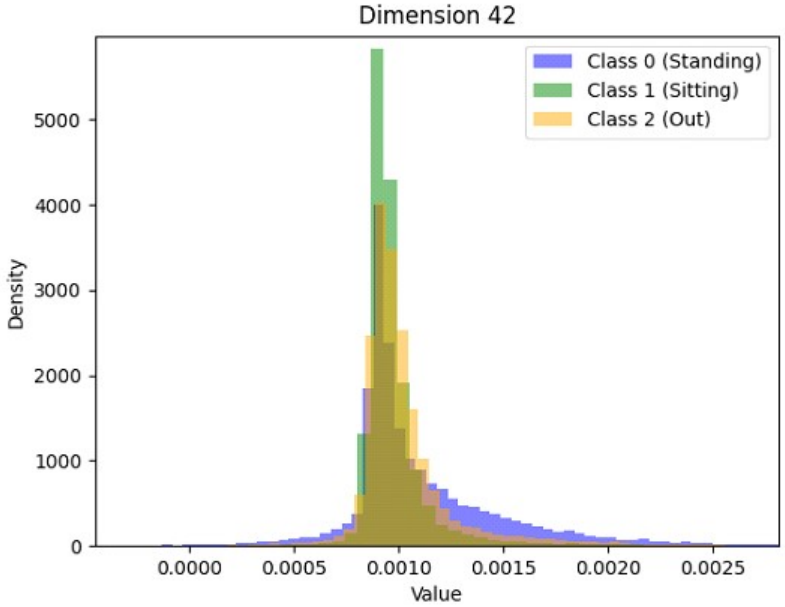

Figure F.43: Histogram of the feature 42.

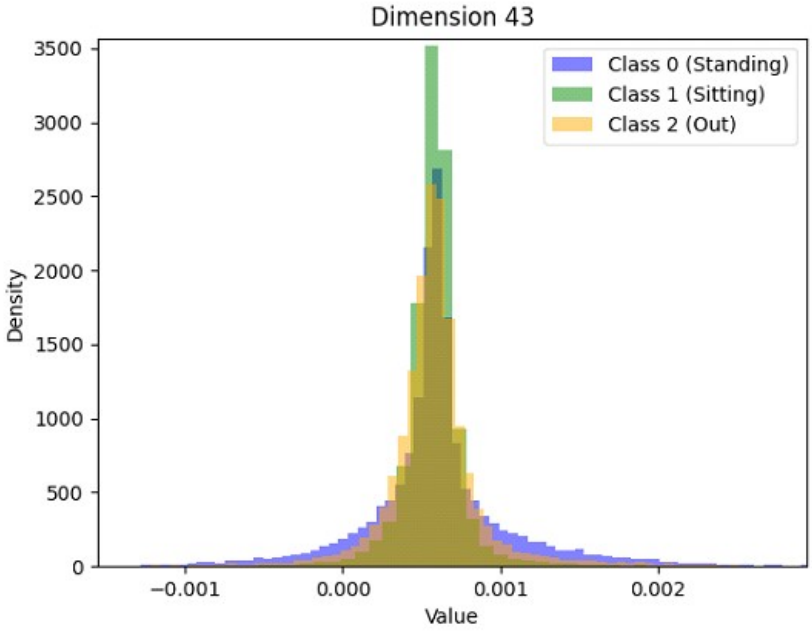

Figure F.44: Histogram of the feature 43.

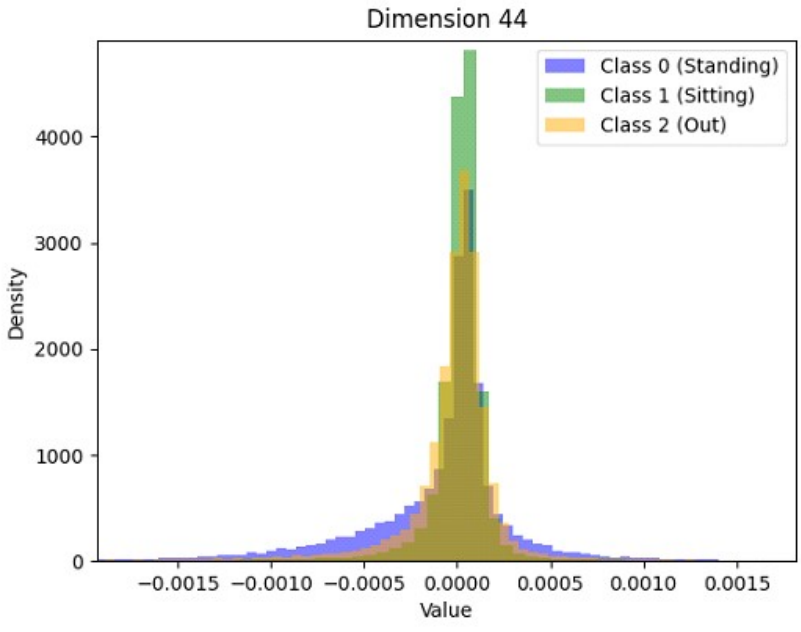

Figure F.45: Histogram of the feature 44.

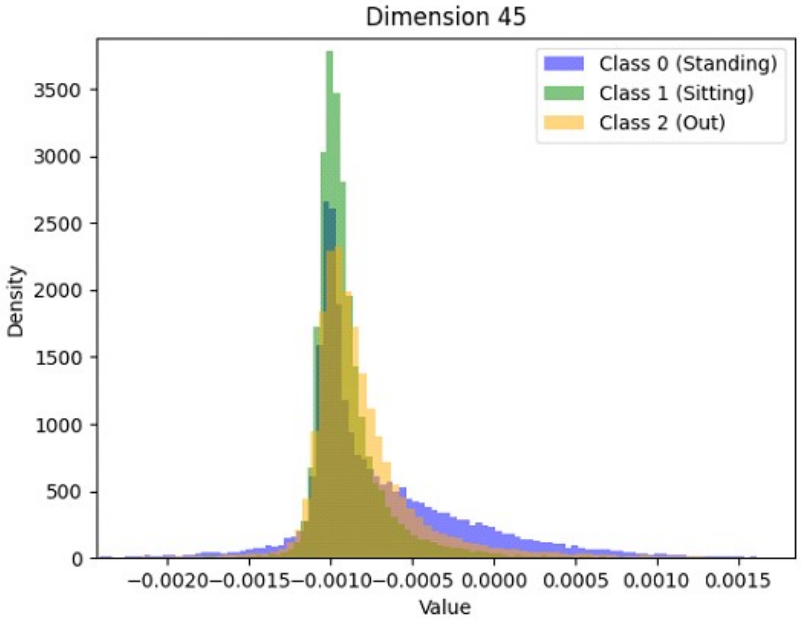

Figure F.46: Histogram of the feature 45.

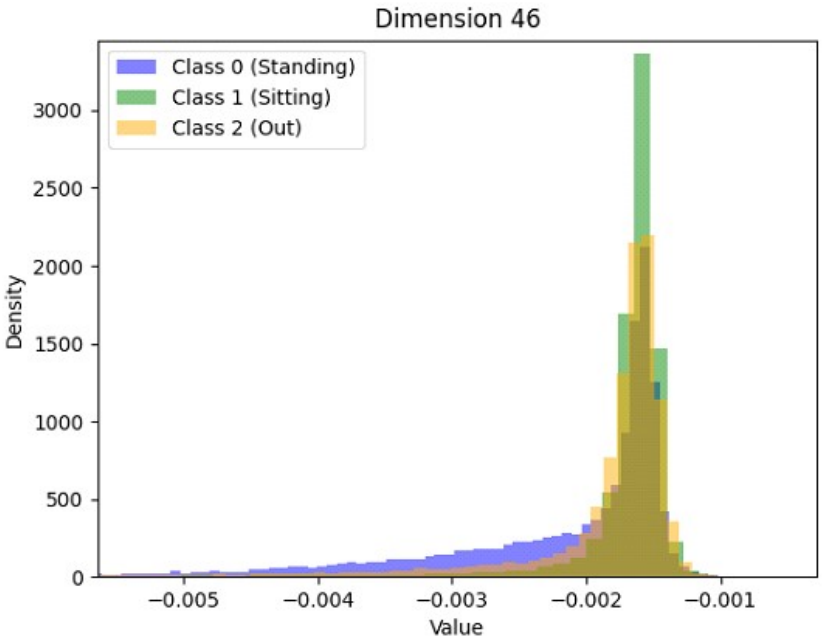

Figure F.47: Histogram of the feature 46.

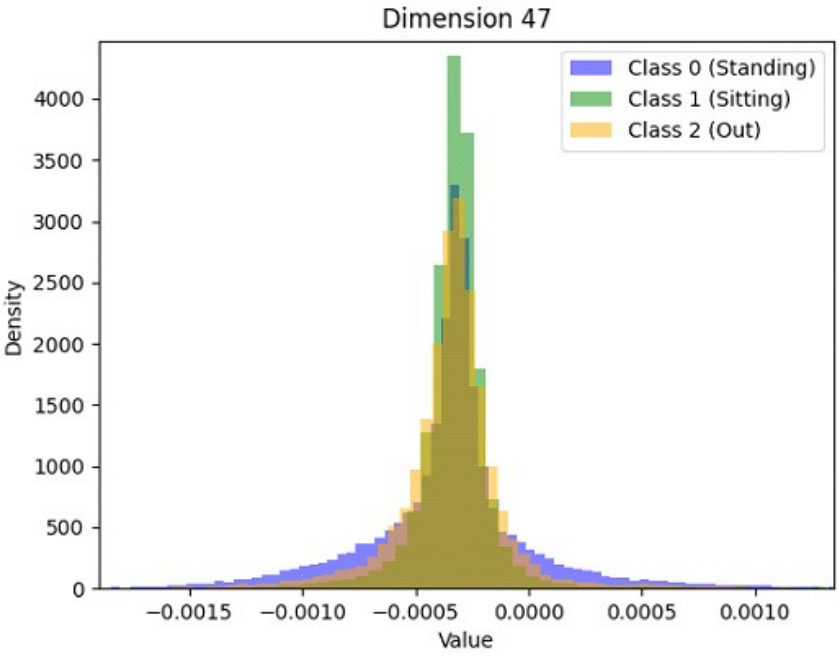

Figure F.48: Histogram of the feature 47.

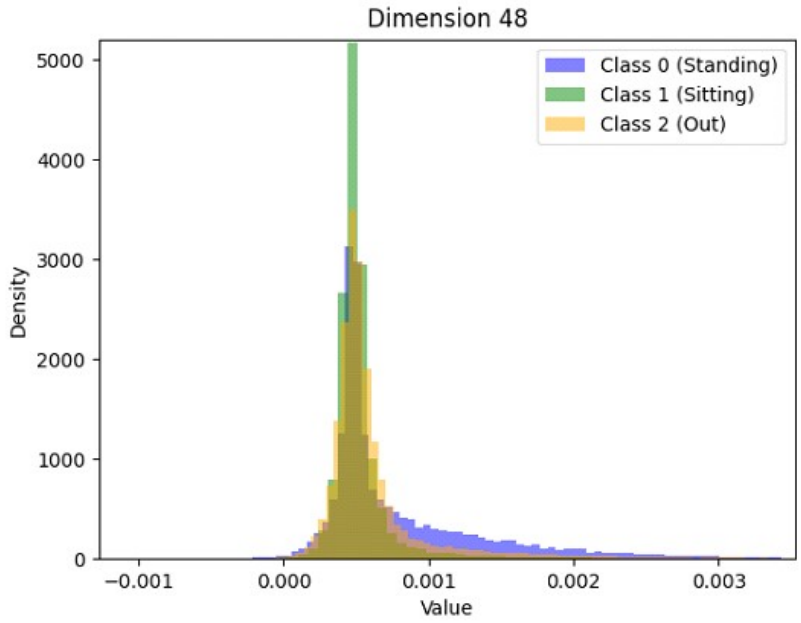

Figure F.49: Histogram of the feature 48.

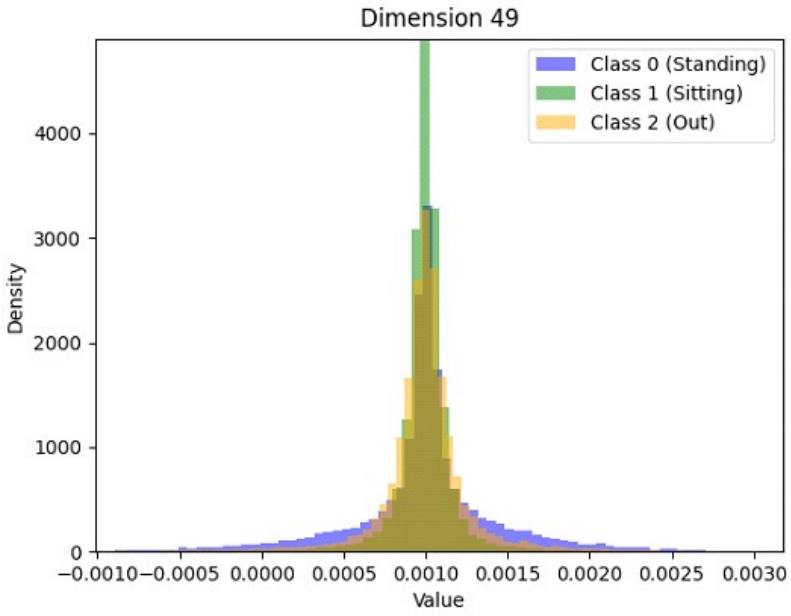

Figure F.50: Histogram of the feature 49.

Appendix 7

In this appendix, we present the histograms for all the statistical features using the complete band.

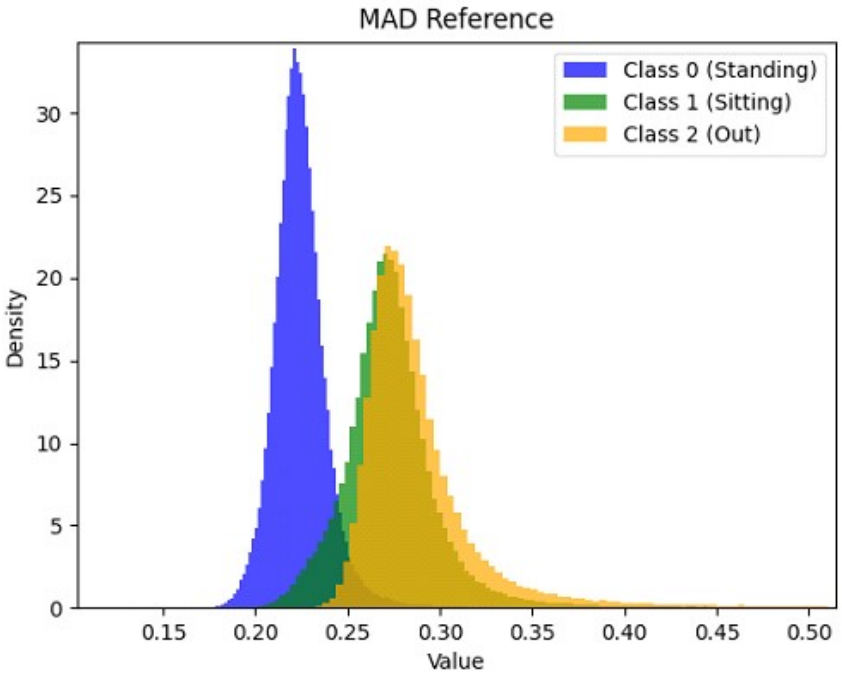

Figure G.1: Histogram of Max Absolute Reference Antenna.

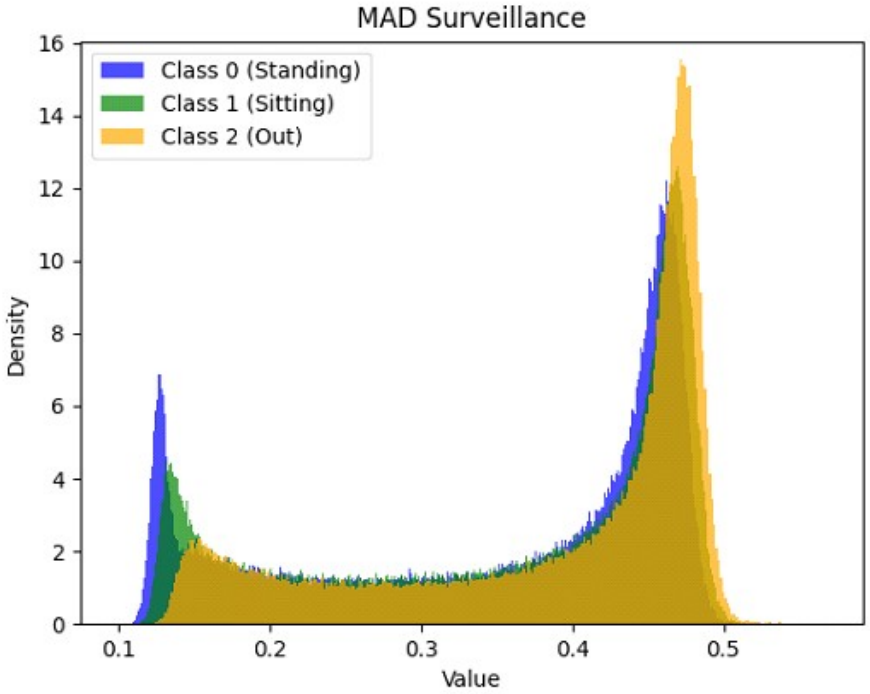

Figure G.2: Histogram of Max Absolute Surveillance Antenna.

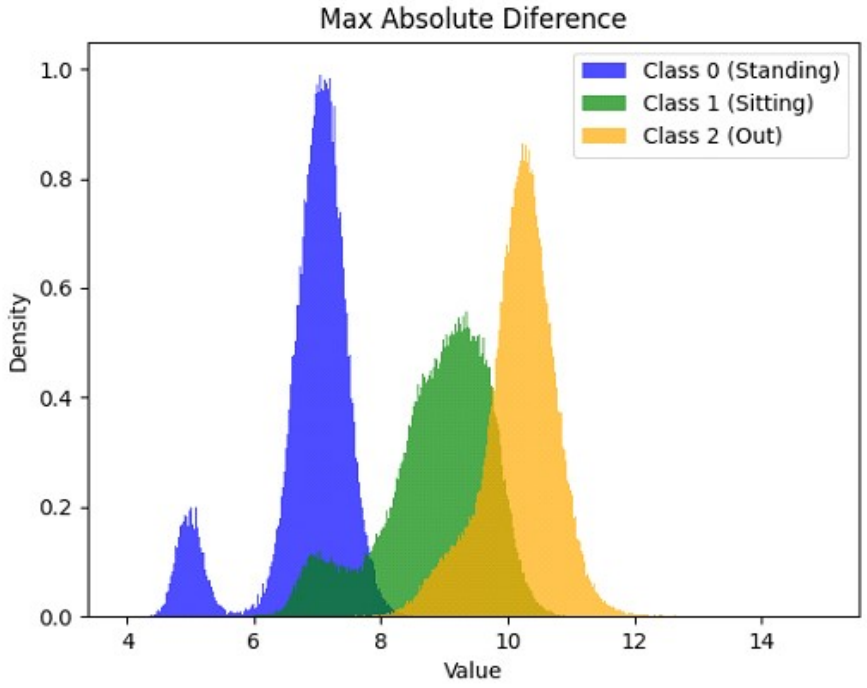

Figure G.3: Histogram of Max Absolute Difference between Antennas.

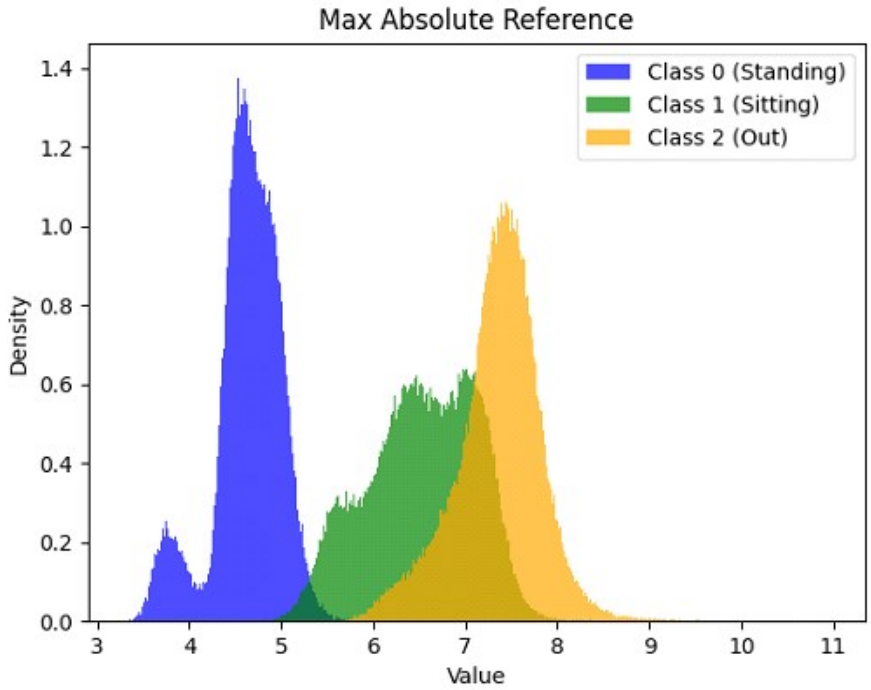

Figure G.4: Histogram of Max Absolute Reference Antenna.

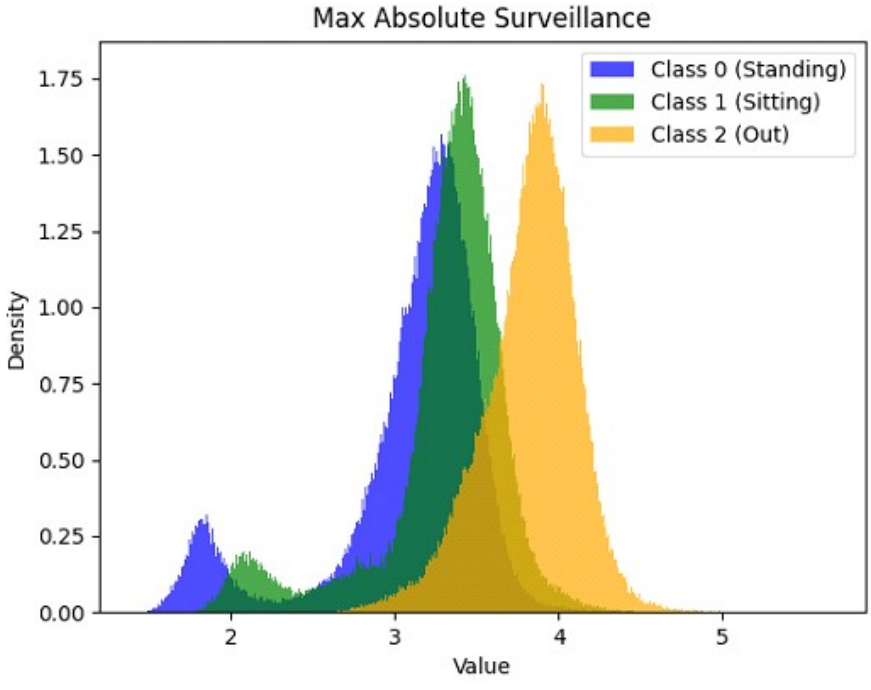

Figure G.5: Histogram of Max Absolute Surveillance Antenna.

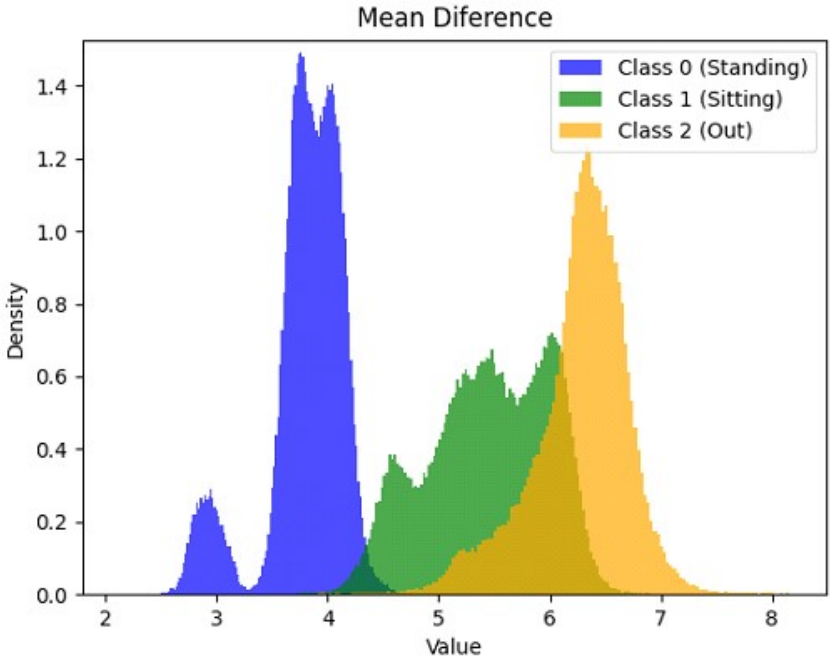

Figure G.6: Histogram of Mean Difference between Antennas.

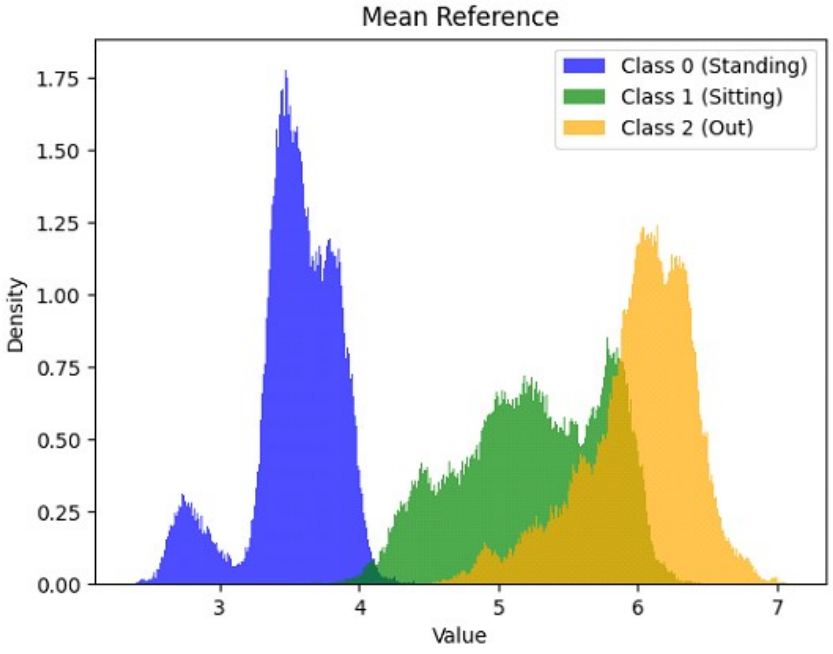

Figure G.7: Histogram of Mean Reference Antenna.

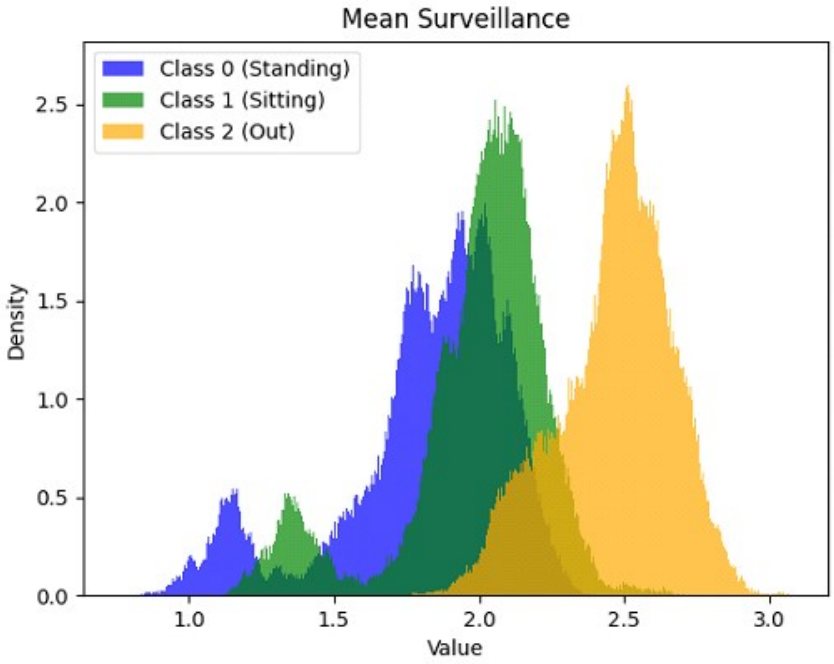

Figure G.8: Histogram of Mean Surveillance Antenna.

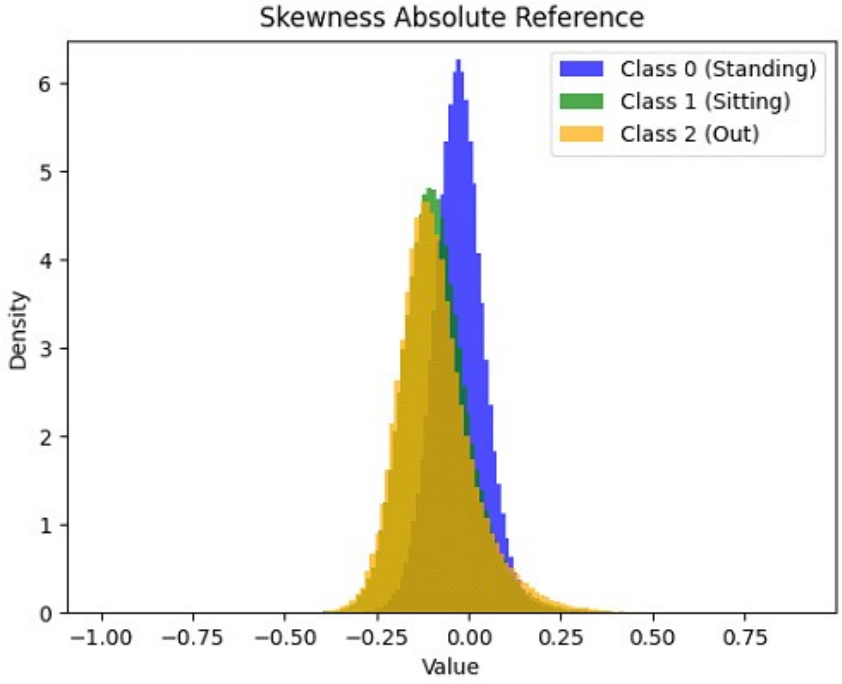

Figure G.9: Histogram of Standard deviation Absolute Reference Antenna.

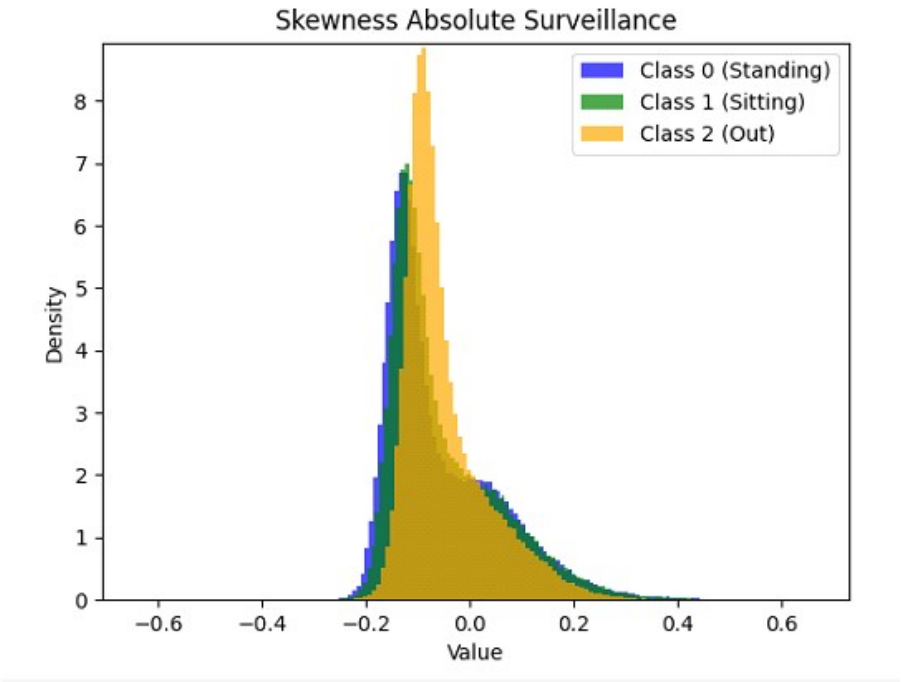

Figure G.10: Histogram of Standard deviation Absolute Surveillance Antenna.

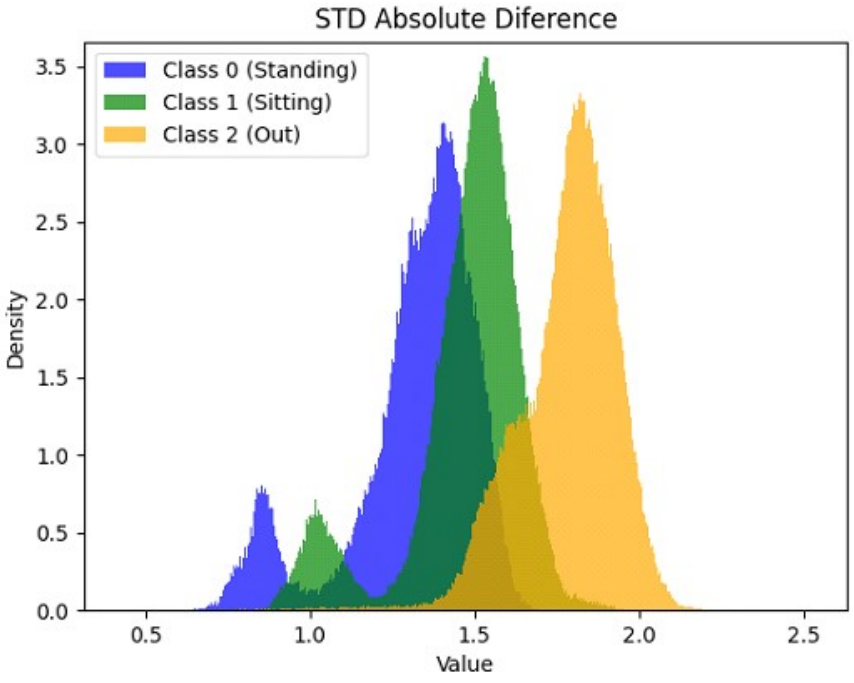

Figure G.11: Histogram of Standard deviation Absolute Difference between Antennas.

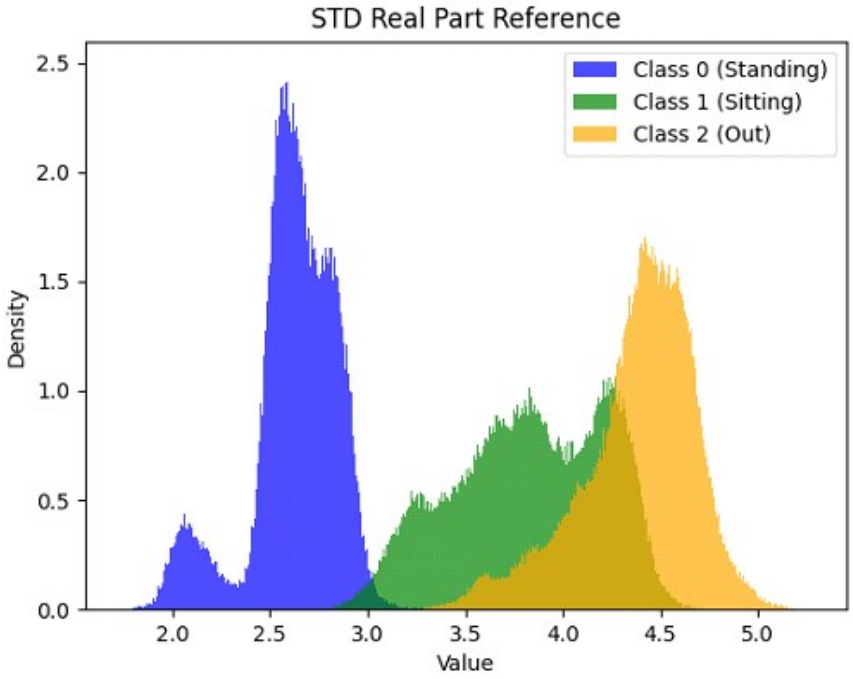

Figure G.12: Histogram of Standard deviation (Real Part) Reference Antenna.

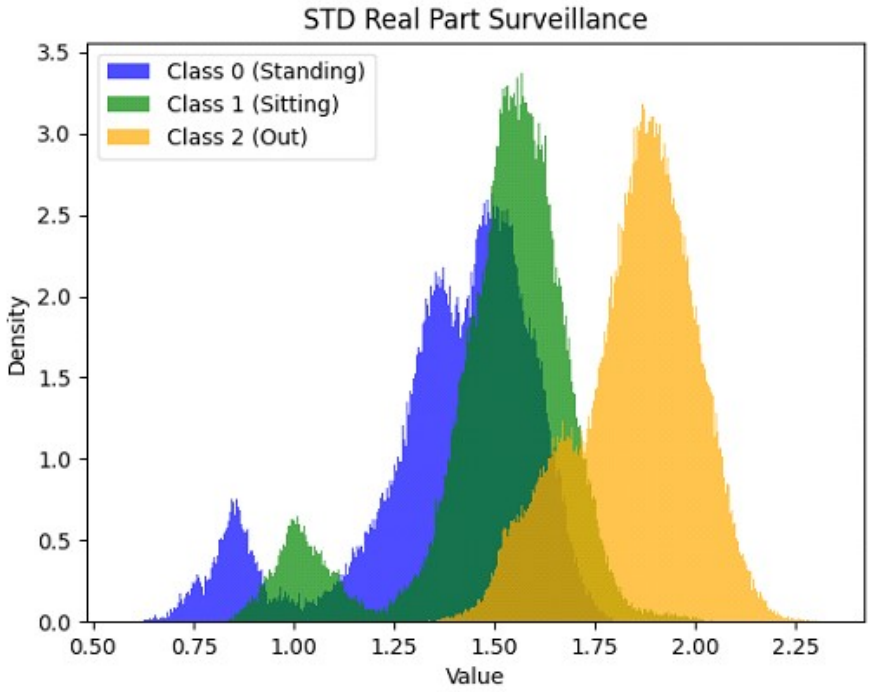

Figure G.13: Histogram of Standard deviation (Real Part) Surveillance Antenna.

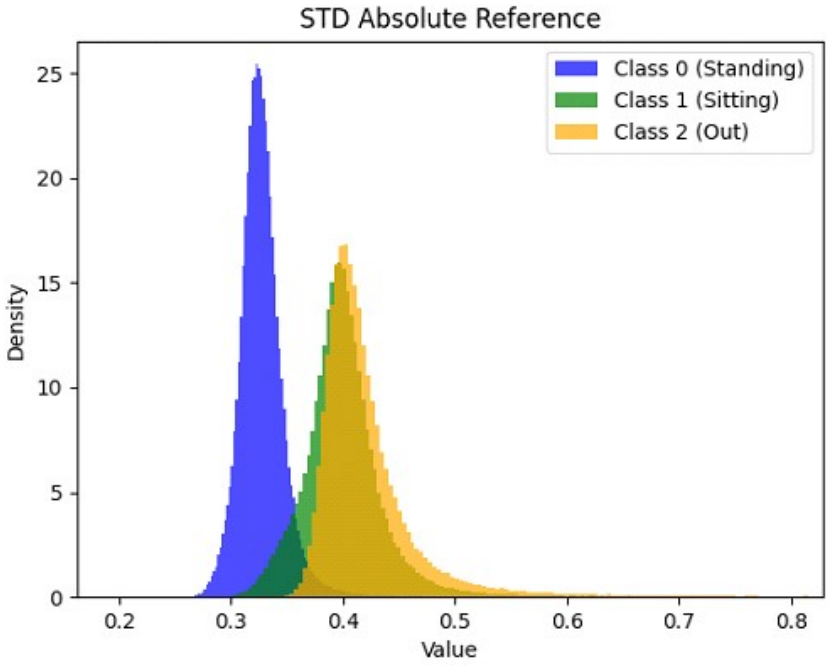

Figure G.14: Histogram of Standard deviation Reference Antenna.

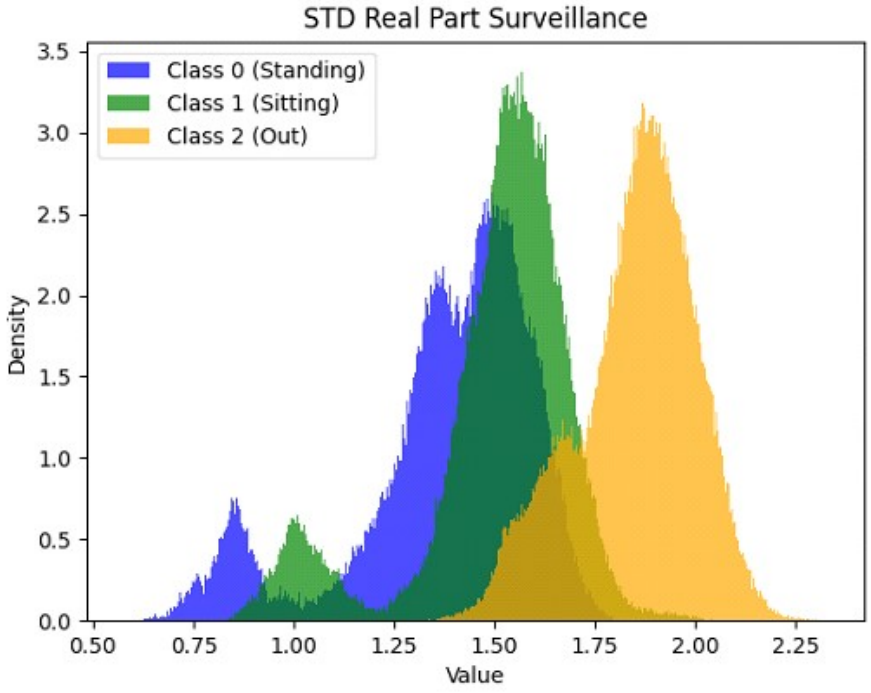

Figure G.15: Histogram of Standard deviation Surveillance Antenna.

2023 Dissertation Report

Ricardo

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY