Lucía Núñez Calvo

Application of Machine Learning for the classification of crying events in newborns as a proxy of encephalopathy

MASTER'S THESIS

Supervised by: Dr. Daniel Urda Muñoz & Dr. Santiago Marco

Master's Degree in Biomedical Data Science

Burgos, 2024

URV Faculty School of Engineering (URV)

Master's Degree in Biomedical Data Science

Dr. Daniel Urda Muñoz & Dr. Santiago Marco, certifies that the student Lucía Núñez Calvo has elaborated the work under their direction and they authorizes the presentation of this Master's Thesis for its evaluation.

Burgos, 7 June 2024

Dr. Daniel Urda Muñoz

Dr. Santiago Marco

To my father, whose love and sacrifice made my dreams possible.
Thank you for your unconditional support and for believing in me.
You have always been willing to support and motivate me, your teachings and memories continue to light my path.

Acknowledgement

First of all, I would like to express my thanks and gratitude to my Master's thesis tutors, Dr. Daniel Urda Muñoz and Dr. Santiago Marco, who have always been there to solve all my doubts, discuss, and propose different approaches to develop the project.

Besides, I would like to express my gratitude to all the professors who have provided us with new knowledge in this Master, who despite being online, have been very close helping us with everything we have needed. I would also like to extend my thanks to the GICAP research group at the University of Burgos for giving me the opportunity to participate in this innovative project, reaffirming my interest in this field, and for allowing me to concentrate on the Master to finish it as soon as possible.

Last but not least, I would like to express my gratitude to my family and friends, who have been a constant source of inspiration and a great support to me. In particular, I would like to highlight my couple, whose unconditional support and trust in me have been essential.

Abstract

Analysing how a newborn baby reacts to different stimuli is crucial to diagnose possible neurological conditions. Traditionally, these assessments have been performed manually by health professionals, which to some extent leads to subjectivity, and possible misdiagnosis. In addition, such assessments require the immediate availability of a specialist, which in critical situations can delay assessments, putting the baby's health at risk.

This project uses advanced tools such as **Deep Learning**, and **Machine Learning** to automate, and improve the detection, and **classification of newborn cries** in order to detect possible pathologies or diseases. Artificial Intelligence analyses of newborn videos, provides a consistent, and accurate assessment of newborn responses to stimuli can be achieved, accelerating the diagnosis of a possible disease.

Specifically, this study focuses on the detection of Hypoxic-Ischaemic Encephalopathy (**HIE**) by assessing newborn crying in response to nociceptive stimuli. To achieve this goal, several audio analysis approaches have been applied, with the project focusing on the employing of Machine Learning models, trained on labelled cry data, which have demonstrated high accuracy in the recognition of cry patterns.

This study explores feature extraction techniques such as Mel-Frequency Cepstral Coefficients (**MFCC**), and Linear Predictive Coding (**LPC**), and has implemented models such as Multilayer Perceptron (**MLP**), Support Vector Machine (**SVM**), and Long Short-Term Memory (**LSTM**). Promising results have been achieved with these models, with accuracies of up to 90%.

The above methods not only help to detect HIE, but also open up the possibility of diagnosing other health problems through cry analysis. This research thus highlights the potential of machine learning to improve paediatric diagnosis, and neonatal care.

Keywords: Deep Learning, Machine Learning, Audio analysis, Classification of crying, Newborns, Mel Frequency Cepstral Coefficients (MFCC), Linear predictive coding (LPC), Multilayer Perceptron (MLP), Support Vector Machine (SVM), Long Short-Term Memory (LSTM), Hypoxic Ischaemic Encephalopathy

Overview

Analysing how a neonate reacts to a stimulus is crucial for a proper diagnosis and treatment of the possible neurological disease it may be suffering from. Historically, such assessments have been performed manually by healthcare staff, basing the classification of the stimulus on their judgement, and interpretation of the reactions observed in the newborn. This approach can sometimes present problems. On the one hand, the assessment can be subjective, and vary among different health specialists, which can lead to erroneous diagnoses. On the other hand, as these are situations that require rapid action, if at the critical moment there is no doctor or specialist available for this specific disease, the assessment of the newborn can be complicated or delayed, putting his or her health at risk.

To solve these problems, advanced tools are now available, such as **Deep Learning** and **Machine Learning**, which have the main feature of being able to automate, and improve processes such as the detection, and classification of certain diseases. Through the collection of videos of newborns, and Deep Learning techniques with which large volumes of data can be analysed, a concise, and uniform assessment of the state of health of babies can be achieved. This avoids inconsistencies in classifications, and increases the accuracy, and speed of diagnosis.

This study aims to provide new knowledge for the detection of **Hypoxic Ischaemic Encephalopathy (HIE)** by assessing only the sound stimuli that a newborn generates. Several **audio analysis** approaches can be applied to solve this problem, such as a spectral analysis in which the sound is decomposed to analyse its frequency components. An extraction of temporal features that may include the duration of the cry, the intervals between cries or the intensity patterns can be carried out. Convolutional Neural Networks (CNNs) can also be used to classify the different types of cries according to the audio characteristics and, Deep Learning, and Machine Learning models can also be applied. With proper training, these

models may be able to recognise complex patterns in audio data, and **classify different types of crying** with good accuracy based on labelled crying datasets.

Therefore, the purpose of this study is to explore the ability of Machine Learning and Deep Learning to identify, and classify the sound responses produced by a neonate to a stimulus by analysing its cry. To achieve this goal, we have opted for feature extraction using techniques such as Mel Frequency Cepstral Coefficients (**MFCC**) or Linear Predictive Coding (**LPC**) and then create different models such as Multilayer Perceptron (**MLP**), Support Vector Machine (**SVM**), and Long Short-Term Memory (**LSTM**). These models were trained first with a public dataset, to see if the results were promising. Subsequently, these same models have been trained with project-specific audios, thus achieving an accurate classification of the different sounds that can be found present in the audios available for this study, with accuracies of up to **90%**.

This approach not only facilitates the detection of HIE, but also leaves the possibility that this technique can be applied to the identification of other health conditions through the response to crying. In this way this research presents a new avenue for paediatric diagnosis with the help of innovative techniques, and the importance of machine learning in improving neonatal disease detection and care.

Keywords: Deep Learning, Machine Learning, Audio analysis, Classification of crying, Newborns, Mel Frequency Cepstral Coefficients (MFCC), Linear predictive coding (LPC), Multilayer Perceptron (MLP), Support Vector Machine (SVM), Long Short-Term Memory (LSTM), Hypoxic Ischaemic Encephalopathy

Table of Contents

List of Figures

List of Tables

Abbreviations and acronyms

HIE Hypoxic Ischemic Encephalopathy

HUBU University Hospital of Burgos

LFCC Linear Frequency Cepstral Coefficients

LPC Linear Predictive Coding

LPCC Linear Predictive Cepstral Coefficients

LSTM Long Short-Term Memory

MFCC Mel Frequency Cepstral Coefficients

MLP Multilayer Perceptron

PCA Principal Component Analysis

RBF Radial Basis Function

SVM Support Vector Machine

t-SNE T-distributed Stochastic Neighbor Embedding

UMAP Uniform Manifold Approximation and Projection

Chapter 1

Introduction

1.1 Problem overview and relevance

Crying is the main way babies communicate in their first months of life, so it is one of the ways they express their needs, emotional states or discomfort. For this reason, it is very important to learn to understand what babies are trying to express in order to be able to react, and respond effectively to their requests. This is the main basis for an accurate detection of **Hypoxic Ischemic Encephalopathy** (HIE) in newborns and if positive, to what degree. This encephalopathy is one of the most common and is caused by a lack of oxygen to the baby's brain at some point during pregnancy or delivery [1]. Its detection is complex, and for a complete and accurate diagnosis, the following tests, and examinations must be performed on the newborn to assess its state of health:

- Different scoring scales, an Apgar[1] test which assesses different factors in the development of any newborn and a Sarnat[2] test which is an assessment of neonatal encephalopathy.

[1] Rapid test performed at one and five minutes after the birth of the baby.[2]
[2] Clinical tool used to assess the severity of neonatal encephalopathy.[3]

- Diagnostic tests, such as monitoring of brain function, magnetic resonance imaging (MRI) to detect possible brain damage, blood tests to assess possible imbalances that may be indicative of hypoxia.
- Monitoring of vital signs, such as heart rate, respiration, and oxygenation to detect any warning signs.
- Clinical assessments, evaluating the circumstances of delivery, and a **physical and neurological examination of the newborn**. This examination assesses the newborn's muscle tone, reflexes, and level of awareness to the stimuli that the health care provider makes on it [4].

As can be seen, the detection of this condition is very broad, so any contribution to this field can be of great help. Giving special relevance to clinical evaluations, as mentioned above, it is very important to observe the baby's reactions to stimuli in the first hours of life. According to the Garcia-Alix score scale [5], there are multiple aspects that must be taken into account to determine the presence of HIE in a newborn. However, García-Alíx and other authors agree that alertness is one of the most important [6, 7]. To assess this, it was decided to observe **four characteristics** in each of the files that make up the dataset, and to determine the newborn's response to a stimulus: mouth, and eye opening, frowning, and crying.

This study focuses on the assessment of crying, so it should be noted that not every cry reflects a healthy state of health in the infant. A cry that is considered indicative of a healthy neurological state must meet certain characteristics: it must be sustained, and last at least eight seconds, it must not be abruptly interrupted, and it must exhibit a progression and modulation in pitch that indicates normal sensory and emotional responsiveness.

Advanced audio processing, and machine learning techniques have been used in numerous studies to classify, and interpret these complex sounds that may indicate abnormal conditions in the newborn, and therefore require immediate medical attention [8, 9]. This has improved non-invasive infant monitoring, and provided caregivers with valuable tools for more informed,

and comprehensive care of newborns with these conditions.

Putting these novel advances into practice, one experimental approach to address the challenge of data acquisition in this sector is that proposed in the NeoCam project by the University of Cadiz [10]. This project proposes a strategy of placing automatic cameras in incubators to monitor the vital signs of the newborn, and interpret possible problems or ailments through an analysis of its movements, facial expressions, respiratory rhythm, sleep cycles etc. With this strategy, automatic data acquisition, and even real-time assessment of crying could be achieved, which would be very beneficial for the early **detection of HIE**.

1.2 Assumptions and limitations

This project, like others that focus on **audio classification** or **stimulus assessment by infants**, is based on several key assumptions. Firstly, that the data used is of good quality and, above all, representative of the different types of crying there is. In this particular study, it was observed that the available audios were quite pertinent as they included **recordings of different babies in various health states**. However, this was also a limitation because **all the available data was from the same region**, specifically from the same hospital, which introduced some selection bias. This consequentially resulted into a likelihood of similar crying patterns. This is because if the data used to train the algorithm is more representative of some groups of people than others, the predictions from the model will be systematically worse for unrepresented or under-representative groups leading to inaccuracies.

Another assumption that has been made in this study is in relation to the conditions under which the data have been obtained, i.e. that the **recordings are free of interference and large or loud ambient noises**, which could distort or mask the relevant sounds. In this study, data provided was from a maternity ward of a hospital, a place that does not usually

have much background noise other than that of other babies or health care staff. However, there is an intrinsic limitation as, the model is fed with mostly clean acoustic signals, therefore biasing the predictions when the data acquired from noisier environments is utilized.

It was also assumed that all the datasets worked with in this project were **sufficient to extract enough acoustic features needed by the models**. To increase the value of this assumption, two different types of feature extraction were used, Mel frequency cepstral coefficients (**MFCC**) on the one hand, and linear prediction features (**LPC**) on the other hand. By comparing the results produced by different models with both techniques, a consistent and accurate classification on the target dataset has been achieved.

Another limitation that stands out in this project as in any other that applies Machine Learning techniques, both when classifying audio and other types of data, is the limitation known as *black boxes*[11]. This problem refers to the opacity of the models on how they reach certain conclusions, therefore leading to improper usage or application by staff hence resulting into biased results. This issue generates difficulties with implementation, and mistrust by health personnel and patients.

Finally, one of the main limitations of the project is the **number of samples** (audios) provided. The collection of health data is always an extremely sensitive issue, firstly, it requires consent from the patient and, secondly, there is an ethical duty of researchers to use this data correctly to avoid misuse. Many studies have expressed concerns about **data privacy** [12] and the challenge of **informed consent** [13] affecting the collection and use of clinical data [14]. In this case, the purpose of the study is to assess crying in preterm infants, an aspect that could not be predicted, making it difficult to obtain parental consent for newborn recordings. Without knowing the baby's health condition at the time of birth, many parents did not want to give consent for their baby to be videotaped, which made it difficult to obtain data for this type of study. This therefore has affected the representativeness and generalisability of the results.

1.3 Ethical and Social Impact

This section is directly related to the previous section, as the assumptions we have made about this project may lead to limitations that impact on the technical feasibility of the project, and also have ethical, and social implications.

As mentioned above, generalization is one of the possible issues faced when there is a lack of diversity within the datasets. Numerous studies have reported the challenges presented by **MLP**, **SVM** and **LSTM** models among others, highlighting the challenges associated with generalising results to different settings [15, 16, 17]. It is for this reason that poor generalisation can lead to incorrect diagnoses, according to the article *Key challenges for delivering clinical impact with artificial intelligence* by Sendak, M. P., D'Arcy, J., Ratwani, R. M. [18]. This article proposed improving transparency and diversity in the datasets, and especially the need to implement these technologies in clinical practice to overcome generalisation issues.

Another aspect that has also been discussed above is the lack of trust in the models. The loss or scarcity of public trust in systems that apply Machine Learning for disease detection presents a major ethical challenge. If the systems present problems in disease analysis that can lead to amplified human error, this leads to health professionals showing distrust and resignation to using this type of technology [19]. In the case of newborns presenting with HIE early in life, therapeutic hypothermia initiated within 6 hours of postnatal life significantly reduces mortality and major neurodevelopmental disability [20]. However, if this treatment is applied incorrectly to neonates who do not meet the criteria for this treatment, it can lead to unnecessary, and potentially serious complications.

Of course, we must highlight the importance of privacy, and data security, according to the *General Data Protection Regulation* (GDPR) [21] Sensitive data is that data which reveals information relating to the health or origin of the patient, their genetic information, etc. In

this project, both videos and audios of newborns are considered sensitive data, and therefore require special treatment to ensure privacy, and robust measures to ensure security.

Finally, a problem raised by this type of study in relation to the ethical and social impact is the responsibility or culpability in case of error: who is responsible for an erroneous diagnosis guided by a disease prediction system? Should it be the doctor, who followed these recommendations? Should it be the programmers who implemented the system? Or perhaps the hospital or company that implemented this technology? [22, 23, 24]. In the scenario where a machine learning-based system erroneously predicts that a neonate is suffering from HIE, and the treatment causes harm to the patient, the responsibility for who should be held accountable for this harm can be complex.

1.4 Personal motivation

My personal motivation for this project stems from a combination of several factors. Firstly, since my first steps at the University of Burgos, my interest in the field of health has been intertwined with the value and importance I placed on acquiring knowledge in computer science. Thus, my first steps in the world of Machine Learning began within the ADMIRABLE group[3] of the University of Burgos with my final degree project dedicated to the analysis and classification of movements of people with Parkinson's[4] . This project not only marked my introduction to the field of artificial intelligence, but also reaffirmed my interest in continuing to carry out projects in which technological solutions are used to improve the quality of people's life.

Working on this project allowed me to learn first-hand how technology can improve the

[3] https://admirable-ubu.es/
[4] https://github.com/lnc1002/TFG-Evaluacion-Ejercicios-Rehabilitacion.git

diagnosis and treatment of diseases, and my interest in this field grew until I found this new project offered by the Applied Computational Intelligence (GICAP)[5] group at the University of Burgos. What has motivated me the most in this project is being able to combine my programming skills with my taste for solving problems with a direct impact on the health and well-being of people, especially in the case of such small patients in whom an early diagnosis can help their quality of life from a very early stage.

This project has also allowed me to put into practice much of the knowledge acquired in the master's degree in Biomedical Data Science at the Rovira and Virgili University and expand my skills in biomedical data analysis. In addition, it has been an excellent opportunity to explore areas such as supervised or unsupervised learning and working with sensitive data while ensuring privacy and information security. This challenge has not only brought new knowledge and value to my professional experience but has also reinforced my view on the importance of developing technology in an ethical and responsible manner.

Looking to the future, I am motivated to continue working in the field of health data analytics because advances in this field can be translated into direct improvements for the well-being of the population, including the global population, by making them more accessible and efficient.

[5] https://gicap.ubu.es/main/home.shtml

Chapter 2

Objectives

2.1 General objectives

The general objectives of this project focus on the understanding and analysis of audio containing infant's cries and also to advance in the field of infant's health by providing new knowledge, so that future research and technological developments can improve medical care for newborns.

- To investigate advanced techniques for **neonatal audio analysis**.
- To carry out an exhaustive investigation of the techniques that have been used to date for the classification of crying.
- Conduct a comparison of results on Machine Learning models used for neonatal sound classification.
- Collect and verify labelled audios for training models that accurately represent different sounds emitted by infants.
- Investigate and apply **audio processing methods** and features that improve the

models' ability to distinguish between infant cries and other sounds.

- Investigate and apply different **Python libraries** for the different phases of the project.
- Manually tag the videos provided by the HUBU for training the models.
- Create diagrams and graphs for a better understanding of the results visually.

2.2 Technical objectives

The technical objectives of this project focus on applying the researched methods to the creation of a robust system that can be used in clinical settings.

- Implement a robust **MLP** model for audio analysis in neonatology.
- Implement a robust **SVM** model for audio analysis in neonatology.
- Implement a robust **LSTM** model for audio analysis in neonatology.
- Use a version control system with GitHub.
- Use a Latex typesetting system for the creation of the report and annexes to guarantee a certain quality in the document.
- Use the Overleaf editor to produce the report and annexes.

2.3 Personal objectives

The personal objectives of this project reflect my enthusiasm to grow as a researcher through this project and to put into practice the knowledge acquired so far.

- Try to contribute a novel approach or technique to improve the **detection of diseases in neonates**.
- To put into practice some of the knowledge acquired in my academic career and, above all, the knowledge acquired in this master's degree stage in a real project.
- To strengthen my scientific and technical communication skills.
- To develop and improve my skills in the analysis and handling of audio data.
- Collaborate with professionals in the field of neonatology to validate and improve the developed models.

Chapter 3

State of the art

3.1 Overview

The relevance of this problem at a global level is reflected in the different articles that exist on this subject. There are many articles that present different models for disease detection and improvement [25, 26], in particular the study *Comparing different supervised machine learning algorithms for disease prediction* by Shahadat Uddin, Arif Khan, et al [27], focuses on the evaluation of different machine learning algorithms to detect diseases with health data. In this study, 48 articles applying different algorithms for single disease prediction were reviewed, concluding that by applying these novel techniques, disease predictions become more accurate and efficient compared to traditional statistical methods.

Literature on the applications of deep learning to audio pattern recognition, specifically to the recognition and classification of baby cries, demonstrates its imperative relevance to medical diagnosis. Progress in certain areas leads directly or indirectly to progress in other areas of interest, to show we have highlighted different aspects considered relevant in the reviewed literature:

- **Data acquisition**, *Deep Learning for Infant Cry Recognition* by Yun-Chia Liang et al [28] tackles the problem of obtaining unbalanced data by exhaustive pre-processing and feature selection.
- **Data augmentation**, a technique used to address the scarcity of data has been used by numerous studies, specifically in the article *Infant Crying Classification by Using Genetic Algorithm and Artificial Neural Network* by Azadeh Bashiri and Roghaye Hosseinkhani [29]. This study carried out different comparisons in which by modifying the audio data, adding or removing noise, increasing the playback speed or changing the audio pitch, they managed to demonstrate the effectiveness of this technique to improve the robustness of machine learning models.
- **Data enhancement**, building on the previous technique, another one that becomes very important is sound enhancement. This result can be achieved by removing noise or interference present in the data. In the study *Deep Learning Assisted Neonatal Cry Classification via Support Vector Machine Models* by Severini et al. [30] it is shown how to improve the results obtained from the model by implementing noise removal techniques and simulation of acoustic scenes.
- **Detection and segmentation**, detecting, and segmenting crying in unlabelled audios can be a significant challenge for researchers. In the study *Classification of Infant Cries with Hypothyroidism Using Multilayer Perceptron Neural Network* by Azlee Zabidi et al. [31], manual labelling techniques were used on the data they had available to ensure segmentation accuracy.

3.2 Related Work

Review of the articles on audio classification using deep learning, revealed a multitude of several combinations of techniques that yielded worthwhile results. In this section we present the different techniques that were utilized, and the results they provided. These results are summarised in table 3.2, emphasising how the different features used by each model have been obtained, and with which metrics these models have been evaluated.

Taking into account the articles mentioned in the previous section, this information can be pertinent together with new articles that present the same objective, that is the classification of crying in infants. The studies that have been consulted present the possibility of applying a wide variety of techniques, including supervised learning models such as **MLP** and **SVM**, and also deep neural networks such as CNNs and **LSTMs**. These methodologies have been combined with feature extraction techniques to achieve robust and accurate models.

One compelling study is the article *A Review of Infant Cry Analysis and Classification* by Chunyan Ji et al [32]. This article outlines recent work on the analysis, and classification of infant cry signals. It also highlights, among others, the relevant results provided by LSTM and SVM classifications. The former, has been able to demonstrate its effectiveness in handling sequential datasets due to its ability to capture long-term dependencies in crying data sequences. The latter, together with a combination of **MFCC** feature extraction techniques was able to achieve accuracies above 90% in the classification of different types of cries.

Finally, other articles such as the one by Silvia Orlandi, Carlos Alberto Reyes Garcia, et al [33] also highlight the use of different models to differentiate cries emitted by full-term infants from those of preterm ones. This study mentions the use of a Radial Basis Function (RBF) kernel for the SVM model. This study highlights that this type of *kernel*, compared to others such as the linear kernel, and the polynomial kernel, offers better results achieving 95.86% accuracy for the identification of cries related to asphyxia.

Title, Year, Authors	Database	Models	Features	Metrics
Infant Crying Classification by Using Genetic Algorithm and Artificial Neural Network, 2018, Azadeh Bashiri, Roghaye Hosseinkhan, et al [29]	Baby Chillanto database, 2268 cries	Genetic Algorithm (**GA**), Artificial Neural Network (**ANN**)	304 **MFCC**, 50 **LPC**	Accuracy, Sensitivity, Specificity
An Efficient Classification of Neonates Cry Using Extreme Gradient Boosting-Assisted Grouped-Support-Vector Network, 2020, Chuan-Yu Chang, Sweta Bhattacharya, et al [34]	Collected from hospitals, 1000 cries	Extreme Gradient Boosting, Grouped-Support-Vector Network (**SVMs**)	12 Acustic **MFCC** features	Accuracy, Precision, Recall
A review of infant cry analysis and classification, 2020, Chunyan Ji, Thosini Bamunu Mudiyanselage, Yutong Gao and Yi Pan [32]	Different databases	KNN, **SVM**, GMM, and neural network architectures such as CNN and RNN	**MFCC**, LPCCs, and LFCCs	Accuracy, Precision, Recall, F1-score
Deep Learning Assisted Neonatal Cry Classification via Support Vector Machine Models, 2021, Ashwini K, P.M. Durai Raj Vincent et al [30]	300 audio records	Support Vector Machine (**SVM with RBF kernel**)	CNN from spectrogram images	Specificity, Sensitivity, Precision, Accuracy, F1 Score, **ROC** and **AUC**
Deep Learning for Infant Cry Recognition, 2022, Yun-Chia Liang, Iven Wijaya, Ming-Tao Yang et al [28]	1607 cries from Far Eastern Memorial Hospital	Convolutional Neural Network (CNN), Long Short-Term Memory (**LSTM**), ANN	MFCC features	Accuracy, Precision, Recall
Classification of Infant Cries with Hypothyroidism Using Multilayer Perceptron Neural Network, 2009, Azlee Zabidi, Wahidah Mansor, Lee Yoot Khuan, et al [31]	25 hypothyroid, 20 normal	Multilayer Perceptron (**MLP**)	30 MFCC features	Accuracy, Mean Square Error, Sensitivity, Specificity
Automatic Infant Cry Pattern Classification for a Multiclass Problem, 2016, N.S.A. Wahid, P. Saad, M. Hariharan [35]	Baby Chillanto database, 1918 cries	**MLP**, Radial Basis Function Network (RBFN)	**MFCC** and LPCC features	Accuracy, Kappa value
Application of Pattern Recognition Techniques to the Classification of Full-Term and Preterm Infant Cry, 2016, Silvia Orlandi, Carlos Alberto Reyes Garcia, et al [33]	3000 crying units	**MLP**, **SVM with RBF kernel**, Random Forest	BioVoice software to extract 22 acoustic parameters	Accuracy, F-measure, **ROC** and **AUC**

Table 3.2: Overview of studies that classify baby crying

Chapter 4

Scope

4.1 Hypothesis

The purpose of this section is to set out in detail, and concisely both the scope of this project, and the main objectives that have been pursued. It seeks to collect new information as from the time of doing this study, there weren't any others that used the techniques mentioned in the previous section to classified audios and advancing the field of HIE detection among neonates. The results from this study will be made easier to extrapolate, and be applied to other projects that are currently developing different solutions within this same line of research. This holistic approach allows for a more comprehensive understanding of the possible practical application and implementation of the project in other settings for better HIE diagnosis.

This project is divided into two phases, the first phase tests different models (**MLP**, **SVM**, **LSTM**) with a public dataset optimising the parameters until satisfactory results are obtained. Then in the second phase, the models are adjusted, and trained with real data (HUBU dataset) from this project.

4.2 Main objective

Apart from the objectives that have been mentioned in the Objectives (Section 2), it is essential to clarify, and detail in depth, the specific purpose of this project, and how its results can be applied in real contexts.

> In this context, the specific objective of the project is to develop a dataset with arrays that detect crying in every second of the recorded audio. This way of presenting the results is crucial, as it provides an essential tool that can later be integrated with other datasets generated by parallel projects.

These projects analyse different aspects of how the neonate reacts to a stimulus, focusing on visual elements, and combined with the results from this study ultimately generates a multidisciplinary approach that increases the understanding of the responses that the neonate may make for improved HIE detection.

4.3 Interrelation with other projects

In the scheme presented in the image 4.1 you can see the integration of the projects that are currently being developed within the Applied High Performance Computing Research Group (GICAP)[1] of the University of Burgos, for the detection of HIE. Within this multidisciplinary group, each research line covers specific areas related to the neonatal response to different stimuli. These areas include, the identification of spasms, and arm or leg movements, as well as the detection of whether the mouth or eyes remain open or closed, and the classification of

[1] https://gicap.ubu.es/main/home.shtml

facial expressions such as frowning, and other visual cues.

The findings from these responses are intended to provide an understanding, and classification of the infant's state of alertness. This understanding is crucial to critically intervene in a timely manner if abnormalities in the newborn's behaviour are detected.

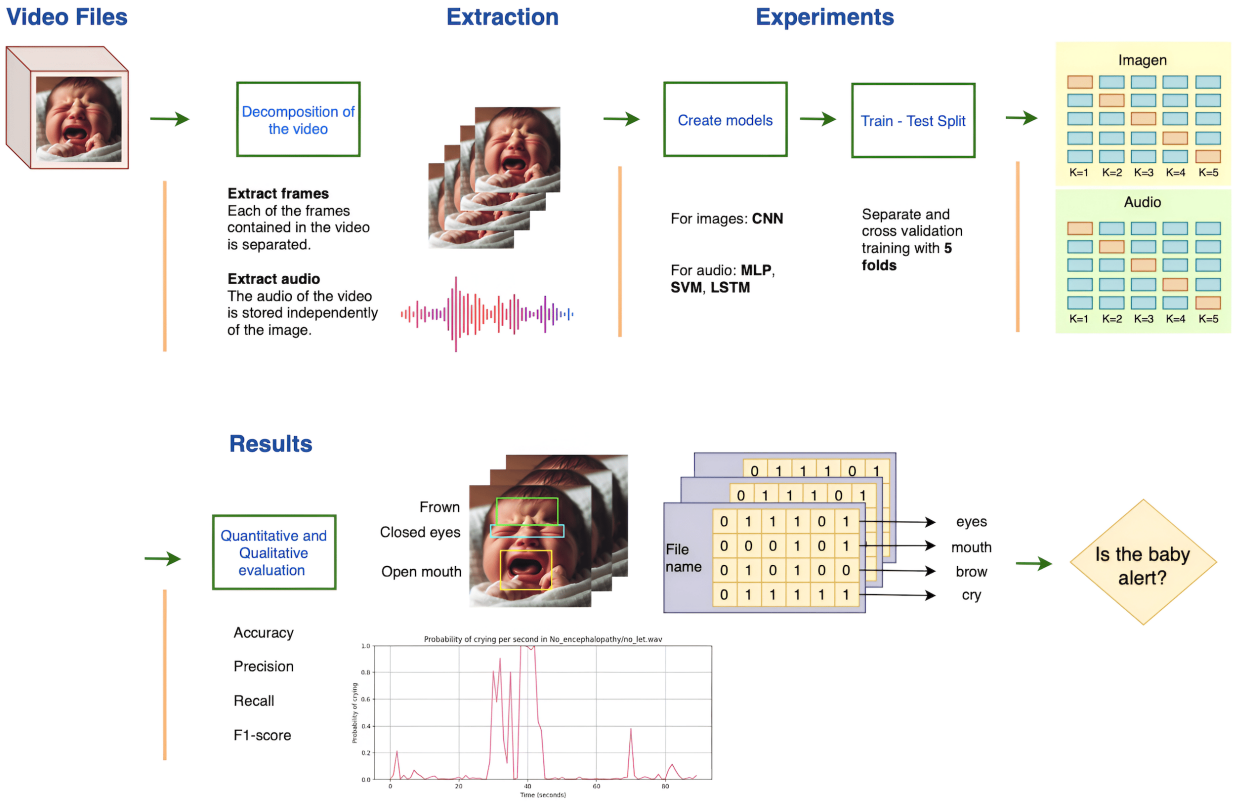

Figure 4.1: Pipeline of the project interrelated with other projects

4.4 First steps

The first steps consisted of a thorough evaluation to determine whether the CNN techniques used in the classification of images of neonates' facial expressions for HIE detection in the work by mentioned in the previous section, could be utilized for sound classification in this study for the same purpose. This preliminary approach is essential to understand the overall development of the project and to establish a correct methodological approach that can be transferred and applied to the field of audio classification.

Initially, the possibility of continuing with the line of work (Convolutional Neural Networks (CNN) to classify the images of neonates' facial expressions) by GICAP group was raised. In this study, with audios of neonates, the acoustic signal was transformed into a waveform representation, which allowed visualising the intensity of the sound over time. Once this representation was created, it was hypothesised that the peaks with the highest amplitude corresponded to the moments when the baby was crying.

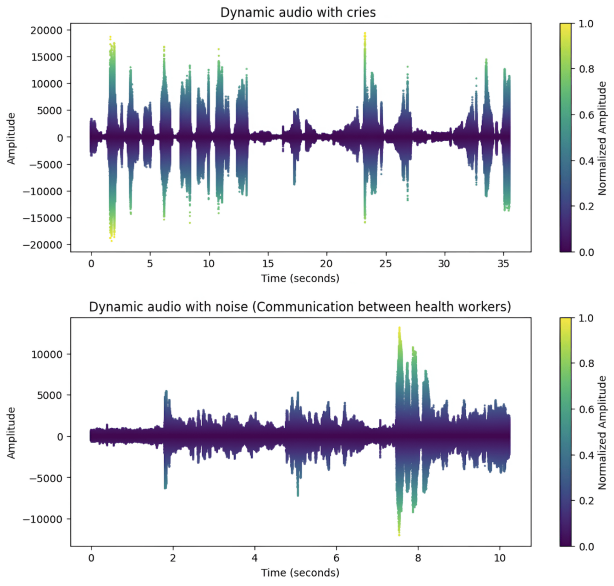

Figure 4.2: Waveform of different audio files

The above technique, the Python **PyDub** library has been used, which is responsible for loading and processing the stored audio files. **Parallel to the audio representation process, a process of analysis of the different sounds in the audios in each available file was followed**. These audios contain a variety of sounds, including, the main sounds expected to be heard, which are the cries of the neonates, but also instructions, and communications among healthcare personnel, or noises from different medical devices can also be heard. This means that the times when the waveforms reach a higher amplitude do not always correspond to a baby's cry. This issue can be clearly seen in the waveform comparisons in the image 4.2. In the first waveform, the baby's cry predominates, making the image a viable, and a good alternative to select the moments when the amplitude peaks in order to identify the cry. However, the second waveform represents an audio in which there is no baby crying but includes other types of sounds. In this second scenario, the limitations of using waveforms for baby cries' detection are evident, as they only identify different types of sound according to their intensity.

This gap presents the need to develop a more sophisticated approach for audio classification. The main goal of this project is to effectively classify neonatal cries, regardless of the intensity with which the neonate cries, or the variety of sound sources with which it is mixed. This is crucial because the audios researchers work with are from babies who are only a few hours old or who have delicate health conditions. For this reason, on many occasions, acoustic reactions to stimuli are expected which may be minimal, and just a slight whimper could be decisive in determining the baby's state of health.

4.5 Second steps

After discarding the first line of action, and addressing the limitations that were found, a review of scientific articles [29, 34, 32, 30, 28, 31, 35, 33] focused on finding solutions to problems similar to the one faced by this project. With this review, a new approach to the project was generated that focused on the possibility of implementing dissimilar machine learning models. In this case, a group of widely recognised models for audio classification were selected: Multi-Layer Perceptron (**MLP**), Support Vector Machines (**SVM**), and Short-Term Memory Neural Networks (**LSTM**). Different Python libraries have been used for the development of these models, in particular the **Scikit-learn** library to integrate the MLP, and SVM models, and **TensorFlow** plus **Keras** to develop the LSTM model.

Once the implementation of machine learning models was chosen, the first drawback arose: **the videos available for audio classification (HUBU videos) were not labelled**, which created a problem in the supervised training of the selected models. These models depend on input examples with real labels so that later, during training, they learn from the input data, and finally make the predictions in the test phase.

To resolve this drawback, a **public dataset** was used to test the selected models. This dataset is organised in a directory structure as shown in the image 5.2. Each directory contains 108 short audio files classified according to the type of sound each one stores.

With this labelled dataset, feature extraction was carried out using various techniques, which are discussed in more detail below. Several experiments have been carried out, some with and some without five-fold cross-validation to assess the generalisation of the models. This was checked using test data, and performance metrics such as **accuracy**, **F1-score**, **precision**, and **recall** that resulted in satisfactory performance for each of the models, that have been used with the real data of this project.

Chapter 5

Project development

5.1 Design, Pipeline and Overview

This section will describe the general process of the project as shown in the image 5.1. This image shows in broad outline the stages that have been developed in this project, focusing only on the procedures followed with the data provided by the University Hospital of Burgos (HUBU).

> The dataset provided by HUBU has **63 video files** organised in different directories: Unclassified (58 files), Mild_encephalopathy (1 file), Severe_encephalopathy (1 file), Moderate_encephalopathy (1 file), and No_encephalopathy (2 files).

The first stage includes the decomposition of the HUBU videos, where audio was extracted from each file, and stored under the same name, but with a different extension. Then, each of the audio files was manually labelled. For the creation of these labels, a vector has been generated with a length equal to the number of seconds that each audio file. The vector is composed of **0's** and **1's** in which the former indicates **that the baby is crying** in that

second of the audio, and the latter **that the baby is not crying**. This is regardless of the sound that is heard when the baby is not crying, as a binary classification is generated depending on whether crying is detected or not.

Once the data had been labelled, the next step was to extract features from the files belonging to the *Unclassified* directory, and this was done using two techniques that is **MFCC**, and **LPC**, and thereafter a comparison of the results was carried out. Once the features were extracted, **MLP**, **SVM**, and **LSTM** models were created both "with" (**5-fold**) and "without" **cross-validation** to compare the results obtained.

Taking into account that the class *Baby_cry* has fewer samples than the class *Baby_not_cry*, **the data has been increased in the minority class (undersampling), and decreasing the data in the majority class (oversampling)**, were performed in order to balance the available data. Once all these steps have been taken, all that remains is to train the models with different combinations of parameters until the best combination is found for each model.

Finally, to validate the results obtained, **two validation strategies** were followed. The first was to evaluate the results using performance metrics. It should be noted that since these are binary classifications, the Receiver Operating Characteristic (**ROC**) curve, and the Area Under the Curve (**AUC**) have been calculated in addition to the common metrics, such as **accuracy**, **recall**, **precision**, and **f1-score**. The second strategy was to evaluate the models using the remaining five audios that were not previously used. Thereafter, a graph of the probability that the baby is crying in each second of the audio was developed.

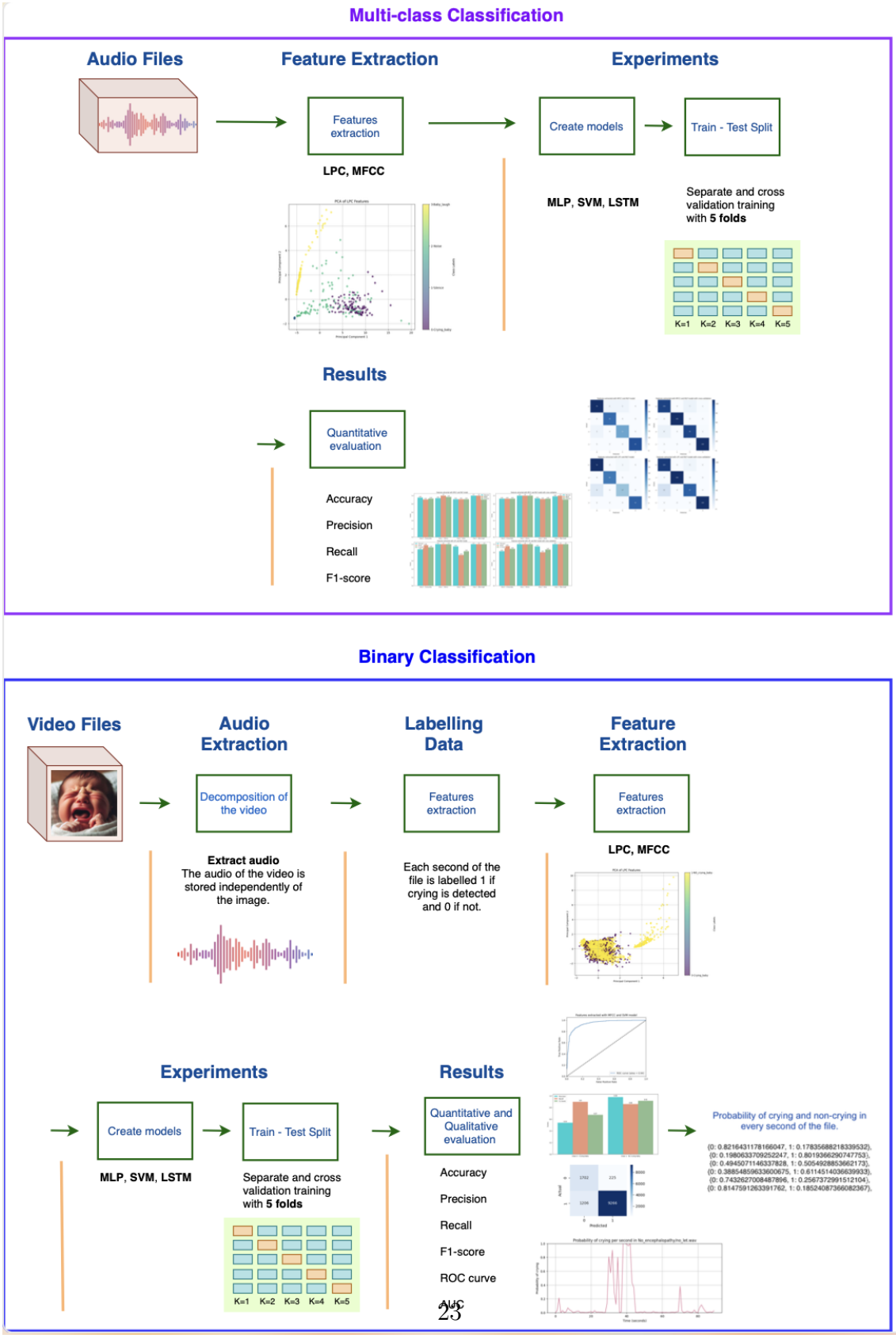

Figure 5.1: General diagram of the project

5.2 Datasets

This project worked with two datasets, first, a public one for initial testing, and the second, and most *important*, a dataset provided by the University Hospital of Burgos.

> The public dataset was only used to assess the functionality and / or viability of the models in classifying infant crying. The dataset provided by HUBU is fundamental as it contains raw data with which the models will be trained to accurately, and precisely identify newborns' cries.

The first dataset comes from the public repository Baby Cry Detection[1] and its structure corresponds to the one shown in the image 5.2. This dataset has the following four directories, each containing one hundred and eight (108) files with extensions (.ogg, and .wav):

- **Noise**, this directory can be considered the most diverse, and it contains files of different noises that any person can experience on a daily basis, for example, an animal bark, vehicle noises from the road, sounds such as birds singing, and those generated by church gongs, etc.
- **Silence**, this directory contains audios that have no significant variation in amplitude over time. They are flat audios (such as white noise) and do not contain absolute silence but contain a series of sounds or noises that are not characterised by anything in particular.
- **Baby_laugh**, this directory has files that contain the laughter of different babies. Unlike crying, the laughter tends to be softer, and more musical without a great intensity or presence of peaks in volume.
- **Crying_baby**, this directory has files containing the cries of different babies. Unlike

[1] https://github.com/giulbia/baby_cry_detection.git

the previous directory, a baby's cry has a higher pitch, and is more variable in frequency compared to a laugh, and this distinction makes the machine learning models effectively differentiate them.

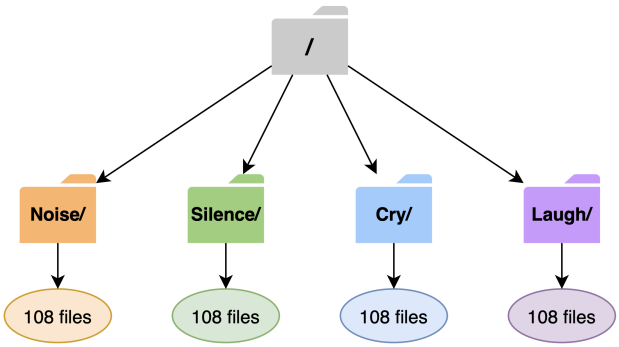

Figure 5.2: Directory structure of the public dataset

As can be seen in the previous files, there is a notable mixture of sounds, which can be essential for a precise identification of crying. However, the models will inevitably have some inefficacies in identifying cries since the audios from the public dataset are so ideal, yet the audios they are eventually trained with are more typical or representative of the environment (i.e. health setting).

The second dataset contains a total of **sixty-three (63) files** provided by the maternity ward of the **University Hospital of Burgos**. These are video files of newborns babies' responses, and reactions to differences stimulus presented by health workers to detect HIE. All these babies were born in 2023, and were less than twenty-four (24) hours old, at the time the videos were made. Some of the babies were premature, and had underlying health conditions and / or complications during birth. For this reason, the cries in this dataset are diverse, and this is key in training the models.

The videos vary in length, ranging from videos as short as thirty seconds and up to sixteen-minutes. Them were captured using two different smartphones, each one with different image properties. Firstly, one set of videos (stored as .mov files) was obtained at a frame rate of 29.89

frames per second (FPS), with each frame having a resolution of 1920 x 1080 pixels. Secondly, another set of videos (stored as .mp4 files) was recorded using a different smartphone at a frame rate of 30 FPS and a resolution of 848 x 480 pixels. However, the audio characteristics remained consistent in both sets of videos. Both sets were recorded using two audio channels and a sampling rate of 44.1 kHz.

The videos from the previous paragraph demonstrate how the HUBU medical staff perform a thorough examination of the newborn or a portion of it. The complete examination consists of numerous stimuli to which the baby must make a response, such as checking whether it's gaze follows a presented object or whether it generates the involuntary movements that are expected in response to a given stimulus. However, in this project, the videos in which a small pinch is performed on the baby to elicit a cry were the main focus. Also, the cry's characteristics such as duration, pitch, continuity or progressiveness, and whether the crying stopped abruptly were also key considerations.

It is important to note that although the videos contain audio with some details, and indications from the doctor about the newborns' conditions, the audio has only been used to analyse crying in this project. In addition, it is worth mentioning that in the absence of prior information related to the videos, it was so crucial to carry out labelling as part of the pre-processing of the study data.

The directory system that was created with the videos is shown in the image 5.3 and has the following five directories:

- **Unclassified**, this directory contains a total of 58 videos with different lengths and video formats. In the videos, medical personnel can be observed exposing the neonates to different stimuli. Sometimes the video contains a single stimulus, thus the shorter files, and on other occasions several stimuli which generates longer files.
- **Severe_encephalopathy**, this directory contains a single video file of a newborn with

severe signs of encephalopathy, which means that this baby does not cry at all when exposed to various stimuli. The video lasted 16 minutes and 54 seconds.

- **Moderate_encephalopathy**, this directory contains a single file with a duration of 90.7 seconds. In it, the same stimulus with varying intensity, is repeated up to three times on the neonate, to test its response to the pain.
- **Mild_encephalopathy**, this directory contains a single video file of a newborn baby exposed to different stimuli. Due to its long duration, audio was extracted and divided into a series of clips that represent each of the stimulations given to the newborn. This was to evaluate more precisely whether the models correctly classified the moments in which the newborn cried.
- **No_encephalopathy**, this directory contains two video files of 89.4 and 55.08 seconds. These files contain the stimulations performed on two different babies. Due to their healthy state, their response to the presented stimuli is that of an intense cry.

As can be seen in the file system above, most of the videos are unclassified in terms of the degree of the babys' encephalopathy. This was not an issue encountered in this project, since the general objective was to be able to detect whether the baby had HIE or not by means of data analysis. Nevertheless, this isn't the main objective of this project which is, to precisely detect whether a baby cries or not at any given time, when exposed to stimuli in a health setting. Additionally, it should be noted that the absence of crying is not a conclusive indicator of a positive HIE diagnosis. An example is some of the babies tested (exposed to stimuli) for HIE in the HUBU videos who had been intubated, could not cry, and this did not necessary mean they had HIE.

All videos used in this study were acquired with the written consent of the newborns' parents, who voluntarily provided their babies' data, to contribute to the advancement of the study of HIE. However, due to privacy and data protection laws, this project will not include the original videos, audios, or images that would allow identification of the babies, nor will names or any other data that could facilitate identification of the study participants or their family members be released.

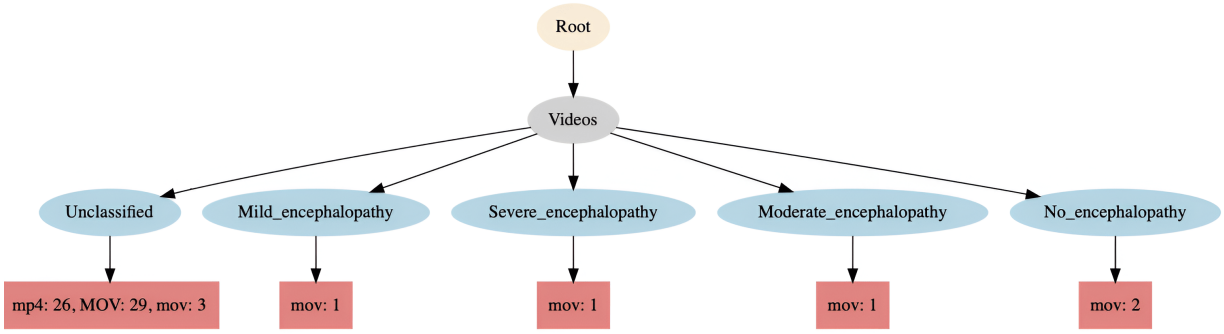

Figure 5.3: Directory structure of the HUBU dataset

5.3 Data Preparation

The first step to work with audio in this project was to separate the video from the audio. This process is available on GitHub in the notebook /src/1)First_steps/1_Notebook-1_Audio_Extraction.ipynb[2] where the audio extraction from the video can be appreciated using the **PyDub** Python library. Next, a directory system has been created, as shown in the image 5.4, to store the audio files. In this way, a quick association between the extracted audio, and the original videos is achieved.

All the extracted audio files have the extension **.wav** and the original name of the different files has been preserved. This has been done with the aim of subsequently creating a dataset that includes the different responses captured in relation to stimuli performed, i.e. for each file a list of the time frame in which the eyes and mouth remain open, the frown, and the baby's cry.

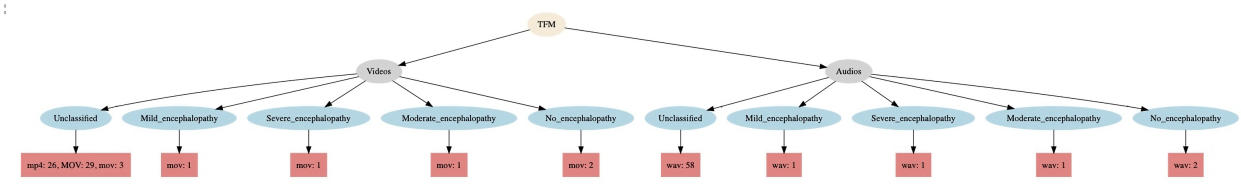

Figure 5.4: Directory structure together with the extraction of audios from the HUBU dataset

The way in which the data has been labelled also needs to be emphasised. For this purpose, a code has been developed that automates the process based on the duration of the audio in seconds. This code generates an array containing **binary values**, where a value of **0 indicates that the baby is crying** and a value of **1 indicates that the baby is not crying**. This vector is created in such a way that it has as many values as there are seconds in the audio, providing an accurate and granular representation of the baby's behaviour over

[2] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/06473b645fd3b11fd7c81186fe494e7f3e4469ed/src/1)%20First_steps/1_Notebook-1_Audio_Extraction.ipynb

time.

This process is simple to perform, in fact, it can be done manually by directly assigning 0s and 1s according to the baby's crying. In any case, the code has been very useful due to the fact that sometimes there are very long audios in which entering values by hand is tedious, especially when the time intervals in which the crying is appreciated are minimal. Despite its great usefulness, due to its simplicity and lack of direct relevance to this project, it has been decided not to show this code and to make only the generated files available to the user.

In the notebook /src/1)First_steps/2_Notebook-2_Audio_Basic_Operations.ipynb[3] the tags that have been created are exposed. As mentioned above, the data is structured in directories, with the *Unclassified* directory having the most data available. For this reason the labels are stored in several text files: *Unclassified_data.csv* (contains the labels for the audio files in the *Unclassified* directory), *Test_data.csv* (contains the labels for the audio files in the other directories, those that classify the baby's cry according to the type of HIE) *Mild_encephalopathy_clips.csv* (contains the labels for the audio clips in the *Mild_Encephalopathy* directory).

The audio file available in the *Mild_Encephalopathy* directory is composed of several time instants of baby crying, but the duration of this file is rather long. For this reason, so that later in the final qualitative analysis it is possible to clearly observe the instants of time in which the baby is crying, it has been decided to create **nine clips** with the different stimulations that are performed on the baby. These clips consist of dividing the audio according to the instants of time in which the baby begins and ends with a stimulation.

[3] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/1)%20First_steps/2_Notebook-2_Basic_Operations.ipynb

It is very important to note that this type of labelling is very simple and the results may appear to be inaccurate in some cases. Unlike the models that have been trained and evaluated in this project, this labelling does not give a probability that the baby is crying but a single value, regardless of the amplitude of the sound, e.g. at the beginning or at the end of crying.

5.4 Feature Extraction

After reviewing the existing literature on which techniques should be used to classify infant crying, this section will present the techniques applied for the extraction of characteristics. The first technique, **MFCC**, has been referenced in many articles, so it was essential to implement it. The second technique, **LPC**, has been referenced in some articles and in this study it has been tested in order to create a comparison with the previous technique and to check how much better results it can offer.

> The public dataset has a total of four classes (Crying_baby, Silence, Noise, Baby_laugh) and the dataset provided by the HUBU has two classes (Baby_cry, Baby_not_cry).

5.4.1 Mel Frequency Cepstral Coefficients (MFCC)

Feature extraction using Mel Frequency cepstral coefficients (**MFCC**) is a complex process by which an audio signal is transformed from its original time domain to a representation that mimics the human perception of sound. The resulting coefficients represent the shape of the spectral envelope of the sound in the domain of the logarithm of the Mel frequencies [36].

These are a good choice for audio analysis and sound classification because they provide a representation that mimics the way humans perceive sound.

This feature extraction process can be broken down into several key steps as shown in the image 5.5.

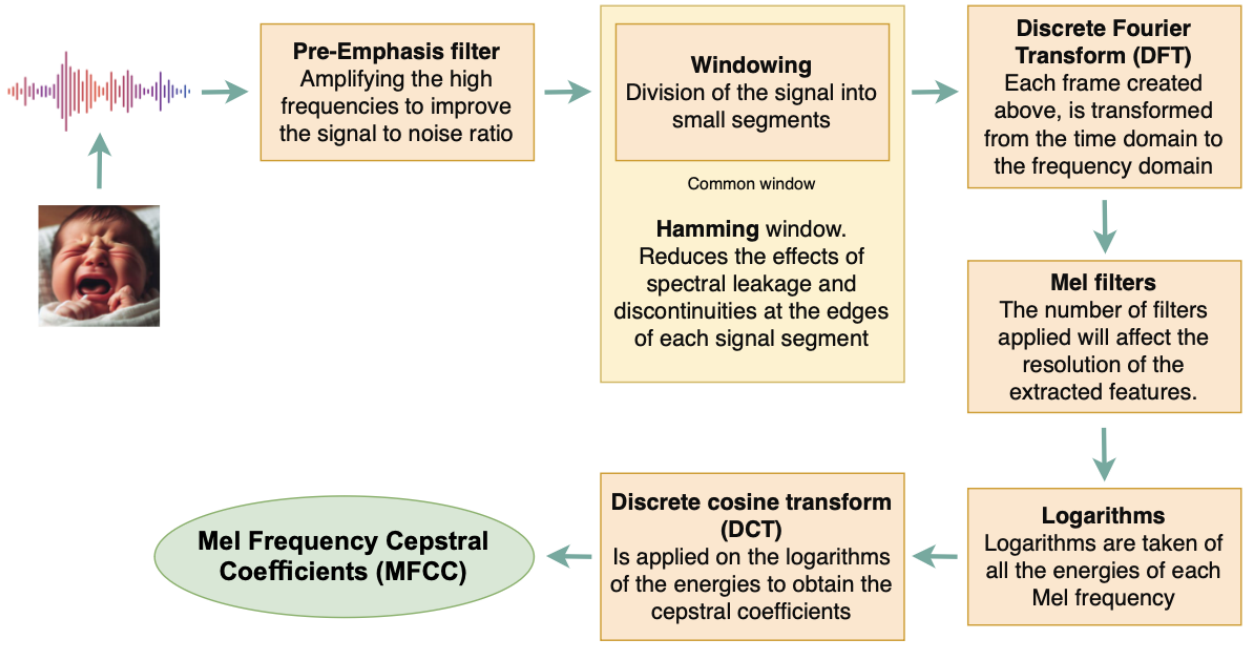

Figure 5.5: Diagram of the MFCC Feature Extraction Process

These values that have been obtained are the MFCC coefficients that were sought to capture the most important characteristics of the audio signals. The objective of these coefficients is to identify the relevant content by ignoring information of little value, such as background noise, which does not contribute anything to the recognition process, on the contrary, it impoverishes it.

MFCC implementation

The Python library **librosa** has been used to implement a feature extraction based on MFCC. This library is widely used in audio processing tasks due to its efficiency and ease of use. Its implementation is available in the notebook 3)Final_steps/7_Notebook-7_Extract_Binary_Features.ipynb[4].

The first step has been to load the audio files with a specific sample rate, in this case a **sample rate of 44100Hz** has been set, which means that the audio is re-sampled at 44100 Hz during the loading. This sampling rate is that of each file in the dataset provided by the HUBU. However, for the sake of clarity, it has been decided to specify it because the librosa library detects whether or not the sampling rate of the file needs to change in order to start re-sampling, so unnecessary processing is not generated.

The second step has been to **divide the audio into one-second segments**, where each of these segments have a length equal to the previously defined sampling rate. Each of these segments is checked for sufficient length, at least 2048 samples. This step has been carried out in order to apply the **Hamming** window correctly, which the **librosa** library uses internally, and subsequently the Discrete Fourier Transform (DFT). After this check, the process is continued by extracting, in this case, **thirteen MFCC coefficients** for each segment. The choice of this number of coefficients is given by the objective pursued by the project. This value provides a balance between information and computational efficiency, being sufficient for speech processing tasks.

Finally, the result obtained is an average of the coefficients over time for each segment, so that each second of audio becomes an independent instance with fixed-length characteristics. **In this way, the models are subsequently provided with consistent data for training**.

[4] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/3)%20Final_steps/7_Notebook-7_Extract_Binary_Features.ipynb

5.4.2 Linear Predictive Coding (LPC)

Linear Predictive Coding (**LPC**) feature extraction works in a very similar way to MFCC. Although both techniques divide the signal into frames and apply a window to reduce edge effects, they differ significantly in how they process those frames to extract features. LPC models the signal as a linear combination of its previous samples and uses autocorrelation to estimate the predictive filter coefficients [37].

> This type of feature extraction attempts to model the human production of sound rather than transmitting an estimation of the sound wave.

This feature extraction process can be divided into several key steps, as shown in the image 5.6.

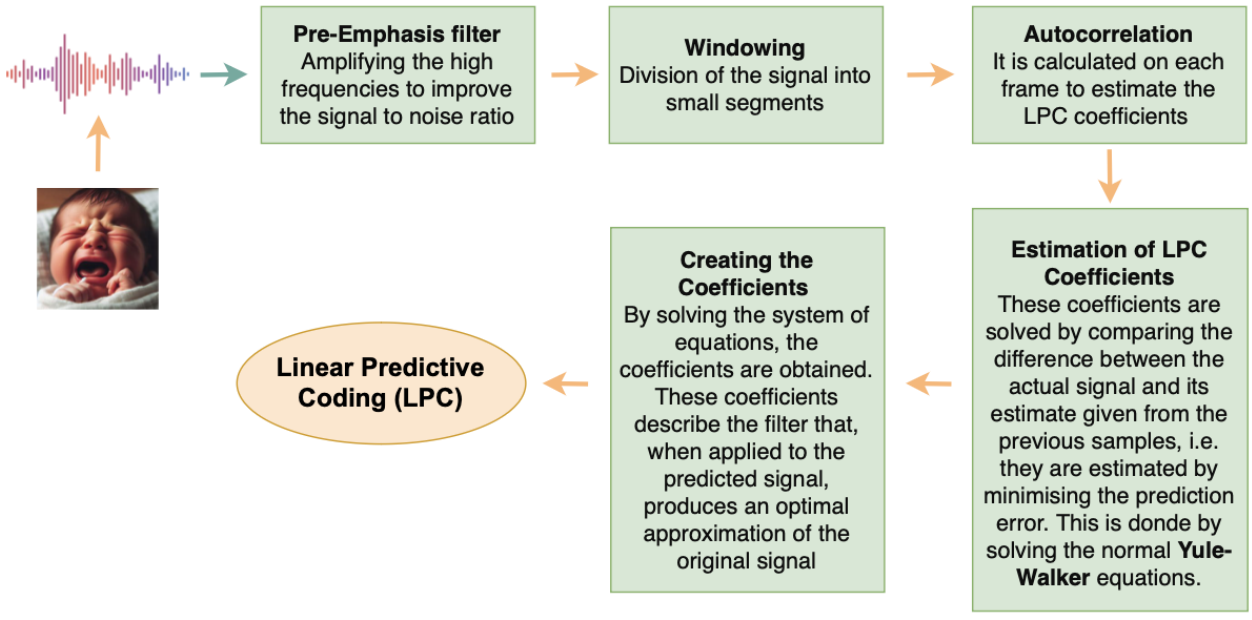

Figure 5.6: Diagram of the LPC Feature Extraction Process

LPC implementation

As in the previous case, the Python library **librosa** is still used for feature extraction, in this case based on LPC. The implementation is available in the notebook 3)Final_steps/7_Notebook-7_Extract_Binary_Features.ipynb[5].

The first step has been to load the audio files with a **sampling rate** equal to **44100Hz**. Then, the **pre-emphasis** filter has been applied to improve the signal to noise ratio. This filter follows the formula where *alpha* is greater than 0 and less than 1.0 [38]. Typical values for *alpha* are between 0.95 and 0.97, on this occasion a value of **alpha=0.97** has been selected with the intention of amplifying the high frequencies to not introduce excessive distortion into the signal.

Afterwards, the audio has been segmented into **one-second fragments** and a **Hamming window** was applied to each segment. After the windowing, the next step was to apply the autocorrelation technique. In this case, the most relevant part for the LPC coefficients has been selected, i.e. the part related to the centre of the forward autocorrelation has been extracted. This step has been intended to capture the temporal structure of the signal, making it possible to determine the current samples from the previous samples. Since the LPC coefficients model the signal according to the linear combination of past samples, this is a fundamental step.

Finally, the Toeplitz equation is solved using the Levinson-Durbin algorithm to obtain the LPC coefficients. These coefficients are stored in a list of vectors in which the length of each vector is equal to the number of coefficients previously defined, in this case the **number of coefficients** has been defined equal to **10**.

[5] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/3)%20Final_steps/7_Notebook-7_Extract_Binary_Features.ipynb

5.5 Evaluated Features

This section presents the main methods to achieve dimensionality reduction, and to visualise the features extracted from the different available data. As explained in the previous section, the number of classes differs between the different data available, that is the multi-class dataset, and the binary dataset.

When feature extraction techniques are applied on the data, such as those used in this project, that is **MFCC**, and **LPC**, each audio sample is transformed into a set of features such as cries, laughs, and silences that detail essential information about the sound. The number of features vary, and depend on the extraction parameters such as the number of coefficients that are defined prior to extraction.

In the case of the **MFCC** features extracted from the HUBU data, a total of 1239 features, and labels have been obtained per audio sample, which means that each sample in the dataset has been represented as a point in a space of 1239 dimensions. Each of these dimensions contains a specific aspect of the audio signal, such as the frequency of the formats, the energy in different frequency bands, etc. Having numerous features has a positive effect on the result, as it allows a more detailed representation of the sound, although, of course at the cost of a higher complexity of the results.

For this reason, it is essential to apply dimensionality reduction techniques to be able to observe the results summarised in the image 5.7. In this case, three techniques have been utilized i.e., Principal Component Analysis (**PCA**), t-Distributed Stochastic Neighbor Embedding (**t-SNE**), and Uniform Manifold Approximation and Projection (**UMAP**). All of them are available in the notebooks 2)Second_steps/3_Notebook-3_Extract_Features.ipynb[6], and

[6] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/2)%20Second_steps/3_Notebook-3_Extract_Features.ipynb

3)Final_steps/7_Notebook-7_Extract_Binary_Features.ipynb[7], and are described in the next paragraphs.

Principal component analysis (PCA) is a technique that consists of projecting observations from a *p-dimensional* space where p is the number of variables to a *k-dimensional* space in which k must be strictly less than p [39]. This transformation models the set of possibly correlated variables into a set of uncorrelated variables called principal components [40]. **The main objective of this technique is to keep as much information as possible while reducing the complexity of the data sets.**

In this project, the **Scikit-learn** Python library has been used to provide an efficient implementation of PCA. Firstly, PCA was used to expose the data in two main components, thus creating a two-dimensional plane for visualisation. Then the data transformation is performed, and the scatter plot is drawn. This action has been performed with both MFCC, and LPC feature extraction methods. In the case of the features extracted with LPC some clustering of the data expressing the baby's non-crying can be observed, however, with MFCC the classes do not form clearly separated groups.

> These results are to be expected as for newborns' cries, the cry is intermittent or discontinuous that is with varying amplitudes over time. This is on account of the intervals at which the baby inhales or exhales. Since amplitude at this point is low, the model might identify this timeframe as "non-crying". This can result in inaccuracies as the model could interpret these instances as non-cries. This is why, although at first glance these results may not seem very encouraging, the truth is that they are the 'expected' ones.

[7] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/3)%20Final_steps/7_Notebook-7_Extract_Binary_Features.ipynb

Another technique **t-Distributed Stochastic Neighbor Embedding (t-SNE)** was used to compare the results. This technique consists of two main stages. The first stage focuses on the calculation of probabilities representing the similarities between pairs of points in a high-dimensional space. These probabilities indicate that closer pairs of points have a high probability of being selected together, as opposed to points that are far apart, which have a near-zero probability of being selected together [41].

In the case of the features extracted with LPC some clustering of the data expressing the baby's non-crying can be observed, however, with MFCC the classes do not form clearly separated groups.

The second stage focuses on mapping the obtained points to a lower-dimensional space, commonly two- or three-dimensional spaces. The aim is to make the similarities between the data in the lower dimension as representative as possible to those in the original space. This is achieved by minimising the Kullback-Leibler divergence between the probability distributions with respect to the locations of the points on the map. Once more, the classes are clustered, and do not end up forming clearly distinct groups. For this reason, a final visualisation has been created to picture the results obtained in the previous feature distributions.

Uniform Manifold Approximation and Projection (UMAP) is a Riemannian variant of t-SNE but, unlike t-SNE, it accomplishes a more effective balance by preserving local and global structures. This technique is mainly based on algebraic topology theory, which creates a projection of the high-dimensional data while preserving its topological structure.

In this project we have used the **umap-learn** library that depends on Python's **Scikit-learn** to be able to observe the characteristics of the data by dimensionality reduction. The results obtained are similar to the ones obtained by the other previous techniques, and therefore, it is conclusive that these techniques are effective for a visual representation, but they are not enough to discriminate among classes effectively.

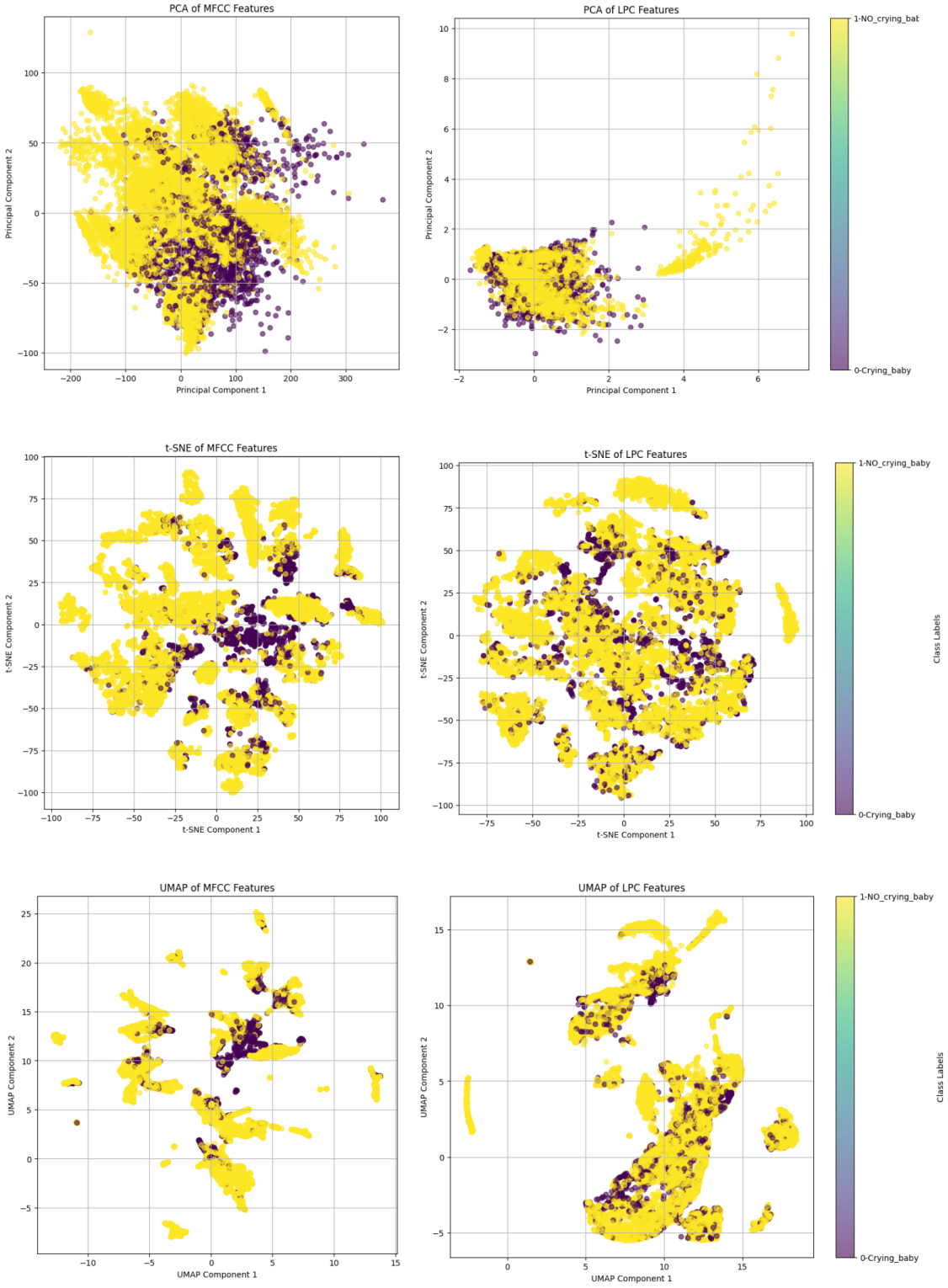

Figure 5.7: Visualisation of the extracted features with different metrics.

5.6 Model Architectures and Implementation Details

In the development of Machine Learning and Deep Learning models, it is as important to select the most appropriate models according to the type of classification being performed, as the way in which these models will be built and the techniques that will be applied to ensure their robustness and effectiveness. In this study, two fundamental aspects have been implemented: cross-validation and oversampling and undersampling techniques.

Firstly, cross-validation is a resampling method based on the idea of dividing the dataset into k partitions, where in each of these partitions a test set is used only once, with the rest of the data being used for training. This process is repeated, rotating the allocation of the test data, so that at the end of the process all the data is used in both training and testing [42].

> The models have been trained with a total of 12399 MFCC and LPC features of which only 1927 (15.54%) belong to the *Baby_cry* class and 10472 (84.46%) belong to the *Baby_not_cry* class.

Most of the audios contain crying data only at the moment when the nociceptive stimulus[8] is performed on the baby, which causes most of the audio file to contain other non-crying sounds creating an unbalanced dataset in which the crying class is significantly under-represented.

This problem is tackled in two different ways, the first is oversampling, extending the minority classes and the second is undersampling which reduces the number of samples in the majority class. In this project both techniques have been developed using **SMOTETomek** which is a hybrid technique combining **SMOTE** (Synthetic Minority Over-sampling Technique) for oversampling and **Tomek Links** for undersampling.

[8] A nociceptive stimulus is a stimulus that activates pain receptors, usually due to tissue damage.

5.6.1 Multilayer Perceptron (MLP)

The Multilayer Perceptron (**MLP**) is a fundamental structure in the field of Artificial Neural Networks (ANN) due to the variety of solutions it presents for use in the field of machine learning. Its main characteristic is that it is capable of solving problems that are not linearly separable, which is also its main limitation [43]. This model consists of at least one input layer, one hidden layer and one output layer, as shown in the image 5.8. In this same image it can also be seen how each layer is connected to the next by neurons.

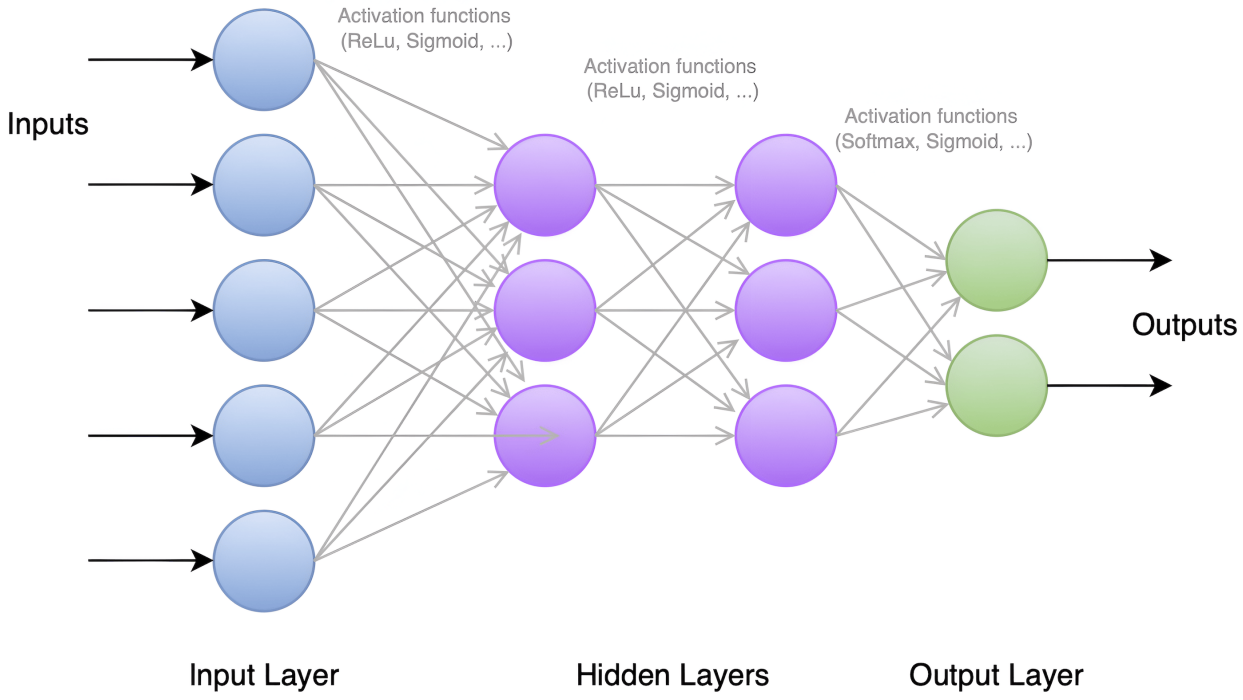

Figure 5.8: Diagram of the MLP model

The **MLP** can have from one to several hidden layers but the neurons that compose this network have the particularity of only transmitting information forward through the network. For this reason, the operation of an MLP model can be broken down into the following steps [44]:

- **Initialisation**, occurs at the input layer. This layer does not perform any computation on the input data, its mission is only to receive the information and distribute it to the subsequent layers. Each neuron in this layer will represent a characteristic of the input data set.
- **Processing**, takes place in the hidden layers. These layers are considered the computational heart of the MLP model because this is where all non-linear processing operations perform. Neurons in this layer apply activation functions on the input data, some of the most common functions are **ReLu** (Rectified Linear Unit), which provides an output for all positive input values and **Sigmoid** which transforms all input values into the range [0-1], these sigmoid functions have the advantage of both nonlinearity and differentiation.
- **Output**, which is produced in the output layer. This layer receives as inputs the outputs generated by the neurons of the last hidden layer and transforms this input into the final outputs of the model. As in the previous layer, this layer also has activation functions, and their use will depend on the distribution of data expected in the output. In this project, for the binary classification (0/1) that evaluates whether the newborn is crying or not, a **unipolar sigmoid** has been used.

The non-linearity of this model allows the network to be able to learn complex relationships and autonomously perform tasks that linear models are not capable of.

MLP implementation

The **MLP** model has been implemented in this project using the **Scikit-learn** library. This Machine Learning library is designed to be integrated well with other Python libraries, such as NumPy and SciPy, making it a competent and robust library for the purpose of this project. However, it should be noted that, although this library is not optimised to the same level of performance as the TensorFlow or PyTorch libraries, for researches such as the one which is being carried out in this project, it is a very competent alternative due to the moderate size of the data that has been collected to date.

Creating a model with a good performance is crucial for any project, for this reason it must be taken into account how the different parameters of the neural network influence and how the different neurons behave during training.

The artificial network that has been created consists on a **single hidden layer with 300 neurons**. The choice of a single hidden layer is driven by the limited amount of data available to date, the need to create a model with high computational efficiency and to meet the requirement that the risk of overfitting is minimised. On the other hand, using **MLPClassifier** the default activation function used for the hidden layers is the **ReLU** (Rectified Linear Unit) activation function, and it is the one that has been established due to the good performance it offers in deep learning problems.

In addition, the parameters that affect the operation of all the neurons in the network must be robust to the values they contain, such as **adaptive weight adaptation**. This way the model will automatically adjust the learning rate if the training is not obtaining favourable results, i.e. if the model stops improving. Then, the **regulation parameter L2** affects all the neurons in the network except those in the input layer and penalises the larger weights in order to achieve a model that generalises better by promoting smaller weights. This parameter has been defined with a value of **0.01**, which allows for a balance between the model's ability to correctly fit the data while avoiding the risk of overfitting.

Finally, the output layer is automatically selected according to the number of classes detected in the training of the model. If more than two classes are detected, the **softmax** activation function is automatically used.

5.6.2 Support Vector Machines (SVM)

Support Vector Machines (**SVM**) belong to the category of linear classifiers because the method it implements is the creation of a hyperplane for the separation of the defined spaces. This separation can be done by means of two techniques, the first and simplest is the separation over the original space. In order to achieve this separation, the input examples have to be linearly separable. If these are not, it is necessary to create a transformed space, called feature space [43].

This supervised learning algorithm tries to find the best hyperplane in the input space among the existing classes, as can be seen in the image 5.9 which, being a two-dimensional representation, the hyperplane can be represented as a straight line. But this is not the only representation that can be made of this algorithm. If the representation is generated in three or more dimensions, the hyperplane becomes a plane that still aims to divide the classes with the maximum possible margin between them.

The concept of this algorithm is actually a combination of computational theories that have existed for decades, such as hyperplane margins. These margins are defined as the smallest distance between the training points closest to the plane, known as support vectors, and the hyperplane. Achieving a robust margin boosts the model's ability to generalise unseen data, leading to greater confidence in the classification.

The basic principle of SVM is that, although it was initially created as a linear classifier, it has been developed in such a way that it can also be used in non-linear problems. This

modification is achieved by incorporating what are called kernel tricks. These kernels are defined as functions that allow the scalar product of two vectors in a transformed feature space to be calculated without having to compute the transformation. This makes it easier to operate with higher dimensional spaces where classes can be separated in a more efficient way without incurring computational costs. The most common kernels used in SVM are:

- **Linear**, this type of kernel is the simplest as it makes no modifications to the input space and uses the scalar product as the similarity measure.
- **Polynomial**, this type of kernel creates a polynomial feature space and is able to model non-linear and curved boundaries. A too low or too high degree can cause over-fitting or under-fitting.
- **Radial Basis Function (RBF)**, this kernel manages to project the data into an infinite dimensional space, being able to handle non-linear and multidimensional data although it can be sensitive to noise and outliers.
- **Sigmoid**, this kernel applies a sigmoid function to the data mimicking the behaviour of a neural network with a sigmoid activation function.

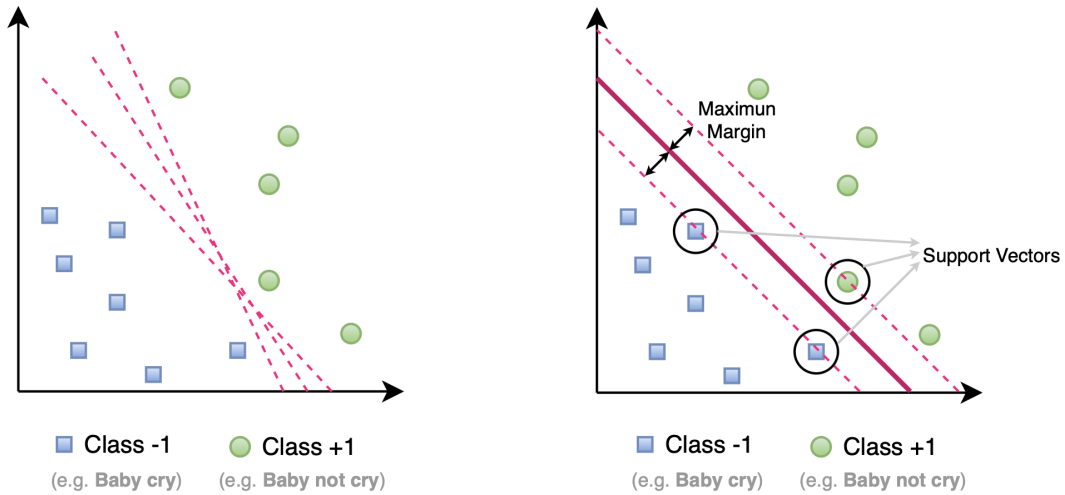

Figure 5.9: Diagram of the SVM model

Multi-class SVMs

Although support vector machines are designed for solving binary problems, it is true that they can be applied to problems containing multi-class classifications [45]. In order to apply SVM on multi-class classification problems, the dataset is divided into multiple binary classifications. These new subsets are then used to train the binary classification model (SVM). Using this approach, **One-vs-One (OvO)** and **One-vs-Rest (OvR)** strategies can be implemented.

Firstly, **One-vs-One (OvO)**, has the particularity of making pairs with the available classes, creating a data set of two classes in which one is the positive class and the other is the negative class, in this way, if there are K classes, $\frac{k(k-1)}{2}$ different classifiers will be trained, as can be seen in the image 5.10. As indeed may be observed, the number of classifiers grows quadratically with the number of classes, this aspect can make the model computationally expensive in terms of time and memory, especially with a large number of classes.

One-vs-One (OvO) is the default for non-linear kernels.

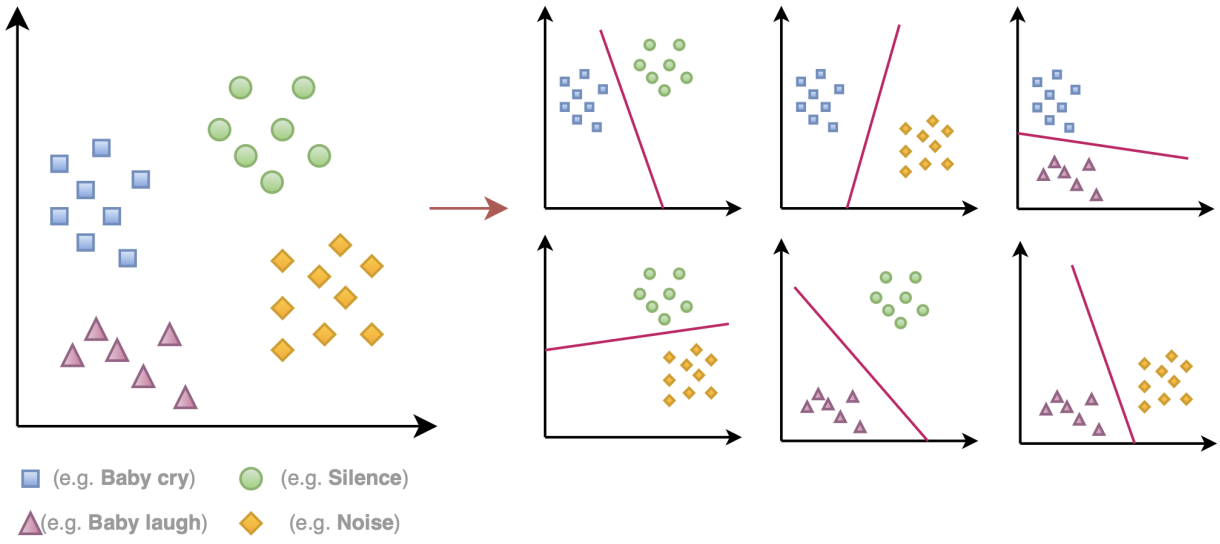

Figure 5.10: Diagram of the One-vs-One

Secondly, **One-vs-Rest (OvR)** again generates different binary classifiers to solve multi-class classification tasks. In this case, an SVM is trained for each class against the rest of the classes combined. That is, if there are a total of K classes, a total of K classifiers will be created, as can be seen in the image 5.11. For each class k, an SVM classifier is trained to assigns the examples of class k as positive and the rest of the classes as negative. The prediction is performed as follows: first, each of the k binary classifiers evaluates the test example (or instance). Then, the class whose classifier returns the highest value of the decision function is selected. This class is considered the most likely class for the observation according to the model.

One-vs-Rest (OvR) is the default value for linear kernels.

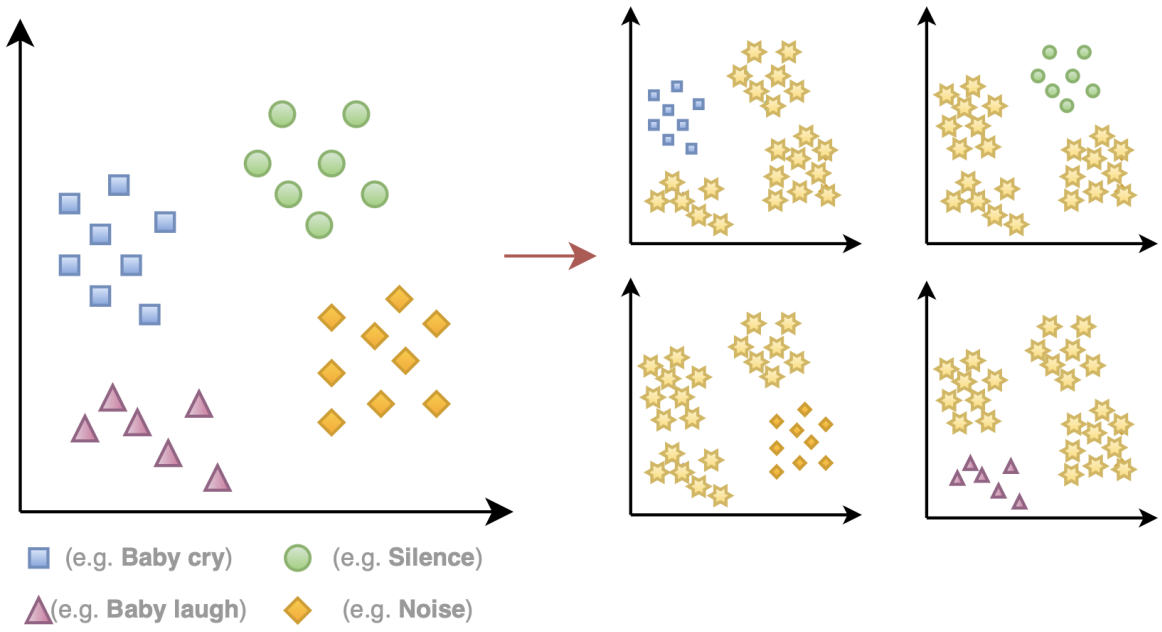

Figure 5.11: Diagram of the One-vs-Rest

SVM implementation

As with MLP, the **Scikit-learn** Python library has also been used to develop the model. The SVC (Support Vector Classification) class has been used for this purpose, as it allows the use of different kernel types and the possibility of calibrating the probabilities in the output.

As has been seen in the literature review, some studies highlight the use of the RBF kernel for its ability to handle non-linear data, and its good performance in classification problems. In this study, as it has been observed in the feature extraction in the images 5.7 it can be seen how the data are not linearly separable, that is, they cannot be separated with a straight line. Therefore, by using the **RBF kernel**, the data is projected into a higher dimensional space where it is more likely to find a hyperplane that separates the classes.

The control parameter C, controls the trade-off between maximising the margin or minimising the classification error. A high value such as $C = 10$ penalises classification errors more, forcing the model to fit the training data more accurately. However, a low heat such as $C = 1$ allows for a wider margin and tolerates more classification errors, which can also help with model generalisation.

In addition, **probability=True** is used to obtain probabilities on the model outputs instead of mere labels. This configuration makes it possible to interpret the SVM predictions in a more intuitive way. Instead of a simple binary classification indicating whether the baby is crying or not, the probabilities give an output between 0 and 1. This output is important to gradually observe changes in the baby's behaviour and to see how the probability of the baby crying varies.

5.6.3 Long Short-Term Memory (LSTM)

A Long Short-Term Memory (**LSTM**) network is a type of recurrent neural network (RNN) designed for the purpose of modelling long-term dependencies on sequential data [46]. The main purpose for which they were designed was to handle the vanishing gradient problem that occurs in traditional RNNs, in these networks error is backpropagated through their multiple layers causing important information to be lost in the process.

LSTM units are able to remember relevant data within a sequence and store this information during different instants of time. This feature makes the memory managed by these networks both long and short term, addressing the limitation that RNNs suffer from.

The architecture of each LSTM unit includes the following main components:

- **Cell**, is in charge of remembering values at different time intervals while the function of the gates is to regulate the flow of information both outside and inside the cell. This makes it feasible to maintain and modify information over long periods of time.
- **State of the cell**, at this point the information from the entry gate and the forgetting gate is combined, i.e. the new information is combined with the information from the previous stage as indicated by the activations of the gates.
- **Entry gate**, selects the new information to be kept within the cell state.
- **Forgetting gate**, chooses which parts of the information belonging to the previous cell will be deleted.
- **Output gate**, selects the relevant information that the cell state will use as output.

This architecture has been widely used in speech recognition tasks because these are able to maintain relevant information over long audio sequences, which makes them especially suitable for the classification of events such as the cry of a newborn.

LSTM implementation

The implementation that has been carried out in this project on the LSTM model has been done by using **TensorFlow** together with **Keras** in Python. This combination results in a seamless integration allowing the construction of deep, and customised models, combining multiple types of layers such as Dropout and Dense.

The model that has been developed in this project is composed of a sequential architecture with **two LSTM layers** and **two Dropout layers** in order to effectively handle the input data streams. The first **LSTM layer** is designed with **64 units** and is configured to return complete sequences, capturing long-term temporal dependencies of the input data.

Afterwards, a **Dropout layer** with a **rate of 0.5** is then applied, allowing a percentage of units from the previous layer to be disabled in order to help prevent overfitting during training. Then, the second **LSTM layer** is designed, this time with **32 units**, summarising the information from the previously processed sequences and returning a single output vector. This layer is also followed by a **Dropout layer** with a **rate of 0.3**, adding additional robustness to the model.

Finally, the model ends with a **Dense layer** with a **sigmoid activation**, adapted for a binary classification problem. The choice of sigmoid activation allows the model output to be interpreted as a probability between 0 and 1, which is especially useful for model evaluation using metrics such as the ROC curve and for decision making based on probabilistic thresholds.

5.7 Evaluate models

To evaluate the performance of the Machine Learning and Deep Learning models, presented in the previous section, it is essential to use **quantitative metrics**. These metrics allow an objective and accurate assessment of the true performance of the models by providing concrete numerical values that facilitate the understanding of the models' performance.

In this context of classification model evaluation, it is essential to keep in mind the following concepts that form the basis of many other metrics that will be discussed below:

- True Positives (**TP**), are the cases where the model correctly predicts the positive class, i.e. both the predicted value and the true value are positive.
- False Positives (**FP**), are the cases where the model incorrectly predicts the positive class, predicting a class as positive when its actual value is negative.
- True Negatives (**TN**), are the cases where the model correctly predicts the negative class, i.e. both the predicted value and the true value are negative.
- False Negatives (**FN**), are the cases where the model incorrectly predicts the negative class, predicting a class as negative when its actual value is positive.

These four definitions above provide the support for creating various quantitative metrics. The precision metrics used in this project are defined below and are also summarised in the image 5.12:

- **Precision**, this value measures how well or poorly the positive predictions of the model have been realised. This is done by calculating the proportion of true positives out of the total number of predicted positive values.
- **Recall**, this value identifies how accurately the positive instances of the model have been classified. This is done by calculating the proportion of true positives over the

total number of true positives.

- **F1-score**, this value is the harmonic mean between the recall and precision values.
- **Accuracy**, this value expresses the proportion of correct predictions out of the total number of predictions made. This metric is useful to get a general idea of the model's performance, but be aware that this result may not be sufficiently accurate. In unbalanced ensembles this value may give a result close to one which would be interpreted as a good result, but this may be because it is conditioned by the predominant class.
- **Confusion matrix**, this matrix clearly, and concisely represents the performance of the model. It shows a comparison between the predictions made by the model and the actual values. With this metric you can effectively compare whether the model's performance is being effective on unbalanced sets by looking at the percentage of data that is misclassified in each class.

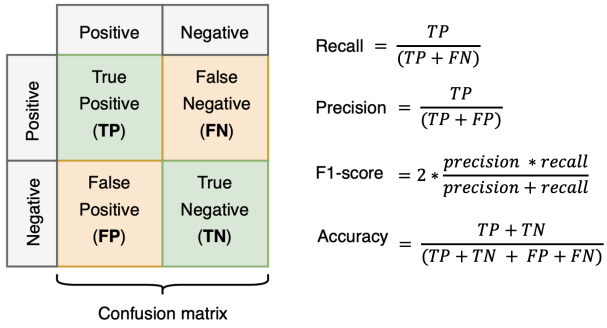

Figure 5.12: Summary of quantitative metrics

Nevertheless, the Receiver Operating Characteristic (**ROC**) curve, and the Area Under the Curve (**AUC**), are metrics that allow the performance of the model to be evaluated across different classification thresholds.

The **ROC** curve is constructed from the true positive rate (**TPR**) previously defined as recall versus the false positive rate (**FPR**). This representation is generated together with a representation of a random classifier, where in the generated graph what would correspond to

the Y-axis are the TPRs and what would correspond to the X-axis are the FPRs, as can be seen in the image 5.13. The result is interpreted by observing how close the generated curve is to the perfect classifier.

On the other hand, the **AUC** value summarises the information from the ROC curve into a single metric, i.e. a numerical value. For a perfect classifier this value would be equal to one, so the closer the result is to this value, the better the model is classifying.

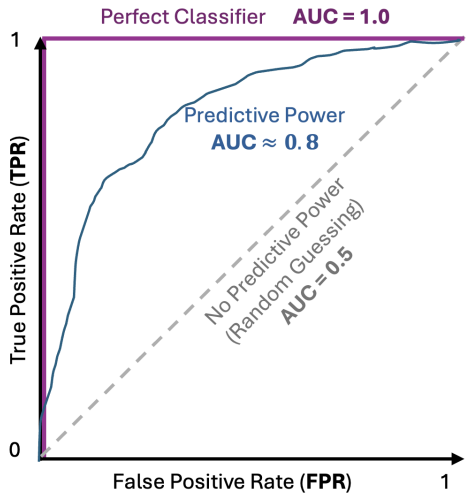

Figure 5.13: Functioning of the ROC Curve

5.8 Experiments

Aforementioned, we want to compare the rankings offered by Machine Learning and Deep Learning models in order to obtain the best one. In addition we want to compare other factors of interest: analysis of feature extraction techniques (**MFCC**, **LPC**), the consequences of applying **cross-validation** on the models (**MLP**, **SVM**, **LSTM**), the effects of L2 regularisation, the use of different kernel types in SVM and possible layer configurations in LSTM. Consequently, we start with experiments using an already labelled dataset (multi-class

classification) due to its high quality. These data clearly represent different types of sound, including baby cries, which makes them perfect for our purpose. The clarity and accuracy of these data are ideal for testing our models, ensuring that they work well and providing a solid basis on which to build the real project data (binary classification).

5.8.1 Multi-class classification

To start with simple experiments that beforehand were expected to obtain good results, we used a dataset available on GitHub[9] characterised by the clarity with which it represents the different sounds, including the plain one. All the experiments performed with this dataset are summarised in the table 5.2.

First of all, several experiments have been performed using Multilayer Perceptron (**MLP**) both "with", and "without" cross-validation. To perform these tests, we started by training a **MLP without cross-validation** with **300 hidden units**, dividing the dataset into 70% for the training part and allocating 30% of the remaining data for the test part. To perform this splitting process a random seed has been used to make this process consistent each time the code is run, ensuring that the training, and test sets are consistent across different runs. This **random seed** has been defined with a value equal to forty-two (42). In addition, the data has been standardised using *StandardScaler* to have a mean of 0, and a standard deviation of 1, allowing for better efficiency, and stability in training the models. This ensures that all features contribute equally to the model predictions.

Afterwards, a stratified **five-fold cross-validation** has been implemented in order to split the data into multiple training, and test subsets. During each iteration of the cross-validation, features have been scaled, and the model has been trained and evaluated with the corresponding

[9] https://github.com/giulbia/baby_cry_detection.git

test subset. In addition, all actual labels and predictions in each of the folds have been accumulated. This accumulation makes it possible to subsequently calculate evaluation metrics such as the confusion matrix, and classification reports.

These experiments are summarised in the notebook 2)Second_steps/4_Notebook-4_MLP_Multi-class.ipynb[10] .

Secondly, a larger number of experiments have been conducted using the Support Vector Machine (**SVM**) both "with", and "without" cross-validation. In this case, the experiments were also carried out with different classification strategies (One-vs-Rest and One-vs-One). On this occasion, the steps followed are the same as those described for the MLP model without cross-validation, except that in this case, a classification function (OvO or OvR) is applied before training the model. That is, for each feature extraction technique (MFCC and LPC) two different SVM models have been trained and evaluated, OvR training one classifier for each class and OvO training one classifier for each pair of classes.

Then, a **five-fold stratified cross-validation** has been implemented using a processing pipeline that includes: feature scaling, feature selection, and dimensionality reduction (PCA). Feature selection selects the best features based on the test score and **PCA** produces a set of **10 principal components**. With this new dataset as much variance as possible is preserved, making the model concentrate on the most informative features. This not only simplifies the model and speeds up the training process, but can also improve the generalisability of the model by reducing the risk of overfitting.

These experiments are summarised in the notebook 2)Second_steps/5_Notebook-5_SVM_Multi-class.ipynb[11].

[10] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/2)%20Second_steps/4_Notebook-4_MLP_Multi-class.ipynb

[11] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/2)%20Second_steps/5_Notebook-5_SVM_Multi-class.ipynb

Thirdly, the last experiments have been performed with Long Short-Term Memory (**LSTM**) and again "with", and "without" cross-validation. For the experiments without cross-validation, data splits of 70% for training and 30% for testing have been created with a **random seed** of value forty-two (42). Then, taking into account that a multi-class classification problem is being handled, a very important step is that the **labels** have been converted to the **one-hot** format. This model, when using the loss function *categorical_crossentropy* it is mandatory to convert the labels to one-hot in order to correctly calculate the loss and update the weights of the model. The last step during training, callbacks such as *EarlyStopping* and *ReduceLROnPlateau* were used to dynamically adjust the learning rate and prevent overfitting. *EarlyStopping* monitors the loss in the validation set and stops training if there is no improvement after a specified number of epochs (*patience*), thus preventing overfitting. *ReduceLROnPlateau* reduces the learning rate if the loss in the validation set stagnates, allowing the model to find a global minimum more efficiently.

Finally, the LSTM neural network was used with a **five-fold stratified cross-validation** to robustly evaluate the model's performance. In this case the model setup is the same as in the previous case except that it is now trained, and evaluated many times on different subsets of the data. These tests, as in the previous models, are performed using both MFCC and LPC.

These experiments are summarised in the notebook 2)Second_steps/6_Notebook-6_LSTM_Multi-class.ipynb[12]

[12] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/2)%20Second_steps/6_Notebook-6_LSTM_Multi-class.ipynb

Model	Features	Cross-validation	Techniques
MLP	MFCC	no	MLPClassifier
MLP	LPC	no	MLPClassifier
MLP	MFCC	yes	StratifiedKFold (5 splits), MLPClassifier
MLP	LPC	yes	StratifiedKFold (5 splits), MLPClassifier
SVM	MFCC	no	OneVsRestClassifier, kernel=linear
SVM	LPC	no	OneVsRestClassifier, kernel=linear
SVM	MFCC	no	OneVsOneClassifier, kernel=rbf
SVM	LPC	no	OneVsOneClassifier, kernel=rbf
SVM	MFCC	yes	StratifiedKFold (5 splits), OneVsRestClassifier, kernel=linear
SVM	LPC	yes	StratifiedKFold (5 splits), OneVsRestClassifier, kernel=linear
SVM	MFCC	yes	StratifiedKFold (5 splits), OneVsOneClassifier, kernel=rbf
SVM	LPC	yes	StratifiedKFold (5 splits), OneVsOneClassifier, kernel=rbf
LSTM	MFCC	no	LSTM, Dropout
LSTM	LPC	no	LSTM, Dropout
LSTM	MFCC	yes	StratifiedKFold (5 splits), LSTM, Dropout
LSTM	LPC	yes	StratifiedKFold (5 splits), LSTM, Dropout

Table 5.2: Summary of experiments on multi-class classification.

5.8.2 Binary classification

The experiments carried out with the data provided by the **HUBU** follow a similar approach to the previous experiments and are summarised in the table 5.4, taking into account that in this case we are dealing with a binary classification. Furthermore, it is important to note that in this occasion the classes are **unbalanced** so the techniques modify the balance of the classes in the dataset, which is crucial in this context. Initially some experiments were carried out without balancing techniques, with the result that most of the correct predictions were concentrated in the majority class and more than 50% of the minority class was misclassified. After observing these unfavourable results, the project focused its attention on experiments with balancing techniques such as **oversampling**, and **undersampling**. As these techniques are fundamental for handling class imbalance, the experiments that have been performed without them are not included in the project.

First of all, experiments with the **MLP** model have been started for binary classification that follow a very similar approach to multiclass classification. Tests have been performed both "with", and "without" cross-validation, and both follow the same process. It starts with feature normalisation using *StandardScaler*, and then combines oversampling, and undersampling using *SMOTETomek* to address class imbalance. In addition, the classification threshold has been adjusted using the **ROC** curve to optimise the trade-off between true positive rate (TPR) and false positive rate (FPR). Again, these tests are performed using MFCC and LPC feature extraction techniques.

These experiments are summarised in the notebook 3)Final_steps/8_Notebook-8_MLP_Binary.ipynb[13].

Secondly, experiments have been followed by training and validating the **SVM** model with both MFCC and LPC feature extraction techniques. In addition, experiments "with", and

[13] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/3)%20Final_steps/8_Notebook-8_MLP_Binary.ipynb

"without" cross-validation have been carried out again, but this time the control parameter C has been modified by increasing its value in the tests with cross-validation. In the tests **without cross-validation** the value is $C = 1$ while in the tests **with cross-validation** the value is $C = 10$. This modification allows the model to fit the training data more closely, which is particularly beneficial to prevent overfitting.

Moreover, the main steps have been kept intact in both experiments: feature normalisation, class balancing with *SMOTETomek*, model calibration with *CalibratedClassifierCV*, adjustment of the classification threshold using the ROC curve.

These experiments are summarised in the notebook 3)Final_steps/9_Notebook-9_SVM_Binary.ipynb[14]

Thirdly, experiments with the **LSTM** model for binary classification have been completed, both "with", and "without" cross-validation, and by comparing the use of MFCC and LPC feature extraction techniques. In all experiments the same model structure is maintained with **two LSTM** layers and **two Dropout** layers using an output layer with *sigmoid* activation instead of *softmax*, and the loss function *binary_crossentropy*, being in this case a binary classification. Certainly, this time we have also used **oversampling** and **undersampling** techniques using *SMOTETomek* to deal with class imbalance and scaled the features using *StandardScaler*.

Finally, in these **LSTM** experiments, techniques are used to improve the efficiency, and robustness of training in these deep neural networks. **EarlyStopping** reduces unnecessary training time in these deep models and **ModelCheckpoint** stores the most accurate, and generalisable version of the model, even if further training leads to a worsening of performance. These techniques show great utility in models such as LSTM due to their multi-layered structure, and the large number of parameters they possess, which translates into a high modelling capacity making them more susceptible to changes in the training data. This

[14] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/3)%20Final_steps/9_Notebook-9_SVM_Binary.ipynb

means that these models can learn complex patterns but also noise, which makes them more susceptible to changes in training data compared to simpler models such as MLP or SVM.

These experiments are summarised in the notebook 3)Final_steps/10_Notebook-10_LSTM_Binary.ipynb[15]

[15] https://github.com/lnc1002/TFM-Newborn_Cries_Classification/blob/4d0c85c8b15c2c117ca147f7c9625d47ff986ad8/src/3)%20Final_steps/10_Notebook-10_LSTM_Binary.ipynb

Model	Features	Data Processing	Cross-validation	Techniques
MLP	MFCC	SMOTETomek, StandardScaler	no	MLPClassifier
MLP	LPC	SMOTETomek, StandardScaler	no	MLPClassifier
MLP	MFCC	SMOTETomek, StandardScaler	yes	StratifiedKFold (5 splits), MLPClassifier
MLP	LPC	SMOTETomek, StandardScaler	yes	StratifiedKFold (5 splits), MLPClassifier
SVM	MFCC	SMOTETomek, StandardScaler	no	SVC, CalibratedClassifierCV, kernel=rbf, C=1
SVM	LPC	SMOTETomek, StandardScaler	no	SVC, CalibratedClassifierCV, kernel=rbf, C=1
SVM	MFCC	SMOTETomek, StandardScaler	yes	StratifiedKFold (5 splits), SVC, CalibratedClassifierCV, kernel=rbf, C=10
SVM	LPC	SMOTETomek, StandardScaler	yes	StratifiedKFold (5 splits), SVC, CalibratedClassifierCV, kernel=rbf, C=10
LSTM	MFCC	SMOTETomek, StandardScaler	no	LSTM, Dropout
LSTM	LPC	SMOTETomek, StandardScaler	no	LSTM, Dropout
LSTM	MFCC	SMOTETomek, StandardScaler	yes	StratifiedKFold (5 splits), LSTM, Dropout
LSTM	LPC	SMOTETomek, StandardScaler	yes	StratifiedKFold (5 splits), LSTM, Dropout

Table 5.4: Summary of experiments on binary classification.

5.9 Results

This section elaborates the results obtained from both the multi-class, and the binary classification of audio files. However, greater importance will be given to the results obtained in the **binary classification**, as this is the data provided by the HUBU, and is that which will enable the achievement of the project objectives.

As mentioned above, to evaluate the performance of the models, different quantitative metrics have been used in the experiments, reserving qualitative metrics only for the binary experiments. These metrics express a more details on the effectiveness of the models, and are detailed in the experiment notebooks cited in the previous *Experiment* section.

5.9.1 Multi-class classification

This section presents the results generated by the Machine Learning and Deep Learning models applied to the multi-class classification. As previously mentioned, this classification consists of four classes: **Crying_baby (0), Silence (1), Noise (2), Baby_laugh (3)**. The results of the different experiments are summarised in the table 5.6.

The **MLP** model with MFCC features exhibits a good performance, and more stable results in the different classes, with an overall accuracy of 95% without cross-validation, and slightly improves to 96% with cross-validation. This model generates an adequate classification in the *Crying_baby* category, which is highly relevant to this project, and ensures proper detection of newborns' cries. However, in this same model it can be observed how the LPC features show a slight decrease in the functioning of the *Noise* category. This can cause various issues, such as detecting false positives in the detection or classification of crying.

Model	Features	Cross-validation	F1-score (0, 1, 2, 3)	Accuracy
MLP	MFCC	No	0.93, 0.97, 0.91, 1.00	0.95
MLP	LPC	No	0.92, 1.00, 0.83, 1.00	0.95
MLP	MFCC	Yes	0.93, 1.00, 0.93, 0.99	0.96
MLP	LPC	Yes	0.89, 1.00, 0.87, 1.00	0.94
SVM	MFCC	No	0.86, 0.98, 0.80, 1.00	0.92
SVM	MFCC	No	0.88, 1.00, 0.73, 1.00	0.92
SVM	LPC	No	0.98, 1.00, 0.96, 1.00	0.98
SVM	LPC	No	0.90, 1.00, 0.77, 1.00	0.93
SVM	MFCC	Yes	0.82, 0.97, 0.70, 0.96	0.87
SVM	MFCC	Yes	0.84, 1.00, 0.80, 1.00	0.91
SVM	LPC	Yes	0.93, 0.99, 0.91, 1.00	0.96
SVM	LPC	Yes	0.88, 1.00, 0.85, 1.00	0.94
LSTM	MFCC	No	0.84, 1.00, 0.76, 0.99	0.90
LSTM	LPC	No	0.88, 1.00, 0.73, 1.00	0.92
LSTM	MFCC	Yes	0.81, 1.00, 0.80, 0.97	0.90
LSTM	LPC	Yes	0.83, 1.00, 0.73, 1.00	0.92

Table 5.6: Summary of results on multi-class classification.

On the other hand, the **SVM** model does not follow the same pattern as previously observed. In this model, the LPC features together with the OvO strategy that show the best results, especially without cross-validation, with an accuracy of 98%. Additionally, as in the previous case, this model shows a slightly improved performance in the classification of *Crying_baby* however a better performance in the *Noise* class stands out. Consequentially, the researcher

can effectively distinguish between the cry of a baby, and the background noise, which has an astounding impact on our project.

Finally, the **LSTM** model obtained very similar results when using MFCC features with, and without cross validation, with an accuracy of 90%. Once more, the LPC features demonstrate a satisfactory performance with an accuracy of 92%, and a consistency in the classifications of the *Crying_baby*, and *Noise* classes.

5.9.2 Binary classification

This comprises of categorizing newborn audios into two classes that is *Baby_cry* and *Baby_not_cry*. The results from this classification provide the real value in this project, as they focus on the detection of baby crying in a typical or representative setting. This data has been directly obtained from a hospital, so the *Baby_not_cry* class comprises only environmental sounds typical in a hospital setting. The usage of this data ensures that the trained models are highly generalizable, and applicable in other health-related organizations, and similar environments.

This section describes the quantitative, and qualitative evaluation conducted in the binary classification including the pertinent metrics such as accuracy, and AUC, as well as a graphical representation of the probability of crying over time.

Quantitative experiments

The results obtained in the binary classification are summarised in the table 5.8. As seen in this table, all models show a higher capacity to detect the *Baby_not_cry* class compared to the *Baby_cry* class. These results are as expected given the diversity of newborn cries in relation to amplitude, pitch, etc. It should be noted that accuracy in classifying the

Baby_not_cry class is critical to reduce false positives, while better classification of the *Baby_cry* class ensures a precise detection of crying.

Firstly, the **MLP** model with **MFCC** features shows strong results in both the "with", and "without" cross-validation, notably the results obtained by the former, which generates an overall accuracy of 89%. On this occasion, it can also be observed how the LPC features underperform, especially in the *Baby_cry* category, which could be problematic in the accurate detection of crying.

Secondly, in the **SVM** model it is observed both models "with" and "without" cross-validation present a solid performance, generating somewhat more accurate results in the experiments performed with **MFCC**, and the use of cross-validation. With this experiment, a remarkable accuracy is achieved in both classes, which can be seen in the F1-score values of 0.71 for the *Baby_cry* class, and 0.93 for the *Baby_not_cry* class. By a very small margin, LPC is slightly less accurate than MFCC which could affect the system's ability to reliably detect crying.

Finally, the **LSTM** model shows promising performance especially when the cross-validation technique is employed. The LPC features show an acceptable performance, although inferior to that of **MFCC**. With MFCC features, the model shows an F1-score of 0.70 for *Baby_cry*, and 0.93 for **Baby_not_cry** without cross-validation, with an accuracy of 0.89. However, the combination that best detects the *Baby_cry*, and the *Baby_not_cry* class, is also by means of the MFCC features but this time with cross-validation, reaching an F1-score of 0.72, and 0.94 respectively, and an **accuracy of 90%**, which was the highest in all conducted experiments. These findings suggest that LSTM models can be effective for the accurate detection of infant crying in hospital settings.

Model	Features	Cross-validation	F1-score (0, 1)	Accuracy	AUC
MLP	MFCC	No	0.69, 0.92	0.87	0.95
MLP	LPC	No	0.62, 0.89	0.83	0.91
MLP	MFCC	Yes	0.71, 0.93	0.89	0.95
MLP	LPC	Yes	0.61, 0.90	0.84	0.91
SVM	MFCC	No	0.67, 0.91	0.86	0.94
SVM	LPC	No	0.61, 0.90	0.84	0.89
SVM	MFCC	Yes	0.70, 0.93	0.88	0.94
SVM	LPC	Yes	0.64, 0.92	0.87	0.90
LSTM	MFCC	No	0.70, 0.93	0.89	0.95
LSTM	LPC	No	0.64, 0.92	0.86	0.90
LSTM	MFCC	Yes	0.72, 0.94	0.90	0.94
LSTM	LPC	Yes	0.63, 0.92	0.86	0.90

Table 5.8: Summary of results on binary classification.

Qualitative experiments

The qualitative analysis focuses on visual assessment, so in this case different graphs reflecting the probability of crying over time have been generated. These results are based on the audio files that were not used in the previous experiments, and the directories are labeled as: *Mild_encephalopathy* (1 file), *Severe_encephalopathy* (1 file), *Moderate_encephalopathy* (1 file), and *No_encephalopathy* (2 files). To compare the results produced by the models, the graphs generated are going to be analyzed. In these graphs 5.14, a binary classification (0-1) can be observed where the probability of crying is either 100% or 0%. The precision or accuracy of the models can be determined if the pertinent graphs are somewhat similar to

the first binary classification. This comparison identifies the nuances or other audio aspects thus offering a more representative understanding of the nature of the newborn cries.

The models' performance in identify audio cries from the new data provided improves slightly when using **MFCC features**, and applying **cross-validation**. For this reason, specific functions are used in this process that include the loading of the audio files, the extraction of MFCC features, the scaling of these features as required by the model, and the prediction of the probabilities for each class. Subsequently, the probabilities of crying per second in each audio file are plotted for a better visual understanding of the results.

As can be seen in the images 5.14, 5.15 all models predict crying quite well. In the case of *Moderate_encephalopathy* around the 18^{th} second a crying label is located but the models do not correctly detect the sound, in fact, the MLP model does not even detect crying at that moment. The rest of the models do detect something but with a probability of around 40%. When the audio is played back to that time frame, a slight moaning cry is heard, so the models are not completely overlooking it.

A snapshot of all the graphs identifies that the LSTM model performed slightly better than the other models in precisely predicting the probability of crying in the presented audios.

*The following images are composed of four graphs: the first (**blue**) shows the **label** graph, the second (**pink**) shows the prediction with **MLP**, the third (**green**) shows the prediction with **SVM**, the fourth (**orange**) shows the prediction with **LSTM**.*

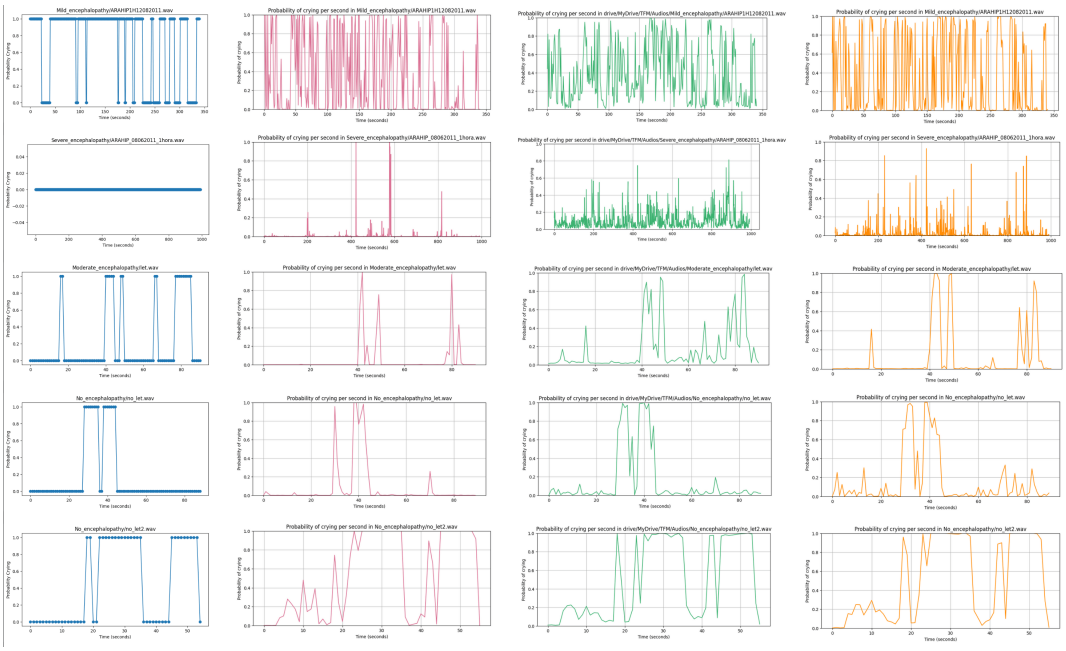

Figure 5.14: Predictions with different models *a)* **blue** label graph textitb) **pink** prediction with MLP textitc) **green** prediction with SVM textitd) **orange** prediction with LSTM

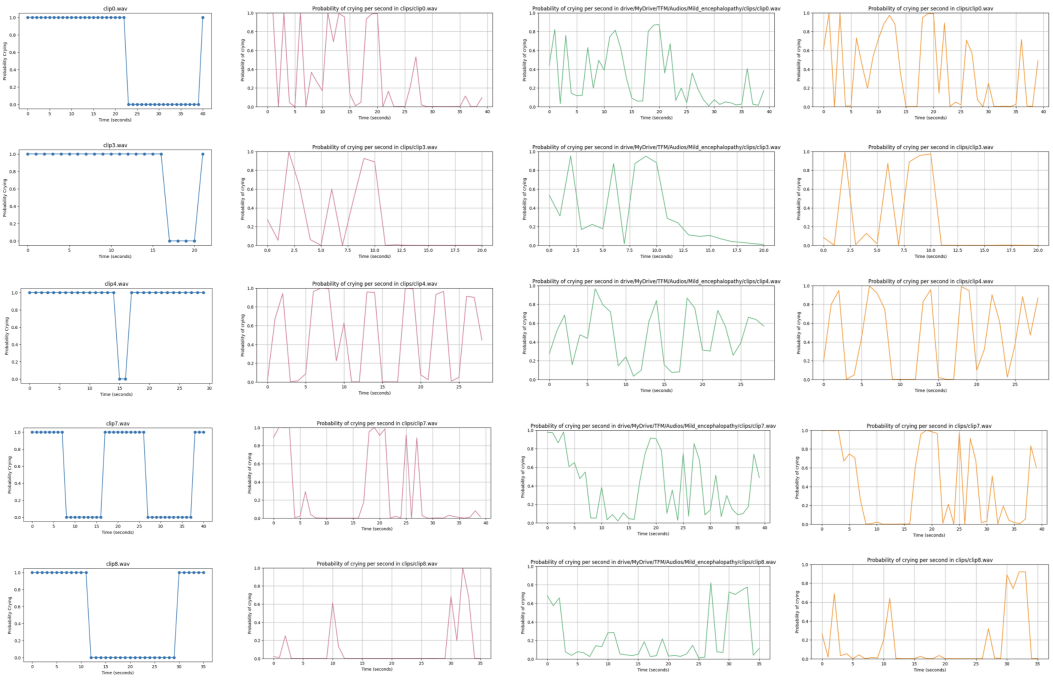

Figure 5.15: Predictions with different models in clips audio, *a)* **blue** label graph textitb) **pink** prediction with MLP *c)* **green** prediction with SVM *d)* **orange** prediction with LSTM

Chapter 6

Conclusions and Outlook

6.1 Conclusions

Firstly, it should be noted that this project has been based on an exhaustive investigation of the state of the art, looking at methods that have been previously developed for applications similar to the ones in this project. This approach allows this research to be seen as a review article.

The more in-depth discussion of the experiments has been done in *Experiments* section, while the results of these experiments can be observed in *Results* section. Therefore, we can conclude that the combination of different hyperparameters in different models gives good results in newborn cry classification, both quantitatively and qualitatively (table 5.8 and images 5.14, 5.15).

As discussed throughout the project, early detection of HIE is essential to be able to treat the neonate as quickly as possible and thus significantly improve long-term adverse effects [47]. For this reason, Deep Learning, and Machine Learning models can be tremendously useful to improve diagnostic accuracy, treatment efficacy, and provide decision support for potential

complications in such vulnerable patients.

This project aims to emphasise the real relevance that the use of Deep Learning, and Machine Learning algorithms can have in solving audio analysis problems, especially in the context of child health. Looking at the results obtained, and taking into account that the number of samples is not too large, in particular the number of crying samples, very promising classifications have been achieved. This suggests that such models could be used in the future to detect infant crying in clinical settings and aid in the diagnosis of HIE.

It is also important to note that this project achieves promising results by using data from a typical or representative setting, provided by a hospital maternity ward. This makes it generalisable to other hospitals or clinical settings, strengthening the reliability and usefulness of Deep Learning, and Machine Learning techniques.

6.2 Future work

Given the timescale of the project and the length of this report, it has been necessary to select those objectives that were achievable within the limits of the project. There were two sets of experiments to be carried out: testing and evaluating different classification models, including **MLP**, **SVM** and **LSTM**, using public multi-class data, and applying these same models to real binary data provided by the HUBU.

Both the first and second groups of experiments have been successfully completed within the theoretical framework of the project and with the data available to date, suggesting that within these working groups there is still room to extend the tasks performed in order to achieve more complete results.

First, the experiments can be further extended by **testing different hyperparameter**

configurations, new models, and architectures. First, we could implement deeper models and analyse the effects of using more layers. We could also use advanced hyperparameter optimisation techniques to improve model performance. Last but not least, we could explore the use of hybrid architectures combining vision and audio models, taking advantage of the results of this project together with those of other projects involving the vision part to obtain a more holistic assessment of the baby's state.

Additionally, the **process of labelling audio data could be automated** using semi-supervised learning techniques. Implementing these techniques would reduce the manual labelling task, and improve the efficiency of the model training process. This is especially relevant given that the current volume of data is limited, as only one hospital is being worked with. However, if this project could be integrated with a larger number of hospitals in the future, the amount of data collected will increase significantly.

Another potential experiment is to integrate **real-time data analysis** for early medical alerts. This could be developed through a system that not only classifies baby sounds, but also identifies patterns that may indicate urgent medical conditions, sending alerts to healthcare professionals. As discussed in the project, this line of research follows audio classification, but there are other lines of research that are classifying the image. These lines of research could be extended to also classify, and evaluate biomarkers and perform real-time analysis.

Another line of future work is to include the possibility of **learning about audio tones**, not only by determining whether the baby cries or not, but also how the baby cries. To achieve this purpose, different types of crying could be analysed, labelled and classified. This would be useful to differentiate between times when the baby starts or stops crying, and times when the baby cries more intensely. Furthermore, it could be useful to determine the type of the baby's crying, since, although probabilities are currently provided and the intensity of the crying can be observed in relation to the probability, this does not provide an output of whether the crying is weak, strong, consistent, etc.

The last group of experiments that could be carried out is the **analysis of the doctors' comments** seen in the videos provided by the HUBU. The doctors provide guidelines about the baby's condition, such as what reaction they are experiencing to a stimulus or the degree of HIE they present. So another possible future work could be to evaluate and classify these comments to improve the accuracy of the assessment of the baby's condition. This approach would integrate additional qualitative data, and could lead to a more accurate and contextualised classification of infant states.

Bibliography

[1] Riley Children's Health. Neonatal encephalopathy. https://www.rileychildrens.org.

[2] MedlinePlus. Apgar test. https://medlineplus.gov/ency/article/003402.htm, n.d.

[3] B.D. Power, J. McGinley, D. Sweetman, and J.F. Murphy. The modified sarnat score in the assessment of neonatal encephalopathy: A quality improvement initiative, 2019.

[4] P. Clarke et al. Management and investigation of neonatal encephalopathy: 2017 update. *BMJ*, 102(4):F346, 2017. urlhttps://fn.bmj.com/content/102/4/F346.

[5] Alfredo Garcia-Alix, Juan Arnaez, Gemma Arca, Thais Agut, Ana Alarcon, Ana Martín-Ancel, Montserrat Girabent-Farres, Eva Valverde, and Isabel Benavente-Fernández. Development, reliability, and testing of a new rating scale for neonatal encephalopathy. *The Journal of Pediatrics*, 235:83–91, 2021.

[6] Jennifer J Kurinczuk, Melanie White-Koning, and Nadia Badawi. Epidemiology of neonatal encephalopathy and hypoxic–ischaemic encephalopathy. *Early human development*, 86(6):329–338, 2010.

[7] Harvey B Sarnat and Margaret S Sarnat. Neonatal encephalopathy following fetal distress: a clinical and electroencephalographic study. *Archives of neurology*, 33(10):696–705, 1976.

[8] Mohammed Hammoud, Melaku N. Getahun, Anna Baldycheva, and Andrey Somov.

Machine learning-based infant crying interpretation. *Frontiers in Artificial Intelligence*, 7, 2024.

[9] Rodica Ileana Tuduce, Horia Cucu, and Corneliu Burileanu. Why is my baby crying? an in-depth analysis of paralinguistic features and classical machine learning algorithms for baby cry classification. *IEEE*, 2018.

[10] A. Ruiz-Zafra, D. Precioso, B. Salvador, S. P. Lubián-López, J. Jiménez, I. Benavente-Fernández, and L. C. Gontard. Neocam: An edge-cloud platform for non-invasive real-time monitoring in neonatal intensive care units. *IEEE Journal of Biomedical and Health Informatics*, 2023.

[11] Anderson Rocha, João Paulo Papa, and Luis A. A. Meira. How far do we get using machine learning black-boxes? *Institute of Computing, University of Campinas (UNICAMP), Department of Computer Science, State University of São Paulo (UNESP), Department of Science and Technology, Federal University of São Paulo (UNIFESP)*, 2023.

[12] K. A. Stewart and A. H. Segars. An empirical examination of the concern for information privacy instrument. *Information Systems Research*, 13(1):36–49, 2002.

[13] J. Dickson. *Enduring and emerging challenges of informed consent.* Delivering Digital Drugs, 2015.

[14] E. A. Whitley, N. Kanellopoulou, and J. Kaye. Consent and research governance in biobanks: evidence from focus groups with medical researchers. *Public Health Genomics*, 15(5):232–242, 2012.

[15] M Rahimzad, A Moghaddam Nia, H Zolfonoon, J Soltani, A Danandeh Mehr, and H H Kwon. Performance comparison of an lstm-based deep learning model versus conventional machine learning algorithms for streamflow forecasting. *Water Resources Management*, 35(12):4167–4187, 2021.

[16] H Oukhouya and K El Himdi. Comparing machine learning methods—svr, xgboost,

lstm, and mlp—for forecasting the moroccan stock market. In *Computer Sciences & Mathematics Forum*, volume 7, page 39. MDPI, 2023.

[17] S K Lakshminarayanan and J P McCrae. A comparative study of svm and lstm deep learning algorithms for stock market prediction. In *AICS*, pages 446–457, December 2019.

[18] Christopher J Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. Key challenges for delivering clinical impact with artificial intelligence. *Nature Medicine*, 27:27–33, 2021.

[19] K Murphy, E Di Ruggiero, R Upshur, D J Willison, N Malhotra, J C Cai, and J Gibson. Artificial intelligence for good health: a scoping review of the ethics literature. *BMC Medical Ethics*, 22:1–17, 2021.

[20] MA Tagin, CG Woolcott, MJ Vincer, RK Whyte, and DA Stinson. Hypothermia for neonatal hypoxic ischemic encephalopathy: an updated systematic review and meta-analysis. *Archives of Pediatrics & Adolescent Medicine*, 166(6):558–566, 2012.

[21] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), 2016.

[22] N Naik, B M Hameed, D K Shetty, D Swain, M Shah, R Paul, and B K Somani. Legal and ethical consideration in artificial intelligence in healthcare: who takes responsibility? *Frontiers in Surgery*, 9:266, 2022.

[23] F Griffin. Artificial intelligence and liability in health care. *Health Matrix*, 31:65, 2021.

[24] F Molnár-Gábor. Artificial intelligence in healthcare: doctors, patients and liabilities. In *Regulating Artificial Intelligence*, pages 337–360. Springer, 2020.

[25] Pahulpreet Singh Kohli and Shriya Arora. Application of machine learning in disease prediction. In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pages 1–5. IEEE, 2019.

[26] Kai Hwang Lu Wang Lin Wang Min Chen, Yixue Hao. Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*, 5:8869–8879, 2017.

[27] Sayeed Uddin, Asif Khan, M. Hossain, et al. Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, 19(1):281, 2019.

[28] Yun-Chia Liang, Iven Wijaya, Ming-Tao Yang, Josue Rodolfo Cuevas Juarez, and Hou-Tai Chang. Deep learning for infant cry recognition. *International Journal of Environmental Research and Public Health*, 19(6311), 2022.

[29] Azadeh Bashiri and Roghaye Hosseinkhani. Infant crying classification by using genetic algorithm and artificial neural network. *Acta Medica Iranica*, 58(10):531–539, 2020.

[30] Ashwini K, P. M. Durai Raj Vincent, Kathiravan Srinivasan, and Chuan-Yu Chang. Deep learning assisted neonatal cry classification via support vector machine models. *Frontiers in Public Health*, 9:670352, 2021.

[31] Azlee Zabidi, Wahidah Mansor, Lee Yoot Khuan, Ihsan Mohd Yassin, and Rohilah Sahak. Classification of infant cries with hypothyroidism using multilayer perceptron neural network. 2009.

[32] Chunyan Ji, Thosini Bamunu Mudiyanselage, Yutong Gao, and Yi Pan. A review of infant cry analysis and classification. *IEEE Reviews in Biomedical Engineering*, 13:77–89, 2020.

[33] Silvia Orlandi, Carlos Alberto Reyes Garcia, Andrea Bandini, Gianpaolo Donzelli, and Claudia Manfredi. Application of pattern recognition techniques to the classification of full-term and preterm infant cry. *Journal of Voice*, 30(6):656–663, 2016.

[34] Chuan-Yu Chang, Sweta Bhattacharya, P. M. Durai Raj Vincent, Kuruva Lakshmanna, and Kathiravan Srinivasan. An efficient classification of neonates cry using extreme gradient boosting-assisted grouped-support-vector network. *IEEE Access*, 8:60729–60741, 2020.

[35] N.S. Abdul Wahid, P. Saad, and M. Hariharan. Automatic infant cry pattern classification for a multiclass problem. *Journal of Telecommunication, Electronic and Computer Engineering*, 8(12):21–24, 2016.

[36] Z. K. Abdul and A. K. Al-Talabani. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access*, 10:122136–122158, 2022.

[37] John Bradbury. Linear predictive coding. 2000.

[38] GrabVoice. Spectral analysis in speech processing: Cepstrum smoothing and lpc analysis calculation, 2024.

[39] L. C. Paul, A. A. Suman, and N. Sultan. Methodological analysis of principal component analysis (pca) method. *International Journal of Computational Engineering & Management*, 16(2):32–38, 2013.

[40] S. Lipovetsky. Pca and svd with nonnegative loadings. *Pattern Recognition*, 42(1):68–76, 2009.

[41] S. Arora, W. Hu, and P. K. Kothari. An analysis of the t-sne algorithm for data visualization. In *Conference on Learning Theory*, pages 1455–1462. PMLR, July 2018.

[42] D. Berrar. Cross-validation. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 2019.

[43] D. Mercado Polo, L. Pedraza Caballero, and E. Martínez Gómez. Comparación de redes neuronales aplicadas a la predicción de series de tiempo. *Prospectiva*, 13(2):88–95, 2015.

[44] Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, and Nikos

Mastorakis. Multilayer perceptron and neural networks. *Faculty of Electromechanical and Environmental Engineering, University of Craiova1 Faculty of Engineering, "Aurel Vlaicu" University of Arad2 "Elena Cuza" College of Craiova3 ROMANIA, Technical University of Sofia4 BULGARIA*, 1(1):1–10, 2019.

[45] Wenjun Liao. Lecture slides: Chapter 9, 2019. Accessed: 2024-06-04.

[46] H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

[47] MPH Yvonne Wu, MD. Clinical features, diagnosis, and treatment of neonatal encephalopathy. Literature review current through: May 2024. This topic last updated: Jan 30, 2024.

Appendix A

Annexes

All images presented in this report have been created manually. Images depicting infant faces have been generated using Artificial Intelligence (A1), ensuring that no real patient data or identities are compromised.

Due to privacy and data protection laws, this project also does not include the original data identifying the babies, nor have names or any other data that could facilitate the identification of study participants or family members been released.

Research plan

This section focuses on the temporal planning of the project. Planning a project correctly is a very important aspect as it allows for efficient time management, prioritising the most specific and relevant tasks of the project. In addition, keeping a realistic task schedule is essential to achieve the objectives set for the project within the timeframe available.

To show the time spent on each task, a Gantt Chart has been developed. This tool is widely used in project management as it allows the project schedule to be visualised, showing the tasks, their duration and the dependencies between them.

The Gantt Chart that has been made is structured by weeks and covers a total of 17 weeks, from 12/02/2024 to 07/06/2024. The weeks are detailed as follows:

- Week 1: from 12/02/2024 to 18/02/2024
- Week 17: 03/06/2024 to 07/06/2024

This diagram includes all the main tasks that a Master's thesis project should have, as well as some specific tasks for this particular project. Thus, eight major phases have been defined, providing a clear and manageable structure for the development of the work.


WEEKS: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Research similar works
Analyse studies
Select metrics and hyperparameters
Select models for this study
Preparing the data for the project
Analysing available data
Organise the data
Label the data
Search for public data to use for our project
Search public data
Analyse and compare datasets
Problem analysis and proposed solutions
Extract audio
Test audio type and duration
Transform audio to image
Test a classification with images
Extract features
Select possible models
Creating different models
Implement MLP model on multiclass data
Implement SVM model on multiclass data
Implement LSTM model on multiclass data
Evaluate models with different performance metrics
Evaluate feasibility of applying the models on binary data
Implement MLP model on binary data
Implement SVM model on binary data
Implement LSTM model on binary data
Evaluate models with different performance metrics
Comparison of results
Analyse the results of feature extraction techniques
Compare the results between yes/no cross-validation
Compare the results of different models
Representation of results
Creation of the report