



N° Ref :.....

Faculty of Mathematics and Computer Science
Department of Computer Science

**Specialty: Information and Communication Sciences
and Technologies.
Master's Thesis**

An approach for infant cry sound classification

Presented by :

Bellili Alaeddine

Bouabibsa Hicham

Supported by the jury :

Faiza Afri	MAA	Abdelhafid Boussouf Univ, Mila	President
Guettiche Mourad	MCB	Abdelhafid Boussouf Univ, Mila	Examiner
Deffas Zineb	MAA	Abdelhafid Boussouf Univ, Mila	Supervisor

Academic Year :2025/2026

ACKNOWLEDGEMENTS

We owe our deepest gratitude to **Allah**, the Almighty, for granting us the clarity of mind and the resilience to transform this academic endeavor into reality.

This research would not have been possible without the invaluable mentorship of our supervisor, **Deffas Z.** We are incredibly fortunate to have benefited from her expertise, critical insights, and inspiring dedication, which constantly pushed us toward excellence.

Beyond the academic sphere, our heartfelt appreciation goes to our families. We are profoundly grateful for their silent sacrifices, unwavering faith in our potential, and for providing a sanctuary of peace and motivation during our most challenging moments.

Finally, we salute our peers and everyone who crossed our paths during this journey, offering a word of encouragement or a helping hand. Your presence made this rigorous process a memorable experience.

BOUABIBSA HICHAM
BELLILI ALAEDDINE

Abstract

Infant cry analysis is a promising application of artificial intelligence and audio signal processing, as crying represents the primary means of communication during the first months of life. This work aims to develop an intelligent system capable of automatically classifying infant cries according to the need expressed by the infant.

The study is based on the Donate-a-Cry dataset, which contains five cry categories: hungry, discomfort, belly pain, burping, and tiredness. This dataset reflects real-world infant crying conditions, where hungry is one of the most common causes of crying during early infancy, naturally resulting in a larger representation of this category compared to the others. Consequently, classification challenges arise that require models capable of maintaining balanced performance across all cry types.

To address these challenges, a Hierarchical Classification Framework (HCF) combining Support Vector Machines (SVM) and Random Forest (RF) is proposed, together with acoustic feature extraction, data augmentation, and SMOTE-based oversampling techniques to improve model generalization. In addition, a **Streamlit** based web application was developed to enable real-time infant cry classification and provide an accessible tool for parents and caregivers.

Keywords: Infant Cry Classification, Hierarchical Classification, Support Vector Machine, Random Forest, MFCC, SMOTE, Audio Signal Processing.

Résumé

L'analyse des pleurs des nourrissons constitue une application prometteuse de l'intelligence artificielle et du traitement du signal audio, les pleurs représentant le principal moyen de communication durant les premiers mois de la vie. Ce travail vise à développer un système intelligent capable de classifier automatiquement les pleurs des nourrissons selon le besoin exprimé.

L'étude s'appuie sur la base de données Donate-a-Cry, qui comprend cinq catégories de pleurs : la faim, l'inconfort, les douleurs abdominales, le rot et la fatigue. Cette base de données reflète les conditions réelles dans lesquelles les pleurs des nourrissons sont observés, la faim étant l'une des causes les plus fréquentes de pleurs au cours de la petite enfance. Il en résulte naturellement une représentation plus importante de cette catégorie par rapport aux autres. Cette caractéristique engendre des défis de classification nécessitant des modèles capables de maintenir des performances équilibrées pour l'ensemble des types de pleurs.

Pour relever ces défis, un cadre de classification hiérarchique (HCF) combinant les Machines à Vecteurs de Support (SVM) et les Forêts Aléatoires (RF) est proposé, en s'appuyant sur l'extraction de caractéristiques acoustiques, l'augmentation des données et le suréchantillonnage par SMOTE afin d'améliorer la capacité de généralisation du modèle. En outre, une application web développée avec **Streamlit** a été réalisée pour permettre la classification des pleurs en temps réel et fournir un outil pratique aux parents et aux aidants.

Mots-clés : Classification des pleurs des nourrissons, Classification hiérarchique, Machine à Vecteurs de Support, Forêt Aléatoire, MFCC, SMOTE, Traitement du signal audio.

ملخص

يُعد تحليل بكاء الرضع من التطبيقات الواعدة في مجال الذكاء الاصطناعي ومعالجة الإشارات الصوتية، نظراً لكون البكاء وسيلة التواصل الأساسية لدى الرضيع خلال الأشهر الأولى من حياته. ويهدف هذا العمل إلى تطوير نظام ذكي قادر على التصنيف الآلي لبكاء الرضع وفق الحاجة التي يعبر عنها الرضيع.

اعتمدت الدراسة على قاعدة بيانات Donate-a-Cry Corpus التي تضم خمس فئات من البكاء: الجوع، وعدم الارتياح، وآلام البطن، والتجشؤ، والتعب. وتعكس هذه القاعدة الظروف الواقعية التي يُجمع فيها بكاء الرضع، حيث يُعد الجوع من أكثر الأسباب شيوعاً للبكاء خلال مرحلة الرضاعة، مما يؤدي بطبيعة الحال إلى تمثيل أكبر لهذه الفئة مقارنة ببقية الفئات. ونتيجة لذلك، تبرز تحديات تصنيفية تتطلب نماذج قادرة على تحقيق أداء متوازن عبر جميع أنواع البكاء. ولمعالجة هذه التحديات، تم اقتراح إطار تصنيف هرمي (HCF) يجمع بين آلات ناقلات الدعم (SVM) والغابات العشوائية (RF) مع توظيف تقنيات استخراج الخصائص الصوتية وتعزيز البيانات وتقنية SMOTE لتحسين قدرة النموذج على التعميم. كما تم تطوير تطبيق ويب باستخدام Streamlit يتيح التصنيف الآلي لبكاء الرضع وعرض النتائج بطريقة سهلة للمستخدم.

الكلمات المفتاحية:

تصنيف بكاء الرضع، التصنيف الهرمي، آلات ناقلات الدعم (SVM)، الغابات العشوائية (RF)، معاملات التردد الميلي (MFCC)، تقنية SMOTE، معالجة الإشارات الصوتية.

CONTENTS

Acknowledgements	I
Abstract	II
List of Figures	VIII
List of Tables	IX
List of Abbreviations	X
General Introduction	1
1 Audio Signal Processing	5
1.1 Introduction	5
1.2 Introduction to Sound and Audio Signals	5
1.2.1 Definition of Sound	5
1.2.2 Properties of Audio Signals	6
1.3 Digital Audio Fundamentals	6
1.3.1 Sampling	7
1.3.2 Quantisation	7
1.4 Audio Preprocessing	7
1.4.1 Noise Reduction	8
1.4.2 Normalisation	8
1.4.3 Segmentation	8
1.5 Feature Extraction Techniques	9
1.5.1 Zero Crossing Rate (ZCR)	9
1.5.2 Chroma Features	9
1.5.3 MFCCs and Related Features	9

1.5.4	Spectrogram and Mel Spectrogram	10
1.5.5	Additional Spectral and Temporal Descriptors	11
1.5.6	Feature Selection	12
1.6	Conclusion	12
2	Artificial Intelligence for Infant Cry Classification	13
2.1	Introduction	13
2.2	Introduction to AI in Healthcare Audio Analysis	13
2.2.1	Definition of Artificial Intelligence	13
2.2.2	AI Applications in Medical Audio and Neonatal Care	14
2.2.3	Why AI for Infant Cry Classification?	15
2.3	Machine Learning for Cry Analysis	16
2.3.1	Supervised Learning in Cry Classification	16
2.3.2	Classification Metrics	16
2.3.3	Class Imbalance and Solutions	18
2.3.4	Classical ML Algorithms	18
2.4	Neural Networks for Cry Recognition	19
2.4.1	The Artificial Neuron	19
2.4.2	Multi-Layer Perceptron (MLP)	20
2.5	Deep Learning Models in Cry Classification	21
2.5.1	Convolutional Neural Networks (CNN)	21
2.5.2	Long Short-Term Memory (LSTM)	22
2.6	Related Work	23
2.7	Conclusion	24
3	Proposed Methodology	25
3.1	Introduction	25
3.2	Dataset Description	25
3.2.1	The Donate-a-Cry Corpus	25
3.2.2	Class Distribution and Imbalance Analysis	26
3.3	Audio Preprocessing	26
3.3.1	Audio Loading and Resampling	26
3.3.2	Data Augmentation	27
3.3.3	Feature Scaling Using StandardScaler	27
3.3.4	Class Balancing Using SMOTE	28
3.3.5	Train-Test Split Strategy	28
3.4	Feature Extraction	28
3.4.1	Mel-Frequency Cepstral Coefficients (MFCCs)	28
3.4.2	Chroma Features	29
3.4.3	Zero Crossing Rate (ZCR)	29

3.4.4	Mel-Spectrogram	29
3.4.5	Supplementary Acoustic Descriptors	29
3.4.6	Feature Vector Composition	30
3.5	Proposed Hierarchical Classification System	30
3.5.1	Motivation for Hierarchical Classification	30
3.5.2	Stage 1 — SVM Binary Classifier	31
3.5.3	Stage 2 — Random Forest Multi-class Classifier	31
3.5.4	Hierarchical Decision Pipeline	32
3.6	Proposed Web-Based System	32
3.6.1	Web Application Overview	32
3.6.2	System Workflow	32
3.6.3	System Architecture	33
3.7	Conclusion	35
4	Implementation, Experimental Results and Discussion	36
4.1	Introduction	36
4.2	Software Tools and Libraries	36
4.3	Models Evaluation	38
4.3.1	SVM Classifier (5-Class)	38
4.3.2	Random Forest Classifier (5-Class)	40
4.4	Hierarchical Model Evaluation	41
4.4.1	Stage 1 Evaluation — SVM Binary Classifier	41
4.4.2	Stage 2 Evaluation — Random Forest Multi-class Classifier	42
4.4.3	Full Hierarchical Pipeline Evaluation	43
4.4.4	Error Analysis	45
4.5	Comparative Analysis	45
4.5.1	Comparison Between Models	45
4.5.2	Discussion of Results	46
4.6	Web Application Implementation	47
4.6.1	Application Overview	47
4.6.2	User Interface Design	47
4.6.3	Backend Implementation	50
4.7	Limitations and Future Work	50
4.7.1	Current Limitations	50
4.7.2	Future Improvements	50
4.8	Conclusion	51
	General Conclusion	52
	Bibliography	54

LIST OF FIGURES

1.1	Analog-to-Digital conversion.	7
1.2	Audio preprocessing pipeline for infant cry classification.	8
1.3	The MFCC extraction pipeline.	10
1.4	Simulated spectrograms comparing a hungry cry and a pain cry.	11
1.5	The Mel-scale filterbank.	11
2.1	Hierarchical Relationship Between AI, Machine Learning, and Deep Learning.	14
2.2	Standard pipeline for AI-based infant cry classification.	16
2.3	Multi-Layer Perceptron (MLP) Architecture for Cry Feature Classification.	20
2.4	CNN architecture applied to Mel spectrograms for infant cry classification.	21
2.5	LSTM cell architecture.	22
3.1	Class distribution of the Donate-a-Cry Corpus.	26
3.2	Hierarchical Classification Framework (HCF).	32
3.3	Full system pipeline flowchart	34
4.1	Software tools and libraries used in this project.	37
4.2	Confusion matrix: SVM (5-class).	39
4.3	Confusion matrix: RF (5-class).	41
4.4	Confusion matrix: Stage 1 SVM binary classifier (hungry vs. not_hungry).	42
4.5	Confusion matrix: Stage 2 Random Forest (4-class).	43
4.6	Confusion matrix: Hierarchical SVM + RF (combined view).	44
4.7	Model comparison bar charts	46
4.8	Streamlit application homepage.	47
4.9	Audio upload interface and waveform visualization.	48
4.10	Prediction result panel.	49
4.11	Probability distribution chart and recommended caregiver actions.	49

LIST OF TABLES

2.1	Standard Evaluation Metrics in Infant Cry Classification	17
2.2	Comparative overview of infant cry classification studies.	23
3.1	Original class distribution of the Donate Cry Corpus dataset.	26
3.2	Dataset size before and after augmentation.	27
3.3	Composition of the dimensional feature vector.	30
4.1	Classification report: SVM (5-class).	39
4.2	Classification report: RF (5-class).	40
4.3	Classification report: Stage 1 SVM binary classifier (hungry vs. not_hungry).	41
4.4	Classification report: Stage 2 Random Forest multi-class classifier (4-class).	42
4.5	Classification report: Full Hierarchical Pipeline (SVM + RF).	44
4.6	Comparative performance summary: all models.	45

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CEC	Constant Error Carousel
DL	Deep Learning
DNN	Deep Neural Network
F0	Fundamental Frequency
FN	False Negative
FP	False Positive
GCN	Graph Convolutional Network
GFCC	Gammatone Frequency Cepstral Coefficient
HCF	Hierarchical Classification Framework
LSTM	Long Short-Term Memory
MFCC	Mel-Frequency Cepstral Coefficient
mHealth	Mobile Health
ML	Machine Learning
MLP	Multi-Layer Perceptron
NICU	Neonatal Intensive Care Unit
RBF	Radial Basis Function
RDS	Respiratory Distress Syndrome
ReLU	Rectified Linear Unit
RF	Random Forest
RMS	Root Mean Square
RNN	Recurrent Neural Network
SMOTE	Synthetic Minority Over-sampling Technique
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
WAV	Waveform Audio File Format
XGBoost	eXtreme Gradient Boosting
ZCR	Zero Crossing Rate

GENERAL INTRODUCTION

Context and Motivation

The rapid advancement of Artificial Intelligence (AI) and digital signal processing has transformed the way acoustic information is analysed and interpreted. Modern intelligent systems are increasingly capable of extracting meaningful patterns from audio signals and applying them to a wide range of domains, including speech recognition, emotion detection, environmental sound classification, and human-computer interaction. Among these emerging applications, infant cry analysis has attracted considerable attention due to its potential to support caregivers in understanding infant needs through objective and automated methods.

During the first months of life, infants have a very limited capacity to communicate with their surroundings. Crying therefore becomes their primary and most important communication channel. Through crying, infants express a variety of needs and states such as hungry, fatigue, discomfort, digestive distress, or the need for attention. While experienced caregivers may sometimes infer the underlying reason for a cry, distinguishing between different cry categories remains challenging because many vocal patterns exhibit similar acoustic characteristics.

Recent developments in machine learning and audio signal processing have opened new possibilities for the automatic interpretation of infant cries. By analysing acoustic features extracted from cry recordings, intelligent systems can learn discriminative patterns associated with different infant needs and provide valuable assistance to parents and caregivers. Such systems have the potential to improve responsiveness to infant needs while reducing uncertainty during daily childcare activities.

Problem Statement

Despite significant progress in audio classification, automatic infant cry classification remains a challenging task. One of the main difficulties lies in the complex and highly variable nature of cry signals, which are influenced by numerous factors including the infant's age, emotional state, recording conditions, and environmental noise.

In this work, the problem is addressed using the Donate-a-Cry Corpus, a publicly available dataset containing recordings belonging to five cry categories: hungry, discomfort, belly pain, burping, and tiredness. Although widely used in research, this dataset presents two major challenges.

The first challenge is the highly imbalanced distribution of cry categories. In the Donate-a-Cry Corpus, hungry cries account for approximately 83.7% of all available recordings, while some other categories are represented by only a limited number of samples. This distribution is not merely a characteristic of the dataset itself; rather, it reflects real-world infant behaviour, where hungry is among the most frequent causes of crying during early infancy. Consequently, datasets collected from natural caregiving environments tend to contain a substantially larger proportion of hungry cries than other cry types.

The second challenge concerns the relatively small size of the dataset. With fewer than five hundred original recordings distributed across five classes, the amount of available training data remains limited. This constraint restricts the applicability of data-hungry deep learning approaches and motivates the use of carefully designed machine learning techniques combined with data augmentation and synthetic oversampling strategies.

These challenges highlight the need for a robust classification framework capable of handling class imbalance while maintaining strong predictive performance across all cry categories.

Objectives of the Thesis

The main objective of this thesis is to develop an intelligent system capable of automatically classifying infant cries according to the underlying need expressed by the infant. To achieve this goal, several specific objectives were defined:

- To study the theoretical foundations of audio signal processing and machine learning relevant to infant cry classification.
- To extract meaningful acoustic representations from infant cry recordings using MFCCs, temporal derivatives, chroma descriptors, and Zero Crossing Rate features.

- To construct a robust 378-dimensional feature vector capable of representing the acoustic characteristics of cry signals.
- To address the class imbalance problem through targeted data augmentation and SMOTE-based synthetic oversampling.
- To design and evaluate a Hierarchical Classification Framework (HCF) combining Support Vector Machines and Random Forest classifiers.
- To compare the proposed framework against conventional classification approaches.
- To develop a user-friendly web application for real-time infant cry classification and result visualization.

Methodology Overview

The methodology adopted in this work follows a complete machine learning pipeline. First, audio recordings undergo preprocessing operations designed to improve signal quality and ensure consistency across samples. Next, relevant acoustic features are extracted and transformed into numerical representations suitable for machine learning algorithms.

To mitigate dataset imbalance, targeted data augmentation and SMOTE oversampling techniques are applied. Several classification models are then trained and evaluated, including conventional classifiers and the proposed hierarchical architecture. Finally, the best-performing model is integrated into a web application developed using `Streamlit`, enabling practical real-time usage.

Thesis Structure

This thesis is organized into four chapters:

- **Chapter 1 – Audio Signal Processing:** Introduces the fundamental concepts of sound, digital audio representation, preprocessing techniques, and acoustic feature extraction methods used throughout this work.
- **Chapter 2 – Artificial Intelligence for Infant Cry Classification:** Presents the theoretical foundations of machine learning and audio classification, together with a review of existing research related to infant cry analysis.
- **Chapter 3 – Proposed Methodology:** Describes the Donate-a-Cry dataset, preprocessing pipeline, feature extraction process, class balancing strategies, proposed Hierarchical Classification Framework, and web application design.
- **Chapter 4 – Implementation, Results and Discussion:** Details the implementation process, experimental setup, performance evaluation,

comparative analysis of the tested models, discussion of findings, limitations, and future perspectives.

Through these chapters, this work aims to demonstrate how carefully designed machine learning techniques can contribute to the automatic interpretation of infant cries and provide practical assistance for understanding infant needs.

CHAPTER 1

AUDIO SIGNAL PROCESSING

1.1 Introduction

This chapter presents the fundamental principles of audio signal processing in the context of infant cry analysis. Infant crying is a complex acoustic signal that carries valuable information about a baby's physiological and emotional state. Understanding how this signal is generated, represented, and processed is essential for developing reliable automated classification systems.

1.2 Introduction to Sound and Audio Signals

1.2.1 Definition of Sound

Sound is a vibration that travels through the air as a wave of pressure changes. When an object vibrates such as a baby's vocal cords it pushes and pulls the surrounding air molecules, creating a series of compressions and expansions that spread outward in all directions. When these pressure changes reach a microphone or the human ear, they are detected and interpreted as sound [1].

Infant cries are a unique type of sound signal that carry rich acoustic information about the infant's physical and emotional state, such as hungry, pain, or discomfort. As discussed by Ji et al [2], infant cry signals exhibit distinctive characteristics that differentiate them from adult speech, requiring specialized signal processing and classification techniques.

1.2.2 Properties of Audio Signals

Every audio signal (including a baby’s cry) has a set of measurable properties that describe its nature. Understanding these properties is the foundation of any audio analysis system:

- **Amplitude (Loudness):** Amplitude reflects how loud a sound is. Acoustic measurements showing that “Intense Cry” recordings (which include pain and colic cries) have a mean intensity of approximately 80 dB, compared to approximately 73 dB for “Regular Cry” recordings (which include hungry and fussy cries). Intensity in decibels plays a critically important role in how the mammalian ear perceives and experiences sound, and likely holds the key to why parents perceive some types of infant cry as more aversive than others. [3]
- **Fundamental Frequency (Pitch):** Fundamental Frequency (F_0), also known as pitch, represents the primary vibration rate of the vocal folds and is perceived as the pitch of the infant’s cry. It is considered an important acoustic feature in infant cry analysis, as it carries physiological information and has been widely used to differentiate between normal and pathological cries. [4]
- **Harmonics (Tone Colour):** In addition to the main frequency, a cry signal contains many other related frequencies called harmonics. Together, they give the cry its unique "color" or texture. Different types of cries have different harmonic patterns, which are clearly visible in their spectrograms. [2]
- **Temporal Characteristics:** Infant crying signals contain informative temporal characteristics that can be captured using time-domain features such as zero-crossing rate (ZCR) and root mean square (RMS). These features reflect variations in the signal over time and contribute to distinguishing between different types of infant cries. [5]

1.3 Digital Audio Fundamentals

Before a computer can analyze a baby’s cry, the continuous sound wave captured by a microphone must be converted into a sequence of numbers. Digital processing of analog signals consists of three main stages: analog-to-digital (A/D) conversion, digital processing, and digital-to-analog (D/A) conversion. The A/D conversion itself is carried out in two steps: uniform sampling and quantization [6]. Figure 1.1 illustrates how this works..

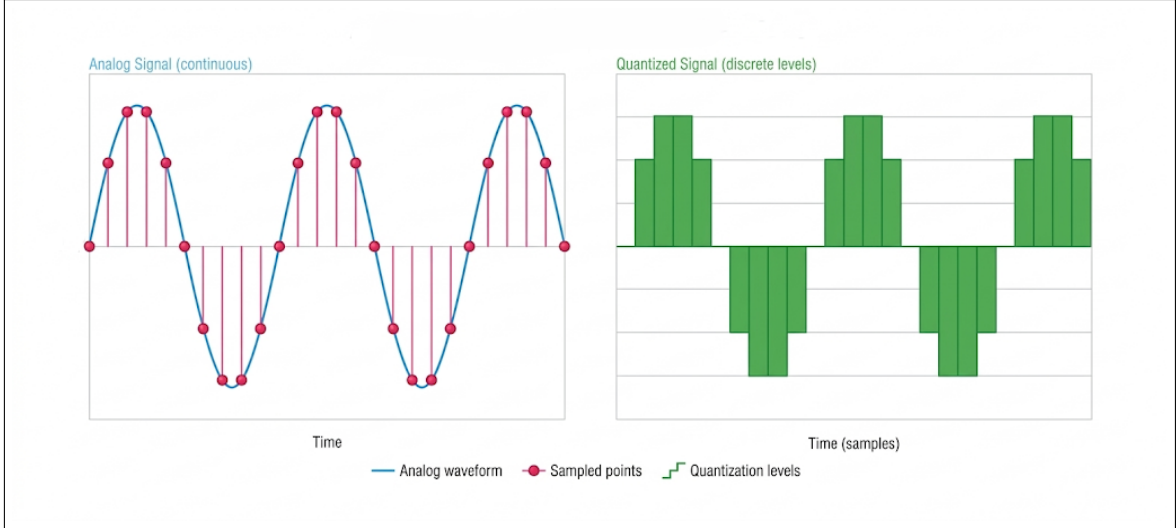


Figure 1.1: Analog-to-Digital conversion.

1.3.1 Sampling

Uniform sampling refers to the process by which an analog signal is mapped into a discrete-time signal by recording its amplitude at equally spaced time intervals, converting the independent variable (time) from continuous to discrete [6]. The rate at which these measurements are taken is called the sampling rate, measured in Hz. The higher the sampling rate, the more detail is captured about the original signal. For infant cry analysis, a sampling rate of 16,000 measurements per second (16 kHz) is the standard used in most research datasets, including the Donate-a-Cry corpus [7]. This rate is high enough to capture all the acoustic details of a baby’s cry while keeping file sizes manageable.

1.3.2 Quantisation

Each sampled amplitude value must be stored as a whole number. Quantization is the process of rounding each measurement to the nearest available representation level, converting the dependent variable from continuous to discrete [6]. Standard digital audio uses 16-bit depth, which provides 65,536 levels. The tiny rounding errors introduced by this process are generally modelled as additive random noise and are negligible in practice.

1.4 Audio Preprocessing

Raw recordings collected from homes, hospitals, or online platforms are rarely clean. They often contain background noise, differences in recording volume, and long silent gaps. Before extracting features, a preprocessing pipeline is applied to standardize

all recordings and remove irrelevant content. Figure 1.2 shows the full preprocessing pipeline.

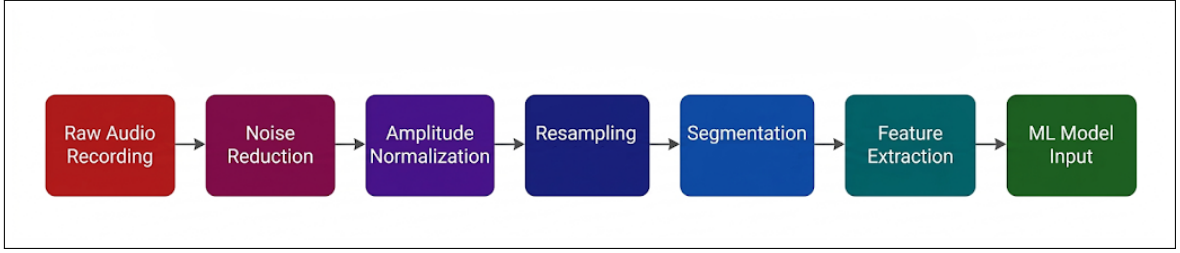


Figure 1.2: Audio preprocessing pipeline for infant cry classification.

1.4.1 Noise Reduction

Environmental sounds such as air conditioning, background voices, or handling noise can contaminate the recording and mislead the classifier. The most common family of speech enhancement algorithms (spectral subtractive algorithms) is based on the principle that, since noise is additive, one can estimate the noise spectrum when the cry is not present and subtract it from the noisy signal [8]. Among the widely used libraries for audio denoising, the `noisereduce` library is used, which relies on a method called spectral gating: it computes a spectrogram of the signal, estimates a noise threshold for each frequency band, and uses that threshold to compute a mask that gates frequency components whose energy falls below the estimated noise level [9]. Care must be taken not to over-apply noise reduction, as being too aggressive can distort the harmonic structure of the cry itself which is precisely the information needed for classification.

1.4.2 Normalisation

Audio normalization is a preprocessing technique used to rescale audio signals to a common reference level in order to reduce amplitude variability between recordings. This helps ensure that classification models focus on relevant acoustic features rather than differences in recording gain or distance [10].

1.4.3 Segmentation

Audio signals are typically non-stationary and vary over time, which makes direct processing of long recordings impractical. Therefore, signals are commonly segmented into short, overlapping frames in order to extract more stable and meaningful features for further processing [11].

1.5 Feature Extraction Techniques

A raw audio signal is a high-dimensional sequence of amplitude values over time. Although it contains complete information about the sound, it is not directly suitable for machine learning models due to its size, noise, and lack of explicit structure. Therefore, feature extraction is used to transform the raw waveform into a compact and informative representation that highlights the most relevant acoustic patterns [2]. Feature extraction reduces the complexity of audio data while preserving important information related to time, frequency, and perception. Common representations include MFCCs, spectrograms, and Mel spectrograms. In addition, dynamic features such as delta and delta-delta coefficients are used to capture temporal variations, while other descriptors like chroma features and zero-crossing rate provide complementary information about harmonic content and signal behavior [12].

1.5.1 Zero Crossing Rate (ZCR)

Zero Crossing Rate (ZCR) is a simple time-domain feature that measures how frequently the audio signal changes its sign from positive to negative or vice versa within a short time frame. It is widely used as an indicator of the noisiness and frequency content of a signal [12]. Higher ZCR values are generally associated with noisy or high-frequency signals, while lower values correspond to smoother or more tonal sounds. In speech and cry analysis, ZCR can help differentiate between voiced and unvoiced segments, making it a useful feature for classification tasks [12].

1.5.2 Chroma Features

Chroma features represent the distribution of energy across the 12 pitch classes of the musical octave (C, C#, D, ..., B), regardless of the actual octave in which they occur. This means that all notes sharing the same pitch class are grouped together, providing an octave-invariant representation of harmonic content [12]. These features are particularly useful in music analysis tasks such as chord recognition, key detection, and harmonic comparison between audio signals. By focusing on pitch class energy rather than absolute frequency, chroma features capture the tonal structure of the audio in a compact form [12].

1.5.3 MFCCs and Related Features

Mel-Frequency Cepstral Coefficients (MFCCs) are among the most widely used audio features. They provide a compact representation of the spectral envelope of sound over short time frames and are designed to approximate human auditory perception by emphasizing lower frequencies more than higher ones [12]. MFCCs are typically

computed over short overlapping windows, producing a sequence of feature vectors over time. Each vector summarizes the spectral characteristics of that frame, forming a 2D representation when combined over time [12]. To capture temporal dynamics, additional derivatives are often computed:

- **Delta (Δ) coefficients:** represent the rate of change of MFCCs over time. [12]
- **Delta-Delta ($\Delta\Delta$) coefficients:** represent the acceleration of these changes. [12]

These dynamic features improve the ability of machine learning models to capture temporal patterns in audio signals [12]. Figure 1.3 shows The MFCC extraction pipeline.

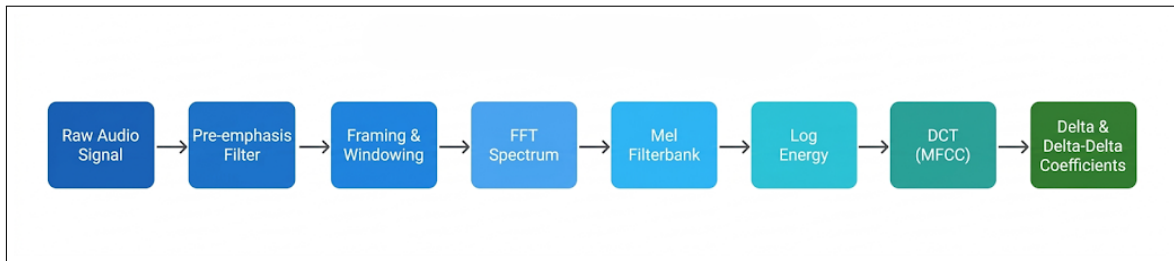


Figure 1.3: The MFCC extraction pipeline.

1.5.4 Spectrogram and Mel Spectrogram

A spectrogram represents the evolution of frequency content over time by applying the Short-Time Fourier Transform (STFT) to short overlapping segments of the audio signal [12]. It provides a time-frequency representation where intensity reflects energy at each frequency. Figure 1.4 presents simulated spectrograms comparing a hungry cry and a pain cry. A Mel spectrogram is a perceptually motivated variation of the standard spectrogram. It uses the Mel scale, which allocates higher resolution to lower frequencies and compresses higher frequencies, aligning more closely with human auditory perception [12]. This makes it a more efficient and meaningful representation for machine learning applications. Figure 1.5 presents The Mel-scale filterbank.

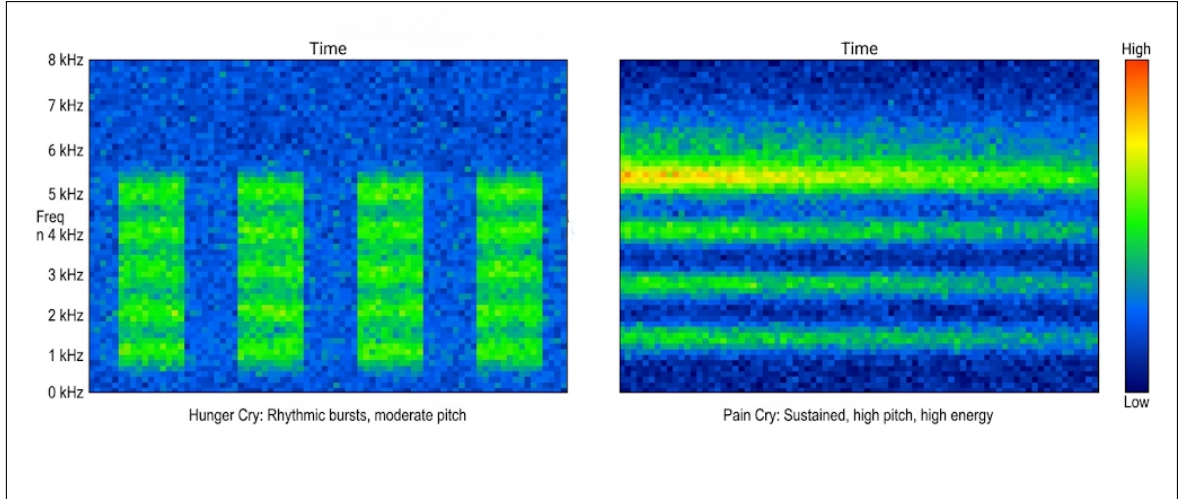


Figure 1.4: Simulated spectrograms comparing a hungry cry and a pain cry.

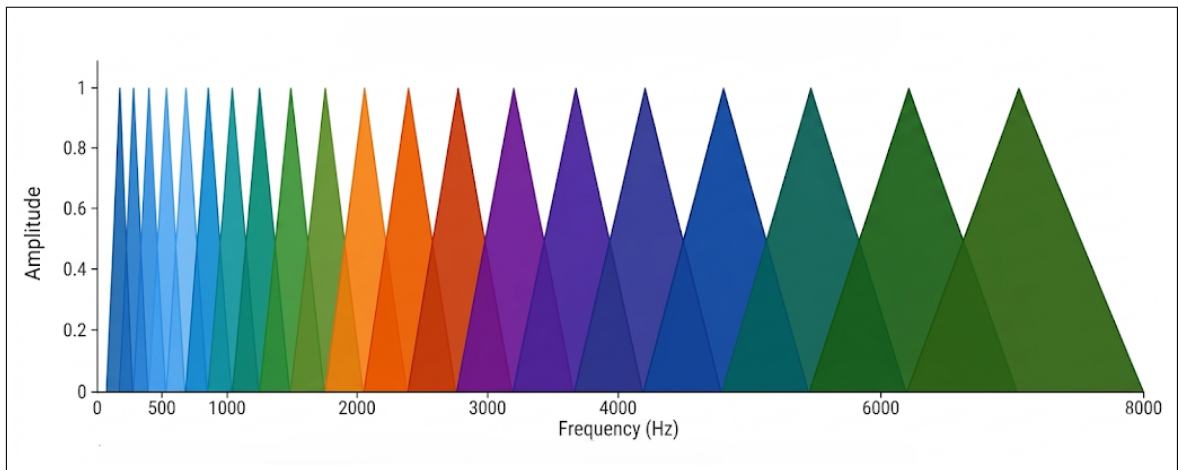


Figure 1.5: The Mel-scale filterbank.

1.5.5 Additional Spectral and Temporal Descriptors

To capture a comprehensive acoustic profile, several additional descriptors are utilized to measure loudness, timbral sharpness, and harmonic stability. [13]

- **Root Mean Square (RMS) Energy:** A time-domain descriptor that measures the total energy of an audio signal within short-term frames, serving as a reliable physical indicator of perceived loudness and amplitude envelope variations over time. [13]
- **Spectral Centroid:** A frequency-domain feature that identifies the center of gravity of the magnitude spectrum. In psychoacoustics, this descriptor is a measure of spectral shape, where higher values strongly correlate with the human perception of sound "brightness". [14]
- **Spectral Rolloff:** A measure of spectral shape that defines the frequency threshold below which 85% of the magnitude distribution is concentrated. [14]

- **Spectral Contrast:** Measures the relative difference in amplitude between spectral peaks and spectral valleys across distinct octave-based sub-bands, effectively reflecting the distribution of harmonic structures versus non-harmonic components or noise. [15]
- **Tonal Centroid (Tonnetz):** A high-level harmonic descriptor that projects 12-bin chroma features into a continuous 6-dimensional space to capture stable tonal relationships, enabling the effective detection of harmonic changes and chord boundaries in musical audio. [16]

1.5.6 Feature Selection

Feature selection is the process of choosing the most relevant features from the extracted feature set to improve the performance of machine learning models in cry classification tasks [17]. In infant cry analysis, not all extracted features contribute equally to distinguishing between different cry types. Some features may be redundant or noisy, which can reduce model performance. By selecting only the most informative features, the model becomes more efficient, reduces overfitting, and improves generalization to unseen data. In classical machine learning models such as Support Vector Machines and Random Forest, feature selection is often performed using statistical ranking or correlation-based methods to identify the most discriminative features. In contrast, deep learning models that use spectrogram or Mel spectrogram inputs can automatically learn the most relevant representations during training, reducing the need for explicit feature selection.

1.6 Conclusion

In this chapter, we explored the essential concepts of audio signal processing as applied to infant cry analysis. We examined how sound is produced and represented digitally, and how preprocessing techniques improve the quality and consistency of audio data. In addition, we discussed key acoustic properties that play a crucial role in distinguishing between different types of cries. Furthermore, we presented several feature extraction techniques, which convert raw audio signals into structured representations suitable for machine learning models. These features capture both spectral and temporal characteristics of infant cries, making them highly effective for classification tasks.

CHAPTER 2

ARTIFICIAL INTELLIGENCE FOR INFANT CRY CLASSIFICATION

2.1 Introduction

The automated classification of infant cries represents one of the most clinically impactful applications of Artificial Intelligence in infant care. Before a cry can be classified, it is essential to establish the theoretical foundations upon which all classification systems are built: the algorithms, architectures, and learning paradigms that transform raw acoustic signals into meaningful status outputs.

2.2 Introduction to AI in Healthcare Audio Analysis

2.2.1 Definition of Artificial Intelligence

Artificial Intelligence (AI) is the field of computer science dedicated to building systems capable of performing tasks that normally require human intelligence, such as learning from data, recognising patterns, reasoning, and making decisions. Russell and Norvig [18] define AI as the study of agents that perceive their environment and take actions, where the goal is to design agents that behave intelligently. Within this taxonomy :

- **Machine Learning (ML):** is the sub-field focused on algorithms that learn statistical patterns from data without being explicitly programmed. [18]
- **Deep Learning (DL):** is a further sub-field of ML that uses multi-layered neural networks to learn hierarchical representations directly from raw data. [19]

Figure 2.1 illustrates this hierarchical relationship.

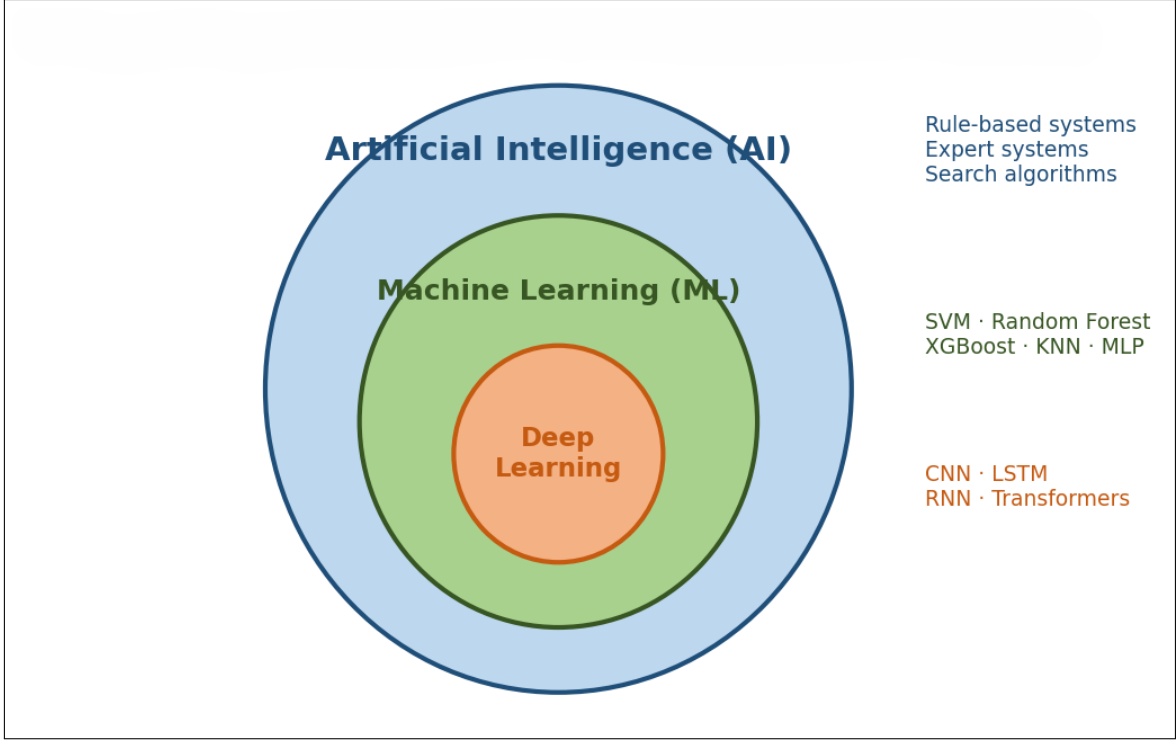


Figure 2.1: Hierarchical Relationship Between AI, Machine Learning, and Deep Learning.

In the context of infant cry analysis, Hashemi et al. [20] confirm that both classical ML and modern DL approaches have proven effective, noting that machine learning techniques can effectively categorise and classify infant needs and certain disorders across a wide range of datasets and feature representations.

2.2.2 AI Applications in Medical Audio and Neonatal Care

AI-driven biomedical audio analysis is reshaping the landscape of non-invasive diagnostic tools. Its applications span a wide spectrum, including speech-based disorder detection, respiratory sound classification, and infant cry analysis. The latter has emerged as a particularly vital area of research when applied to the neonatal period (defined as the first 28 days of life). This interval represents the highest-risk period for child mortality. According to global statistics from 2022, neonatal deaths accounted for approximately 2.3 million casualties, representing 47% of all under-five mortalities. The critical nature of this window is further underscored by its timeline: 75% of these neonatal deaths occur within the first week of life, with 25% to 45% occurring within the very first 24 hours [20]. These critical statistics highlight an urgent need for continuous, real-time monitoring solutions to bridge the diagnostic gap during these hazardous early hours. Within this domain, AI and deep learning frame infant cry analysis through two fundamental tasks:

- **Cry Detection:** A binary classification task dedicated to identifying the presence or absence of a crying event within a continuous audio signal, which

serves as the foundation for automated alert systems.

- **Cry Classification:** A more complex, multi-class classification task that aims to decode the acoustic signal to interpret its underlying cause. This involves differentiating between normal physiological needs (e.g., hungry, pain, diaper discomfort) and severe pathological conditions (e.g., birth asphyxia, hearing impairment, or neurological disorders) [20].

By leveraging these techniques, AI-driven cry analysis provides a non-invasive, continuous, and objective supplementary diagnostic tool. These intelligent systems are highly versatile and capable of being integrated into various care environments, ranging from Neonatal Intensive Care Units (NICUs) to assist clinical staff, to home monitoring systems and mobile health (mHealth) applications for parental support. Recent advancements validate the viability of these models; for instance, a deep learning architecture proposed by Ji et al. achieved an outstanding accuracy of 95.21% in detecting specific distress conditions, such as hungry and pain [20]. This demonstrates that infant vocalizations contain rich acoustic biomarkers that deep learning models can effectively decode, often surpassing the diagnostic capabilities of untrained human perception.

2.2.3 Why AI for Infant Cry Classification?

Infant cries carry important information about physiological needs and potential pathological conditions. However, accurate interpretation by caregivers can be challenging, particularly when atypical acoustic characteristics are present, motivating the development of automated cry analysis systems. [20]. AI-based systems can process large numbers of acoustic features simultaneously and operate continuously without fatigue, making them particularly suitable for real-time monitoring applications. Zayed et al. [21] demonstrated that pathological cries associated with conditions such as Respiratory Distress Syndrome (RDS) and sepsis exhibit subtle but discriminative acoustic biomarkers. By leveraging a deep neural network (DNN) with multi-domain feature fusion, their approach achieved up to 97.50% classification accuracy [21]. Furthermore, Goodfellow et al. [19] highlight that data-driven deep learning approaches are intrinsically better equipped than rule-based systems to handle the acoustic variability introduced by different recording environments, infant ages, and demographic backgrounds.

2.3 Machine Learning for Cry Analysis

2.3.1 Supervised Learning in Cry Classification

Supervised learning is a widely used paradigm in infant cry classification. In this approach, models are trained on labelled datasets consisting of acoustic feature vectors and their corresponding class labels. The objective is to learn to map input features to output labels by minimising a loss function over the training data [19]. The typical supervised learning pipeline, illustrated in Figure 2.2 :

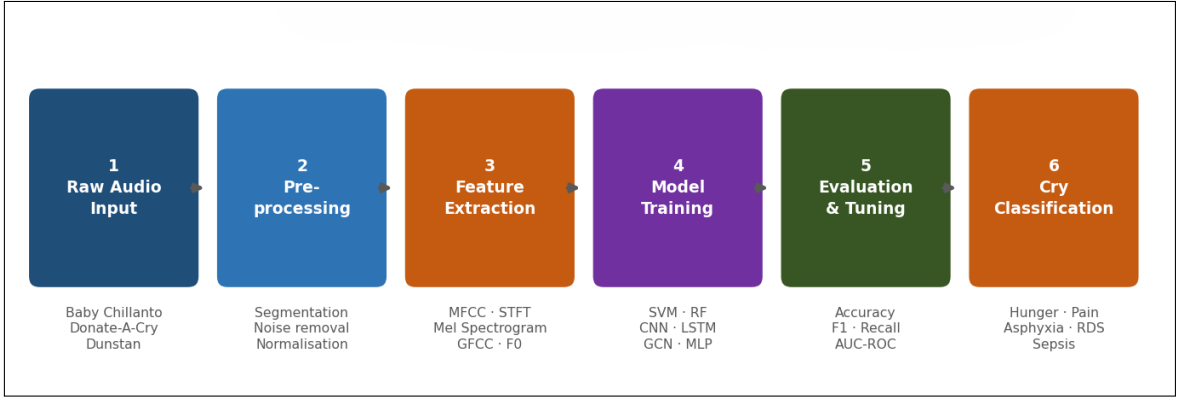


Figure 2.2: Standard pipeline for AI-based infant cry classification.

A major challenge in infant cry analysis is the scarcity of large, well-annotated datasets. Most available corpora, such as Baby Chillanto (2,268 samples), Donate A Cry Corpus (457 samples), and Dunstan Baby Language (315 samples), are relatively small and often exhibit class imbalance, demographic homogeneity, and recording environment bias. These limitations require the use of careful training strategies, including cross-validation, data augmentation, and regularisation, in order to reduce overfitting and improve generalisability.

2.3.2 Classification Metrics

Given the clinical nature of infant cry classification, the choice of evaluation metric is critical. The following standard metrics are defined by Zayed et al. [21] and employed throughout the literature reviewed in [20]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall (Sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

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

Where:

- **TP** (True Positives): correctly detected positive.
- **TN** (True Negatives): correctly detected negative.
- **FP** (False Positives): negative cases incorrectly classified as positive.
- **FN** (False Negatives): positive cases incorrectly classified as negative.

The evaluation metrics detailed above describe specific performance aspects of the model:

- **Accuracy** measures overall correctness, though it remains unreliable under class imbalance.
- **Precision** indicates the proportion of positive predictions that are truly positive, which minimises false alarms in automated monitoring.
- **Recall** represents the proportion of actual positive instances correctly identified. High recall is clinically critical, as missing a pathological cry may delay life-saving intervention.
- **F1-Score** is the harmonic mean balancing both metrics, making it preferred for imbalanced cry datasets.

Metric	Formula	Clinical Significance
Accuracy	$(\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$	General correctness
Precision	$\text{TP}/(\text{TP} + \text{FP})$	Minimise false alarms in NICU monitoring
Recall	$\text{TP}/(\text{TP} + \text{FN})$	Avoid missing pathological cries
F1-Score	$2 \times (\text{P} \times \text{R})/(\text{P} + \text{R})$	Balanced metric for imbalanced datasets

Table 2.1: Standard Evaluation Metrics in Infant Cry Classification

2.3.3 Class Imbalance and Solutions

Class imbalance is a pervasive challenge in infant cry datasets. In the Donate A Cry Corpus, hungry cries account for 382 of 457 total samples while burping cries represent only 8; in the SPLAN dataset, hungry cries (5,536 samples) vastly outnumber colic instances (225 samples). Models trained on such distributions bias predictions towards majority classes, yielding high accuracy while achieving near-zero recall on clinically critical minority classes. Three main strategies are employed:

Class-weighting : addresses class imbalance by assigning different weights to classes according to their sample frequencies, without modifying the original dataset. This approach aims to reduce misclassification of the minority class by assigning it a higher weight, while decreasing the influence of the majority class through lower weighting, thereby directly affecting the learning process during model training[22].

Synthetic Minority Over-sampling Technique (SMOTE) : generates synthetic samples by interpolating between existing minority-class feature vectors in the feature space, creating plausible new training examples rather than simply duplicating existing ones [23].

Data Augmentation : generates additional training examples through signal-level transformations including time-stretching, pitch-shifting, additive noise injection, and dynamic range compression [20].

2.3.4 Classical ML Algorithms

Support Vector Machine (SVM)

SVM is a supervised learning classifier that works by creating a hyperplane or multiple hyperplanes for separation, assigning training data labels based on the optimal hyperplane that best categorises each new sample [21]. For non-linearly separable data, kernel functions such as the Radial Basis Function (RBF) map inputs into higher-dimensional spaces where linear separation becomes feasible. SVMs are highly dependent on three key hyperparameters: the regularisation parameter C , which controls the trade-off between model complexity and training error; the kernel type, which determines the nonlinear mapping; and the kernel coefficient gamma, which controls the influence of each training example on the decision boundary [21].

One of SVM's greatest strengths in the clinical domain is its ability to create effective decision boundaries from small samples [21], critical given the restricted size of most infant cry datasets. In the diagnostic system of Zayed et al. [21], the SVM classifier achieved 94.47% accuracy on GFCC features and 94.79% on combined GFCC and Harmonic Ratio features after hyperparameter tuning (optimal: $C = 2$, gamma = 0.5). For the Baby Chillanto dataset, Badreldine et al. reported 98.5% accuracy on a 3-class asphyxia problem using Wavelet-based MFCC features [20].

Random Forest (RF)

Random Forest builds multiple decision trees and predicts via majority voting [21]. Its ensemble mechanism improves robustness and reduces overfitting. In [21], RF achieved 90.00% accuracy on GFCC features.

XGBoost

XGBoost (eXtreme Gradient Boosting) is a scalable, regularised gradient boosting framework that builds a sequential ensemble of weak learners (shallow decision trees) where each successive tree corrects the residual errors of its predecessors [24]. XGBoost introduces engineering optimisations including parallel tree construction, L1 and L2 regularisation terms, and built-in missing value handling, making it computationally efficient on tabular feature data [24]. In infant cry classification, XGBoost has been applied to MFCC feature vectors and used within stacking ensemble systems. Chang et al. proposed an Extreme Gradient Boosting-Assisted Grouped-Support-Vector Network for neonatal cry classification. [21]

2.4 Neural Networks for Cry Recognition**2.4.1 The Artificial Neuron**

The artificial neuron is the fundamental computational unit of all neural network architectures. Haykin [25] and Kumar [26] describe a neuron as a unit that computes a weighted linear combination of its inputs, adds a bias term, and passes the result through a non-linear activation function. Formally, for input vector $\mathbf{x}$, weight vector $\mathbf{w}$, and bias b , the neuron output is:

$$y = f(\mathbf{w}^\top \mathbf{x} + b)$$

where $f(\cdot)$ is the activation function [25]. The number of layers and neurons determines model complexity, the activation function determines how each neuron's output is calculated, and the learning rate controls the step size during gradient descent [21]. Common activation functions include the Rectified Linear Unit (ReLU),

$$f(z) = \max(0, z)$$

used throughout hidden layers; and SoftMax, applied at multi-class output layers to produce a normalised probability distribution over classes [21]. In the diagnostic DNN of Zayed et al. [21], ReLU is used in all hidden layers and SoftMax at the output layer for three-class pathology classification (Healthy, RDS, Sepsis).

Training proceeds through backpropagation combined with gradient-based optimisation [19]. A forward pass computes predictions; a categorical cross-entropy loss function quantifies the discrepancy between predictions and ground truth; and a backward pass propagates gradients to update weights in the loss-minimising direction [19].

2.4.2 Multi-Layer Perceptron (MLP)

A Multi-Layer Perceptron (MLP), also known as a feedforward neural network, is composed of an input layer, one or more hidden layers, and an output layer, where information flows strictly from input to output without feedback connections [27]. Each hidden layer applies a learned affine transformation followed by a non-linear activation function, enabling the network to model complex, non-linear decision boundaries that cannot be captured by linear classifiers. Figure 2.3 illustrates a typical MLP architecture for cry classification.

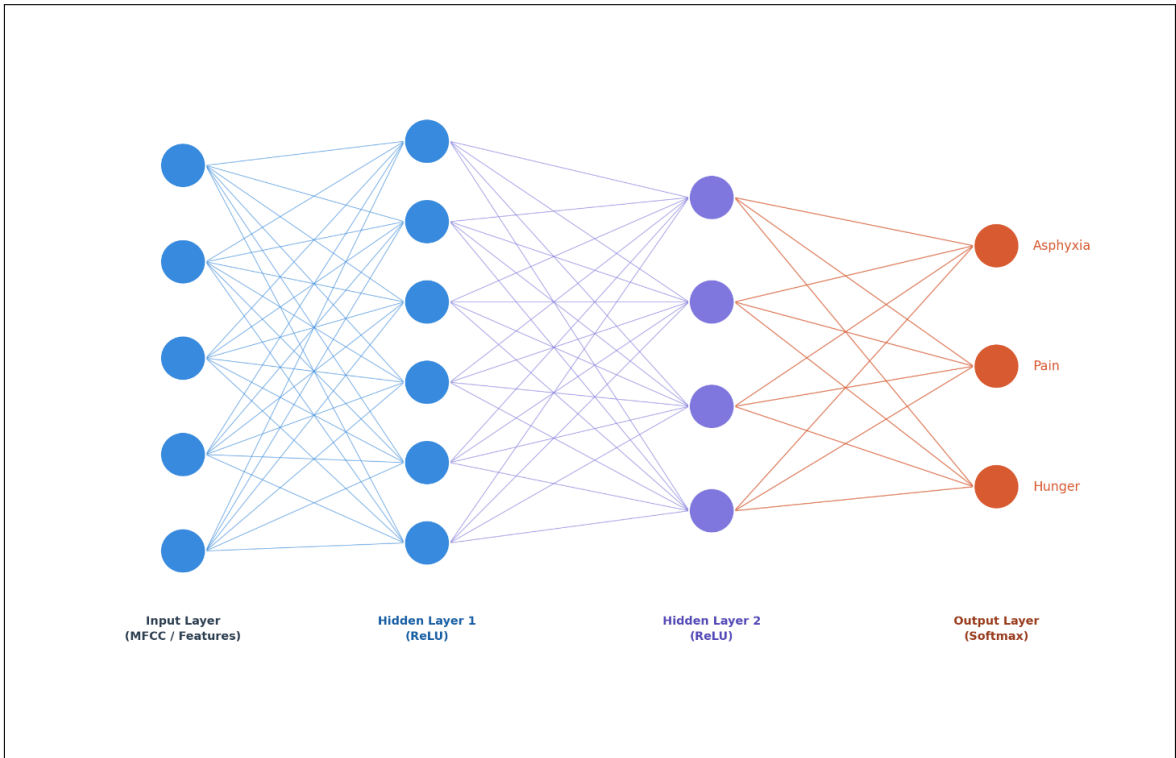


Figure 2.3: Multi-Layer Perceptron (MLP) Architecture for Cry Feature Classification.

In infant cry classification, the MLP takes a fixed-dimensional acoustic feature vector (typically a concatenation of MFCC coefficients and their delta derivatives) as input, and outputs a probability distribution over cry categories. Zayed et al. [21] employ a four-layer DNN with an input layer of 4096 spectrogram features (extracted via VGG16), two fully connected hidden layers of 1024 and 256 units with ReLU activations, and a 3-unit SoftMax output layer. This architecture achieved 93.00%

accuracy with spectrogram features alone, rising to 97.50% when GFCC and harmonic ratio features were fused through the learning process. Among the studies reviewed by Hashemi et al., Zabidi et al. applied MLP models to MFCC-based infant cry features for hypothyroidism detection, reporting accuracies around 88% with recall values reaching 100%.

2.5 Deep Learning Models in Cry Classification

2.5.1 Convolutional Neural Networks (CNN)

Using Mel spectrogram representations, Convolutional Neural Networks (CNNs) can effectively analyse audio signals by treating them as images and learning time-frequency patterns [20]. A CNN processes a spectrogram through convolutional layers, where learnable filters extract local spectro-temporal features, followed by pooling layers that reduce spatial resolution and introduce translation invariance [20]. Figure 2.4 illustrates a VGG-style CNN architecture applied to cry spectrograms.

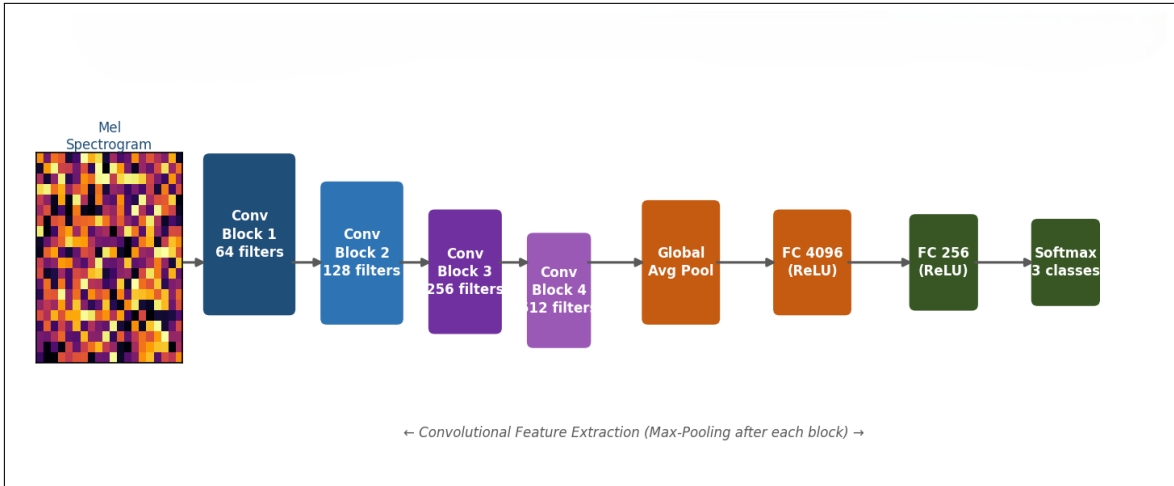


Figure 2.4: CNN architecture applied to Mel spectrograms for infant cry classification.

In the system proposed by Yara Zayed et al. [21], VGG16 pretrained on ImageNet was used as a transfer learning feature extractor for 224×224 RGB spectrogram images. The architecture includes convolutional layers with 64, 128, 256, and 512 filters followed by max-pooling operations, producing a 4096-dimensional feature vector used for classification. The spectrogram-based model achieved 92.05% accuracy, outperforming GFCC features (89.26%) and harmonic ratio features (59.70%) when used independently [21].

2.5.2 Long Short-Term Memory (LSTM)

While CNNs excel at learning spatial patterns from fixed-size inputs, the infant cry is a temporal signal whose diagnostic information is distributed across time. Recurrent Neural Networks (RNNs) maintain a hidden state updated at each time step as a function of both the current input and the previous hidden state. However, standard RNNs suffer from the vanishing gradient problem gradients diminish exponentially during backpropagation through long sequences, preventing learning of long-range dependencies [19]. The Long Short-Term Memory (LSTM) network, introduced by Hochreiter and Schmidhuber [28], addresses the vanishing gradient problem through memory cells and gating mechanisms. The original LSTM architecture includes an input gate, which controls what information is stored in the memory cell, and an output gate, which controls what information is exposed to the hidden state [28]. The cell state, maintained through the Constant Error Carousel (CEC), acts as a long-term memory capable of preserving information over long sequences [28]. Figure 2.5 illustrates the LSTM cell structure..

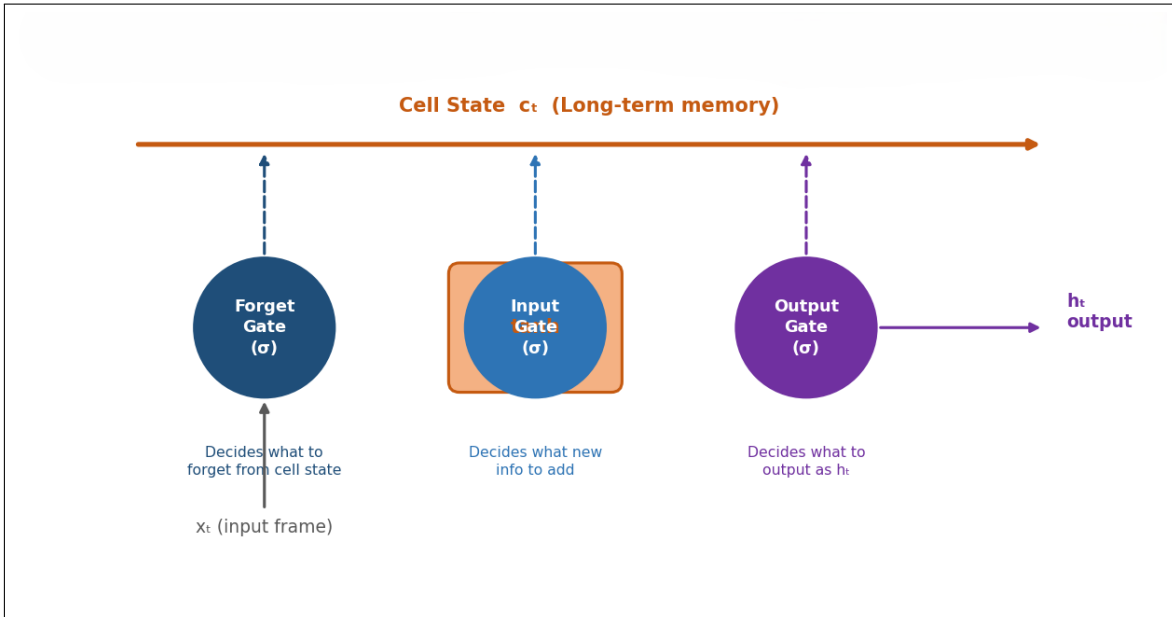


Figure 2.5: LSTM cell architecture.

Applied to infant cry classification, LSTM processes sequences of acoustic feature frames MFCC vectors computed at consecutive time windows ,modelling the temporal evolution of the cry signal [20]. Previous studies reviewed in [21]investigated deep learning approaches for infant cry classification using MFCC and cepstrum-based features. ANN, CNN, and LSTM architectures were evaluated for distinguishing healthy and pathological infant cries. In another study reviewed in [21], cepstrum-based representations were provided to DFFNN, LSTM, and CNN models, where CNN achieved the highest accuracy of 95.31% in binary healthy/pathological classification,

outperforming the other deep learning models [21]. These findings suggest the effectiveness of CNN-based representations for infant cry analysis.

A **hybrid CNN-LSTM** architecture leverages both CNN’s spatial feature extraction and LSTM’s temporal modelling capacity [20]. CNN layers extract local spectro-temporal features from spectrogram frames; the resulting feature sequence is then modelled by LSTM. Maghfira et al. applied this CNN-RNN architecture to the Dunstan Baby Language dataset and achieved 94.97% accuracy, demonstrating that joint spatial-temporal modelling captures both instantaneous acoustic characteristics and the temporal evolution of phonetic cry patterns.

2.6 Related Work

Paper	Dataset	Dataset Details	Features	Algorithm(s)	Performance
Ji et al. (2021) [29]	Baby Chillanto	2,267 samples, 5 classes: Asphyxia, Deaf, hungry, Normal, Pain.	Spectrogram (ResNet50)	GCN (semi-supervised) with TL-CNN feature extractor	94.39% (semi-sup., 5-class)
Dharwadkar et al. (2022) [30]	Baby Chillanto	2,274 samples, 5 classes: Pain, hungry, Asphyxia, Normal, Deaf.	MFCC + Pitch + Delta features	DNN (Feed-Forward Neural Network); RNN also tested	97.00% (DNN, 5-class)
Liang et al. (2022) [31]	Clinical (prop.)	1,607 samples, 2 classes: Healthy vs. Sick.	MFCC	CNN, ANN, LSTM	95.00% (CNN, 2-class)
Joshi et al. (2022) [32]	Clinical (prop.)	Private dataset, 4 classes: Sleep, hungry, Pain, Diaper change.	Mel Spectrogram	CNN (VGG16, YOLOv4) + Stacking Ensemble	93.70% (Ensemble, 4-class)
Ekinci & Küçükkülahlı (2023) [33]	Donate-a-Cry	457 samples, 5 classes: Hungry, Discomfort, Burping, Belly pain, Tired.	MFCC	k-NN, SVM, Random Forest, MLP	98.80% (k-NN, $\times 10$ aug.)
Younis et al. (2024) [34]	Donate-a-Cry		Mel Spectrogram	CNN, Multi-input CNN, ViT, ViT-SPT-LSA	98.33% (ViT-SPT-LSA, 5-class)
Hammoud et al. (2024) [5]	Donate-a-Cry		MFCC, ZCR, RMS, Mel Spectrogram, TSI	DT, RF, k-NN, SVM, XGBoost, NB, Bagging	96.39% (MFCC + RF, 5-class)
Rezaee et al. (2024) [35]	Donate-a-Cry		STFT	DeepSVM + INCA feature selection	98.34% (DeepSVM, 5-class)

Table 2.2: Comparative overview of infant cry classification studies.

The reviewed studies reflect a clear evolution in infant cry classification, both in terms of architectural complexity and dataset diversity. Early approaches relied on handcrafted acoustic features such as MFCC and pitch combined with relatively shallow models, whereas more recent works have progressively adopted deep learning architectures and graph-based methods. Notably, Ji et al. [29] demonstrated that modeling relational structure between cry segments through Graph Convolutional Networks can outperform standard CNNs even with significantly less labeled data, highlighting the importance of capturing long-range dependencies inherent to cry signals. A recurring limitation across these studies, however, is the heavy reliance

on small or heavily imbalanced datasets, particularly the Donate-a-Cry corpus, which comprises only 457 recordings with extreme class disproportion. The high accuracy figures reported on this corpus, consistently above 96%, should therefore be interpreted with caution, as they may partly reflect dataset-specific artifacts rather than genuine generalization capability. In contrast, studies conducted on clinical datasets [31] [32] tend to report more modest yet arguably more meaningful results, given the controlled acquisition conditions and clearer class definitions. Overall, while classification performance has steadily improved, a standardized evaluation benchmark remains absent from the field, making direct cross-study comparison methodologically limited.

2.7 Conclusion

This chapter has established the complete theoretical and methodological foundation for AI-based infant cry classification. Starting from first principles, it defined Artificial Intelligence, Machine Learning, and Deep Learning as a conceptual hierarchy, and motivated their application to neonatal cry analysis through the clinical urgency of early pathology detection and the inherent limitations of manual cry interpretation.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Introduction

This chapter presents the complete methodology adopted in this research for the automatic classification of infant cry sounds. The proposed system addresses the fundamental challenges posed by the Donate Cry Corpus dataset, namely severe class imbalance and limited data availability. The methodology is structured around a novel **Hierarchical Classification Framework (HCF)** that decomposes the five-class problem into two sequential classification stages, each specifically designed and optimized for the statistical characteristics of its target classes.

3.2 Dataset Description

3.2.1 The Donate-a-Cry Corpus

The dataset used in this study is the Donate Cry Corpus, a publicly available collection of infant cry recordings. The recordings are categorized into five classes, each representing a distinct physiological or emotional state of the infant: hungry (*hungry*), physical discomfort (*discomfort*), abdominal pain (*belly pain*), the need to burp (*burping*), and fatigue (*tired*). All recordings are in WAV format and were collected from real-world environments under varying acoustic conditions. Table 3.1 and Figure 3.1 summarizes the original class distribution of the dataset.

Class	Samples	%
hungry	382	83.7%
Discomfort	27	5.9%
Tired	24	5.3%
Belly Pain	16	3.5%
Burping	8	1.8%
Total	457	100.0%

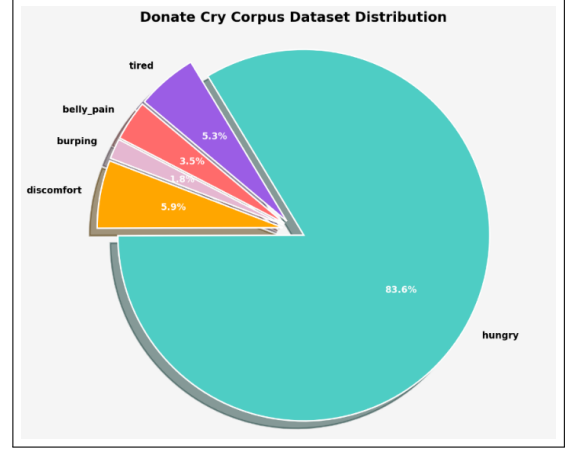


Table 3.1: Original class distribution of the Donate Cry Corpus dataset.

Figure 3.1: Class distribution of the Donate-a-Cry Corpus.

The dataset exhibits extreme class imbalance: the *hungry* class accounts for approximately 83.7% of all recordings, while the *burping* class contains only 8 samples. This severe skew would cause standard classifiers to be dominated by the majority class, motivating the hierarchical approach proposed in this work.

3.2.2 Class Distribution and Imbalance Analysis

A critical observation when examining the Donate-a-Cry Corpus is the pronounced class imbalance. The hungry class is substantially over-represented compared to the remaining four classes. Specifically, the dataset contains approximately 382 audio samples for the hungry class, while *belly pain*, *burping*, *discomfort*, and *tired* together contain 75 samples. This disparity arises naturally from the crowdsourcing collection method, as hungry is the most frequent and easily identifiable cause of infant crying.

This imbalance poses a significant challenge for standard machine learning classifiers, as they tend to develop a bias toward the majority class, resulting in poor performance on minority classes. To address this issue, the proposed methodology employs a combination of data augmentation on minority classes and Synthetic Minority Over-sampling Technique (SMOTE) to achieve a balanced training distribution.

3.3 Audio Preprocessing

3.3.1 Audio Loading and Resampling

All audio files were loaded using the `librosa` library at a uniform sampling rate of 22,050 Hz. Standardizing the sampling rate across all recordings ensures consistency and eliminates any frequency-scale discrepancies arising from different recording devices.

or conditions. The audio signals were stored as floating-point arrays for subsequent feature extraction.

3.3.2 Data Augmentation

Given the severely limited number of samples in the four minority classes (*belly pain*, *burping*, *discomfort*, and *tired*), data augmentation was applied exclusively to these classes to artificially expand their representation in the training set. For each original recording, five augmented variants were generated using the following transformations:

- **Time Stretching (rate = 0.9)**: slows down the audio by 10% without altering pitch, simulating slower cry utterances.
- **Time Stretching (rate = 1.1)**: speeds up the audio by 10% without altering pitch, simulating faster, more urgent cries.
- **Pitch Shifting (+2 semitones)**: raises the pitch by two semitones while preserving duration, simulating higher-pitched distress sounds
- **Pitch Shifting (−2 semitones)**: lowers the pitch by two semitones while preserving duration.
- **Additive Gaussian Noise ($\sigma = 0.005$)**: introduces low-level random noise to simulate real-world recording variability and background interference

The *hungry* class was excluded from augmentation as it already contains a sufficient number of samples. These transformations preserve the acoustic characteristics relevant to cry classification while introducing sufficient variability to reduce overfitting. After augmentation, the dataset expanded from 457 to 832 samples, as summarized in Table 3.2:

Class	Original	Augmented	Total
hungry	382	0	382
Discomfort	27	135	162
Tired	24	120	144
Belly Pain	16	80	96
Burping	8	40	48
Total	457	375	832

Table 3.2: Dataset size before and after augmentation.

3.3.3 Feature Scaling Using StandardScaler

Prior to classification, all extracted feature vectors were normalized using `scikit-learn`’s `StandardScaler`, which transforms each feature to have zero mean

and unit variance. Feature normalization is essential in machine learning pipelines particularly for distance-based and kernel-based classifiers such as SVM as features with large magnitudes can disproportionately dominate the learning process. By standardizing the feature space, the classifier treats all features equally and converges more efficiently during training. A separate `StandardScaler` instance was fitted independently for Stage 1 and Stage 2 of the hierarchical pipeline to avoid data leakage between the two classification stages.

3.3.4 Class Balancing Using SMOTE

Even after data augmentation, the training set still exhibited residual class imbalance. To further address this, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training partition only. SMOTE operates by synthesizing new samples in the feature space for the minority classes by interpolating between existing minority-class samples and their nearest neighbors. The number of nearest neighbors (k) was set to $k=2$ for Stage 2 due to the small sample count in some minority classes, with a fallback to $k=1$ when necessary, ensuring that SMOTE remains applicable even in very low-resource conditions. This approach generates synthetic, yet plausible, feature vectors that expand the minority class distributions without simple duplication.

3.3.5 Train-Test Split Strategy

The dataset was split into 85% training and 15% testing using stratified splitting to preserve class proportions. A fixed random seed (`seed=42`) was used throughout for full reproducibility.

3.4 Feature Extraction

Feature extraction transforms raw audio signals into compact numerical representations suitable for machine learning algorithms. In this work, a combination of spectral and temporal audio features was extracted from each recording. The selection of features was guided by their established effectiveness in speech and cry analysis in the literature.

3.4.1 Mel-Frequency Cepstral Coefficients (MFCCs)

Based on the feature extraction techniques introduced in Chapter 1, MFCCs were selected as the primary acoustic representation for the proposed model due to their effectiveness in capturing perceptually relevant spectral information from infant cry signals. For each audio recording, 40 MFCC coefficients were extracted

using short overlapping analysis frames. In order to incorporate temporal dynamics, first-order Delta coefficients (Δ) and second-order Delta-Delta coefficients ($\Delta\Delta$) were also computed. These additional features provide information about the temporal evolution of the cry signal and improve the model’s ability to distinguish between different cry patterns.

3.4.2 Chroma Features

Chroma features were included as complementary descriptors to the MFCC representation. A set of 12 chroma coefficients was extracted from each recording to capture the distribution of harmonic energy across pitch classes. Although chroma features are commonly used in music information retrieval, they can also provide useful harmonic characteristics in infant cry analysis, particularly when combined with spectral features such as MFCCs.

3.4.3 Zero Crossing Rate (ZCR)

The Zero Crossing Rate (ZCR) was used as an additional temporal feature to characterize variations in signal activity and frequency behavior. For each audio sample, the ZCR was computed over short frames and integrated with the other extracted features. This descriptor provides complementary information related to signal noisiness and temporal structure, which may assist the classifier in differentiating between cry conditions.

3.4.4 Mel-Spectrogram

In this work, a Mel-spectrogram representation was practically implemented using the `librosa.feature.melspectrogram` function to capture the time-varying power distribution of the cry signals across 40 distinct Mel bands ($n_mels = 40$). To better align the raw acoustic energy with human loudness perception and optimize feature scaling for the machine learning pipeline, the extracted power spectrogram was converted into a decibels scale via the `librosa.power_to_db` utility. By scaling the features relative to the peak intensity of each recording, this transformation effectively enhanced the model’s robustness against background noise and variation in recording volumes.

3.4.5 Supplementary Acoustic Descriptors

In addition to the primary representations, five supplementary acoustic descriptors were extracted to capture subtle variations in timbral quality, energy distribution, and harmonic structure:

- **RMS Energy:** Extracted to track the temporal envelope and intensity dynamics of the signal.
- **Spectral Centroid and Rolloff:** Included to capture the shifting spectral brightness and high-frequency noise components within the vocalizations.
- **Spectral Contrast:** Extracted across 7 sub-bands (yielding 7 coefficients per frame) to evaluate the relative sharpness of harmonic peaks relative to spectral valleys.
- **Tonal Centroid (Tonnetz):** Extracted as a 6-dimensional representation to capture embedded tonal stability and complex resonance patterns.

3.4.6 Feature Vector Composition

For each feature type, the mean and standard deviation were computed across all analysis frames to generate a fixed-length representation regardless of the sample duration:

Feature	Coefficients	Dimensions
MFCCs (static)	40	80
Delta (Δ)	40	80
Delta-Delta ($\Delta\Delta$)	40	80
Chroma	12	24
ZCR	1	2
Mel-spectrogram	40	80
RMS	1	2
Spectral Centroid	1	2
Spectral Rolloff	1	2
Spectral Contrast	7	14
Tonnetz	6	12
Total	—	378

Table 3.3: Composition of the dimensional feature vector.

3.5 Proposed Hierarchical Classification System

3.5.1 Motivation for Hierarchical Classification

A direct multi-class classification approach treating all five cry types simultaneously was found to be challenging due to the acoustic similarity between certain cry categories

(e.g., *discomfort* and *burping*) and the dominant presence of the *hungry* class. To overcome these limitations, a hierarchical classification architecture was designed, in which the problem is decomposed into two sequential sub-problems of reduced complexity.

The key motivating observation is that the hungry cry has distinct acoustic characteristics notably higher energy, regular rhythmic patterns, and a more pronounced fundamental frequency. That make it relatively separable from the remaining cry types. By first isolating hungry cries using a high-precision binary classifier, the second stage can focus exclusively on distinguishing the four acoustically closer categories (*belly pain*, *burping*, *discomfort*, *tired*), thereby improving overall classification accuracy.

3.5.2 Stage 1 — SVM Binary Classifier

The first stage of the hierarchical pipeline is a Support Vector Machine classifier configured for binary classification between the *hungry* and *not hungry* classes (where *not hungry* is an aggregate label encompassing *belly pain*, *burping*, *discomfort*, and *tired*). SVM was selected for this stage due to its well-established effectiveness in high-dimensional feature spaces and its ability to find maximum-margin decision boundaries, which is particularly beneficial when separating one class from a combined aggregate.

The SVM was configured with a Radial Basis Function (RBF) kernel, $C=10$, and $\text{gamma}=\text{'auto'}$. The RBF kernel allows the classifier to model non-linear decision boundaries in the 378-dimensional feature space. The training was performed on the SMOTE-balanced feature vectors scaled by the Stage 1 StandardScaler.

3.5.3 Stage 2 — Random Forest Multi-class Classifier

The second stage employs a Random Forest classifier to distinguish among the four remaining cry categories: *belly pain*, *burping*, *discomfort*, and *tired*. Random Forest was chosen for this stage due to its robustness to overfitting, its ability to handle small datasets effectively through ensemble averaging, and its native support for multi-class classification without requiring one-vs-rest decomposition.

The Random Forest was configured with 1000 decision trees ($\text{n_estimators}=1000$), a maximum depth of 20 ($\text{max_depth}=20$), a minimum of 1 sample per leaf ($\text{min_samples_leaf}=1$), and $\text{class_weight}=\text{'balanced'}$ to account for residual class imbalance. Training was performed on the SMOTE-balanced Stage 2 training set.

3.5.4 Hierarchical Decision Pipeline

During inference, an input audio recording is first preprocessed and its 378-dimensional feature vector is extracted. The vector is then passed to the Stage 1 SVM binary classifier. If the prediction is 'hungry', the system immediately outputs 'hungry' as the final classification result. Otherwise, the same feature vector is passed to the Stage 2 Random Forest classifier, which outputs one of the four remaining class labels. This sequential decision process ensures that the high-confidence hungry detection at Stage 1 does not interfere with the more nuanced four-class discrimination at Stage 2. Figure 3.2 illustrates the complete hierarchical classification architecture, showing the two-stage decision pipeline and all possible classification paths.

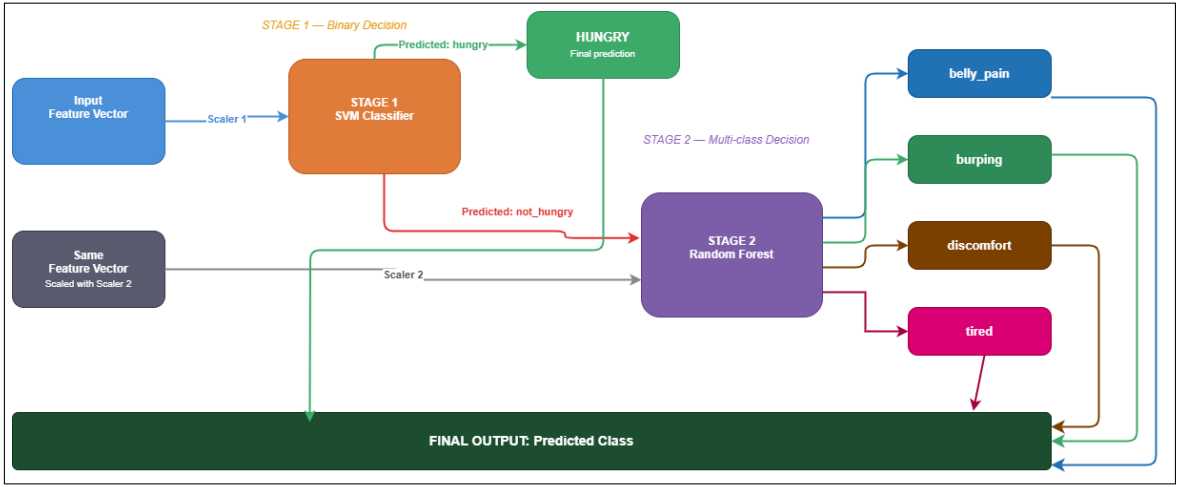


Figure 3.2: Hierarchical Classification Framework (HCF).

3.6 Proposed Web-Based System

3.6.1 Web Application Overview

To demonstrate the practical applicability of the proposed classification system, a web-based user interface was developed using Streamlit, a Python-based open-source framework for building interactive data science applications. The application allows end-users including parents, caregivers, and healthcare professionals to upload a WAV audio recording of an infant cry and receive an instant classification result along with a confidence score and probability distribution across all five cry categories.

3.6.2 System Workflow

The end-to-end workflow of the web application proceeds through the following stages:

1. **Audio Upload:** the user uploads a WAV file through the drag-and-drop or browse-file interface.
2. **Audio Preview:** the application displays an interactive audio player and renders the waveform of the uploaded recording.
3. **Preprocessing:** : the audio is loaded and resampled to 22,050 Hz using `librosa`.
4. **Feature Extraction:** the 378-dimensional feature vector is computed using the same pipeline as during training.
5. **Classification:** the feature vector is passed through the two-stage hierarchical pipeline (Stage 1 SVM $\rightarrow$ Stage 2 RF if applicable).
6. **Result Display:** : the predicted cry type is displayed with a confidence score, stage indicator, probability distribution bar chart, and recommended caregiver actions.

3.6.3 System Architecture

The application is entirely self-contained within a single Streamlit Python script. The pre-trained SVM and Random Forest models, along with their associated `StandardScaler` objects, are serialized using Python's `pickle` library and loaded at application startup. The frontend and backend logic are tightly integrated within the Streamlit framework, which handles HTTP request routing, session state management, and reactive rendering of UI components. The trained model artifacts are loaded from local disk, ensuring low-latency inference without the need for external API calls. Figure 3.3 presents the three-layer architecture of the web application, illustrating the interaction between the user interface layer, the backend application processing layer, and the trained model storage layer.

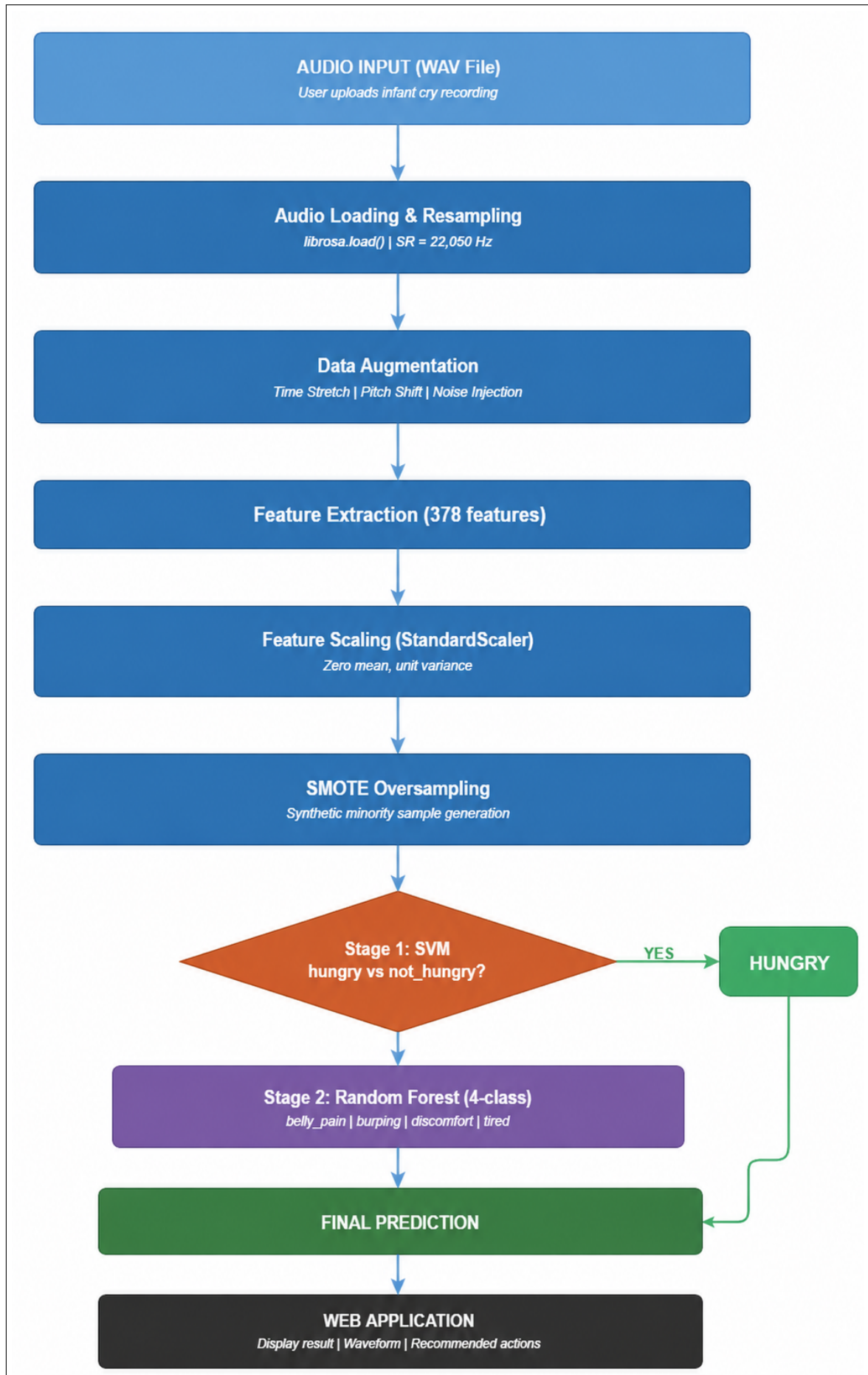


Figure 3.3: Full system pipeline flowchart

3.7 Conclusion

This chapter has presented the complete proposed methodology for the infant cry classification system. The pipeline integrates audio preprocessing, targeted data augmentation for minority classes, comprehensive acoustic feature extraction (MFCCs, Chroma, ZCR ...), SMOTE-based class balancing, and a novel two-stage hierarchical classification architecture combining SVM and Random Forest. The system was further operationalized through a Streamlit web application that enables real-world deployment and usability. The following chapter presents the implementation details, experimental results, and a thorough comparative analysis of the proposed system against classifiers.

CHAPTER 4

IMPLEMENTATION, EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the implementation details and experimental evaluation of the proposed infant cry classification system. The chapter begins with a description of the experimental environment, including hardware specifications and software dependencies. It then reports and analyzes the results of all classification models evaluated in this study. The chapter concludes with a discussion of the web application implementation, system limitations, and directions for future work.

4.2 Software Tools and Libraries

The following tools and libraries were used throughout the development of this project, each playing a key role in data processing, model training, and application deployment. Figure 4.1 shows Software Tools and Libraries Used in This Project.

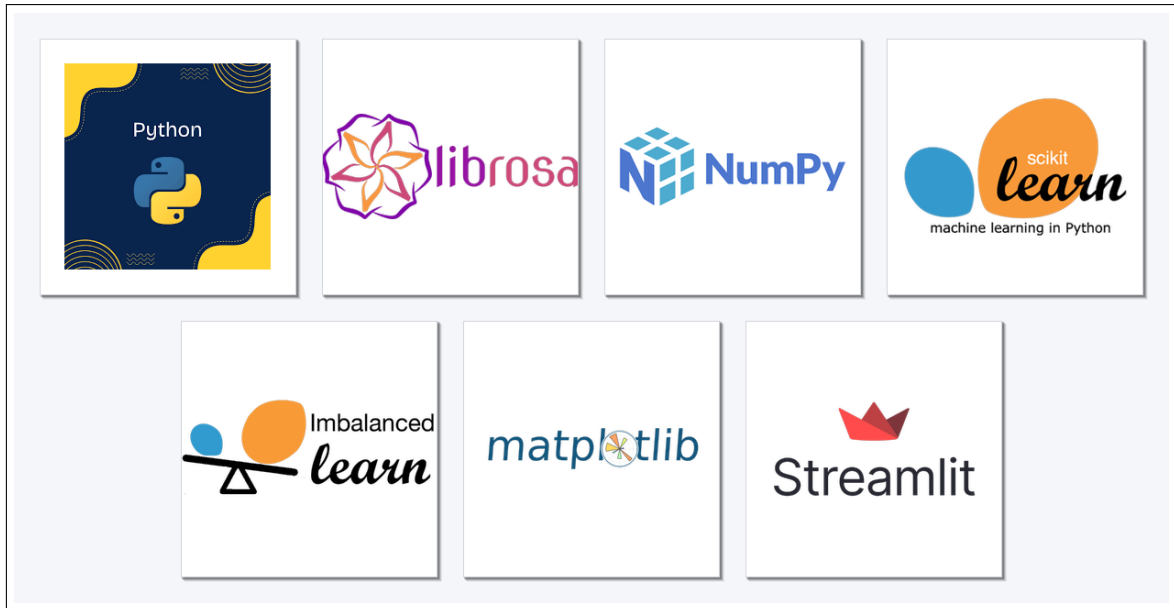


Figure 4.1: Software tools and libraries used in this project.

- **Python:** Python is a well-crafted language suitable for programming in the real world. It is a general-purpose, dynamic, object-oriented, and high-level programming language that utilizes interpreters and has a wide range of applications. Due to its simplicity, readability, and rich ecosystem of scientific libraries, Python was chosen as the primary programming language for this project. [36]
- **Librosa:** Librosa is an open-source Python library designed for audio and music signal analysis. It provides a comprehensive set of tools for feature extraction from audio signals, including Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, and zero-crossing rate, which are essential for building audio classification systems. In this project, Librosa was used to load audio files, apply data augmentation techniques such as time stretching and pitch shifting, and extract the acoustic features fed into the classification pipeline. [37]
- **NumPy:** NumPy is the fundamental library for scientific computing in Python. It provides support for large multi-dimensional arrays and matrices, along with a rich collection of high-level mathematical functions to operate on them. In this project, NumPy was used throughout the data processing pipeline for array manipulation, feature stacking, label encoding, and random state management to ensure fully deterministic and reproducible results. [38]
- **Scikit-learn:** Scikit-learn is one of the most widely used open-source machine learning libraries for Python. It provides efficient and reliable implementations of a broad range of supervised and unsupervised learning algorithms. In this project, two main algorithms were employed: Support Vector Machine (SVM), used in the first stage to perform binary classification between hungry and

not-hungry cry types, and Random Forest, used in the second stage to distinguish between the remaining four cry categories. Beyond model training, Scikit-learn also provided essential utilities such as StandardScaler for feature normalization, StratifiedKFold for cross-validation, and tools for model evaluation including classification reports and confusion matrices. [39]

- **Imbalanced-learn:** Imbalanced-learn is a Python library specifically developed to address the challenge of class imbalance in machine learning datasets. It is fully compatible with Scikit-learn and provides a range of resampling techniques, among which SMOTE (Synthetic Minority Over-sampling Technique) was employed in this project. SMOTE generates synthetic samples for minority classes by interpolating between existing instances, which significantly improves classifier performance in scenarios where certain cry types are underrepresented in the dataset. [40]
- **Matplotlib:** Matplotlib is a comprehensive Python library for creating static, animated, and interactive visualizations. It was used in this project to generate and save confusion matrices for each stage of the classification pipeline, providing a clear visual representation of the model performance across all cry categories. [41]
- **Streamlit:** Streamlit is an open-source Python framework that allows developers to rapidly build and deploy interactive web applications directly from Python scripts, without requiring any front-end development expertise. It follows a simple script-based paradigm where standard Python code is automatically rendered as a fully interactive web interface. In this project, Streamlit was used to develop the web application connected to the trained classification model, enabling users to upload infant cry audio files and receive real-time predictions through an intuitive and accessible interface. [42]

4.3 Models Evaluation

4.3.1 SVM Classifier (5-Class)

The first model is a SVM classifier configured with an RBF kernel ($C=10$, $\gamma='auto'$) applied directly to the five-class problem. The model was trained on the SMOTE-balanced training set and evaluated on the stratified 15% hold-out test set. Table 4.1 presents the per-class classification report.

Class	Precision	Recall	F1-Score	Support
belly_pain	0.82	0.75	0.78	12
burping	1.00	0.57	0.73	14
discomfort	0.79	0.96	0.86	23
hungry	0.97	1.00	0.98	57
tired	0.89	0.89	0.89	19
macro avg	0.89	0.83	0.85	125

Table 4.1: Classification report: SVM (5-class).

The SVM achieved an overall accuracy of 90.0% and a macro-averaged F1-score of 0.850. While the classifier performed well on the hungry class (F1=0.98) and the discomfort class (F1=0.86), it struggled significantly with the burping class (F1=0.73), which is the smallest and most acoustically variable class in the dataset. The confusion matrix below (Figure 4.2) provides a detailed view of misclassification patterns.

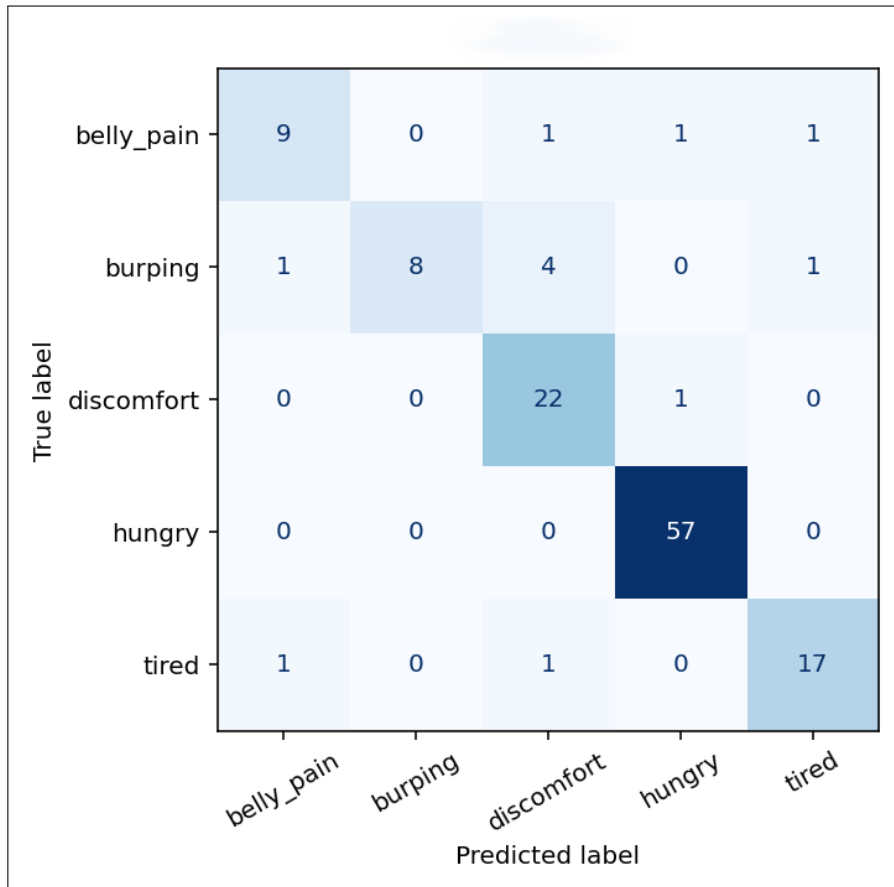


Figure 4.2: Confusion matrix: SVM (5-class).

4.3.2 Random Forest Classifier (5-Class)

The second model is a Random Forest classifier (`n_estimators=1000`, `max_depth=20`, `min_samples_leaf=1`, `class_weight='balanced'`) applied directly to the five-class problem using the same shared test set. Table 4.2 presents the per-class results.

Class	Precision	Recall	F1-Score	Support
belly_pain	0.80	0.67	0.73	12
burping	1.00	0.64	0.78	14
discomfort	0.79	0.83	0.81	23
hungry	0.87	0.96	0.92	57
tired	0.84	0.84	0.84	19
macro avg	0.86	0.79	0.82	125

Table 4.2: Classification report: RF (5-class).

The RF achieved a lower overall accuracy of 86.0% and a macro F1-score of 0.816 compared to the SVM. Notably, the RF exhibited significant confusion between the tired and hungry classes. This cross-class confusion is a recurring challenge in multi-class approaches where all five categories compete simultaneously in a single decision space. Figure 4.3 illustrates the confusion matrix for the RF .

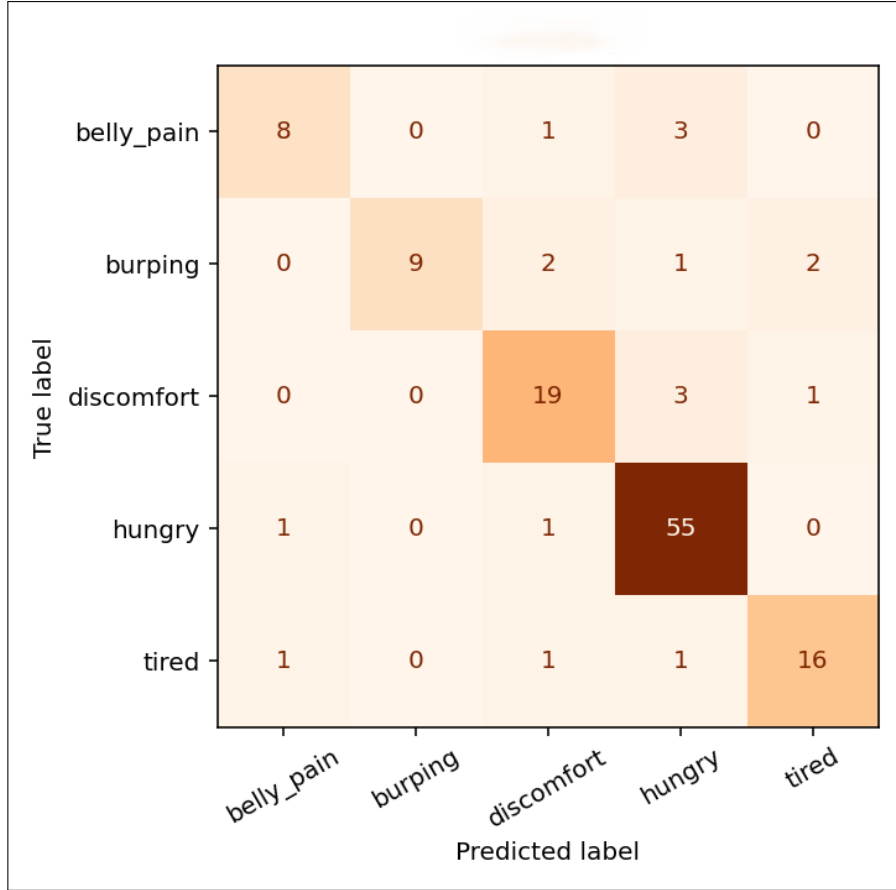


Figure 4.3: Confusion matrix: RF (5-class).

4.4 Hierarchical Model Evaluation

4.4.1 Stage 1 Evaluation — SVM Binary Classifier

The Stage 1 SVM binary classifier was evaluated on its ability to distinguish hungry from not_hungry cries. Table 4.3 presents the classification report for this binary task.

Class	Precision	Recall	F1-Score	Support
hungry	1.00	1.00	1.00	57
not_hungry	1.00	1.00	1.00	68
macro avg	1.00	1.00	1.00	125

Table 4.3: Classification report: Stage 1 SVM binary classifier (hungry vs. not_hungry).

Stage 1 achieved exceptional performance with a macro-averaged F1-score of 1.00 and near-perfect precision and recall for both classes. Out of 57 true hungry samples, all were correctly identified. The not_hungry class also achieved perfect recall (all 68 samples correctly identified). This perfect accuracy at Stage 1 ensures

that the hierarchical pipeline begins with a reliable first-stage filter, minimizing error propagation to Stage 2. Figure 4.4 shows the binary confusion matrix.

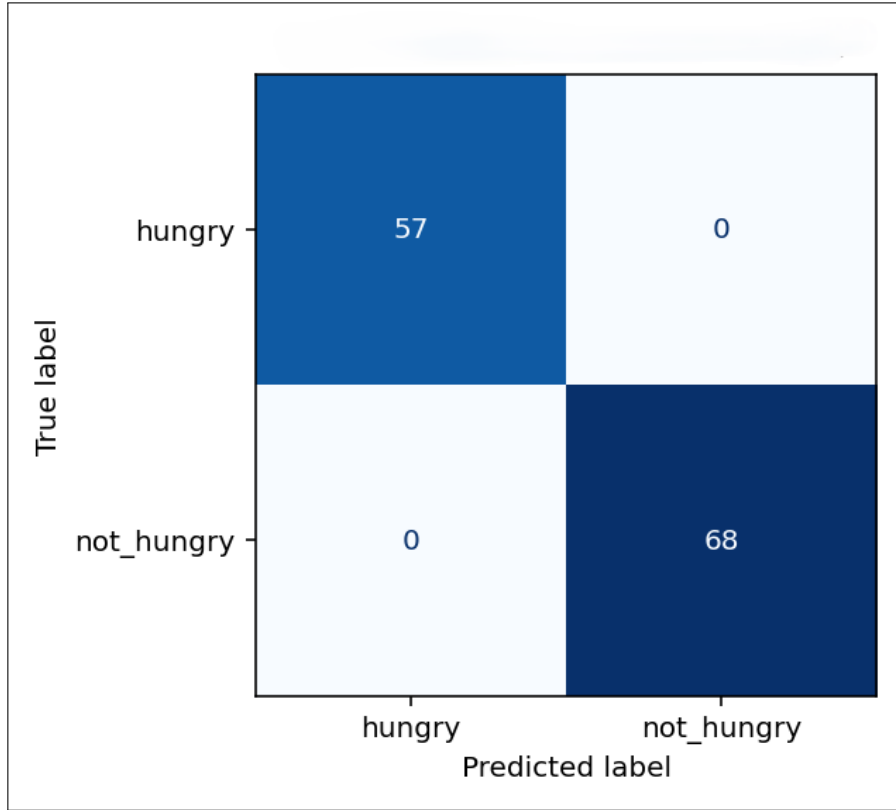


Figure 4.4: Confusion matrix: Stage 1 SVM binary classifier (hungry vs. not_hungry).

4.4.2 Stage 2 Evaluation — Random Forest Multi-class Classifier

The Stage 2 Random Forest classifier was evaluated on the four-class sub-problem (belly_pain, burping, discomfort, tired). This classifier operates on samples that have already been identified as not_hungry by Stage 1. Table 4.4 presents the per-class results for Stage 2.

Class	Precision	Recall	F1-Score	Support
belly_pain	0.92	0.92	0.92	12
burping	1.00	0.71	0.83	14
discomfort	0.82	1.00	0.90	23
tired	0.94	0.89	0.90	19
macro avg	0.92	0.88	0.89	68

Table 4.4: Classification report: Stage 2 Random Forest multi-class classifier (4-class).

Stage 2 achieved a macro F1-score of 0.89, with the most notable challenges encountered in the burping class (F1=0.83) due to its limited number of training

samples and high acoustic overlap with discomfort. The discomfort class achieved the highest recall (1.00), reflecting its dominant presence in the Stage 2 training partition. Figure 4.5 displays the Stage 2 confusion matrix.

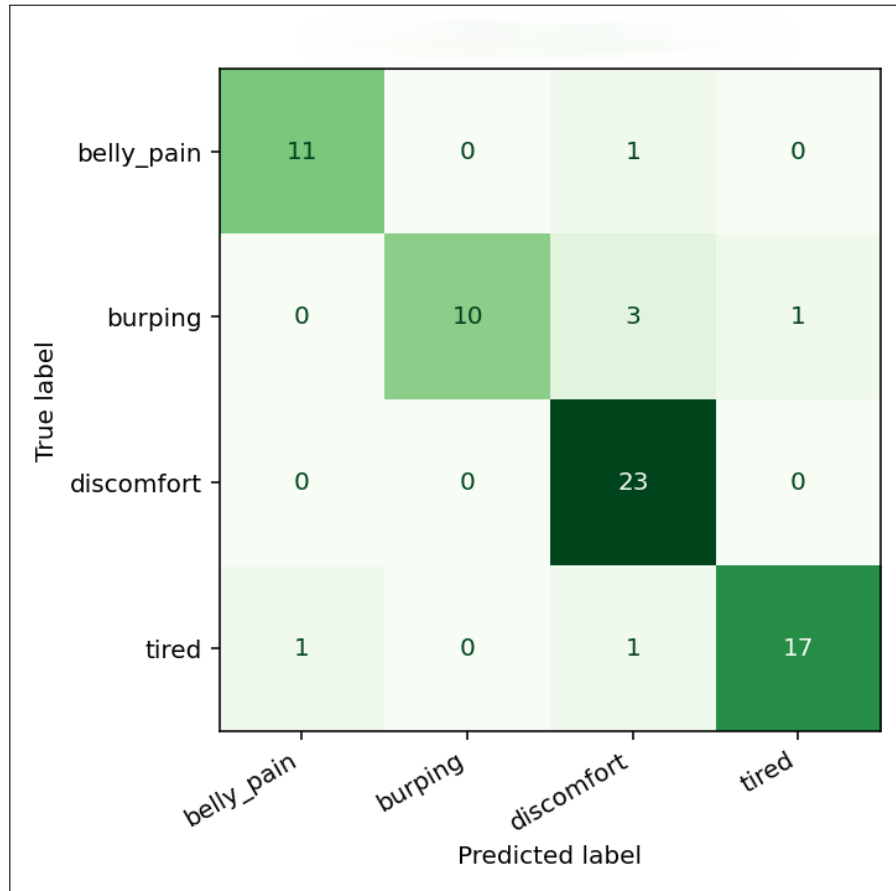


Figure 4.5: Confusion matrix: Stage 2 Random Forest (4-class).

4.4.3 Full Hierarchical Pipeline Evaluation

The complete hierarchical pipeline (Stage 1 SVM + Stage 2 RF) was evaluated end-to-end on the shared 15% test set. Table 4.5 presents the full per-class classification report.

Class	Precision	Recall	F1-Score	Support
belly_pain	0.92	0.92	0.92	12
burping	1.00	0.71	0.83	14
discomfort	0.82	1.00	0.90	23
hungry	1.00	1.00	1.00	57
tired	0.94	0.89	0.92	19
macro avg	0.94	0.91	0.91	125

Table 4.5: Classification report: Full Hierarchical Pipeline (SVM + RF).

The full hierarchical pipeline achieved an overall accuracy of 94.4% and a macro-averaged F1-score of 0.91, representing a substantial improvement over both s. Perfect classification was achieved for hungry (F1=1.00), and strong performance was obtained for belly_pain (F1=0.92), discomfort (F1=0.90), and tired (F1=0.92). The most challenging class remained burping (F1=0.83), where a limited number of samples and acoustic overlap with discomfort contributed to misclassifications. Figure 4.6 display the confusion matrix for the hierarchical pipeline.

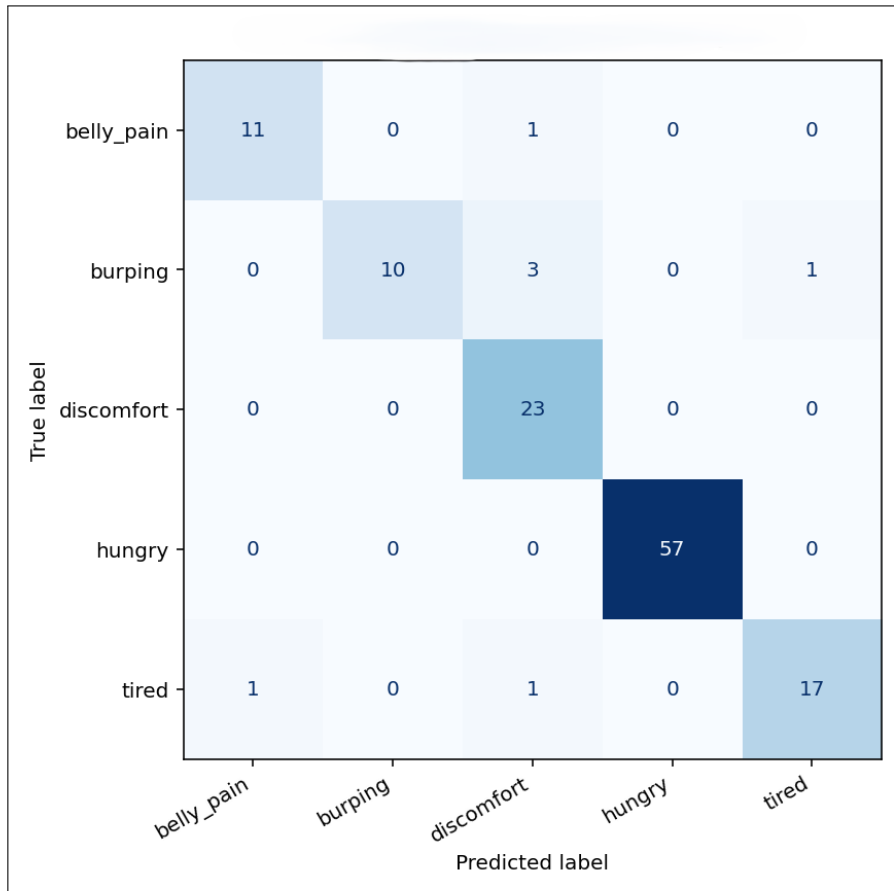


Figure 4.6: Confusion matrix: Hierarchical SVM + RF (combined view).

4.4.4 Error Analysis

Examination of the misclassified samples reveals one primary source of error. Stage 1 performed perfectly with no misclassifications between hungry and not_hungry classes, eliminating the risk of error propagation from Stage 1 to Stage 2. The burping class consistently exhibited lower performance across all models, attributable to its small sample size (8 original recordings) and its high acoustic similarity to discomfort cries. Increasing the number of burping recordings in future work would likely be the most impactful intervention for improving overall system performance.

4.5 Comparative Analysis

4.5.1 Comparison Between Models

Table 4.6 summarizes the performance of all three models evaluated in this study across accuracy and F1-macro metrics.

Model	Accuracy	F1-Macro
SVM ()	0.90	0.85
RF ()	0.86	0.82
Hierarchical (SVM+RF)	0.94	0.91

Table 4.6: Comparative performance summary: all models.

Figure 4.7 provides a visual comparison of F1-macro and accuracy scores across all three models, clearly illustrating the superior performance of the hierarchical pipeline.

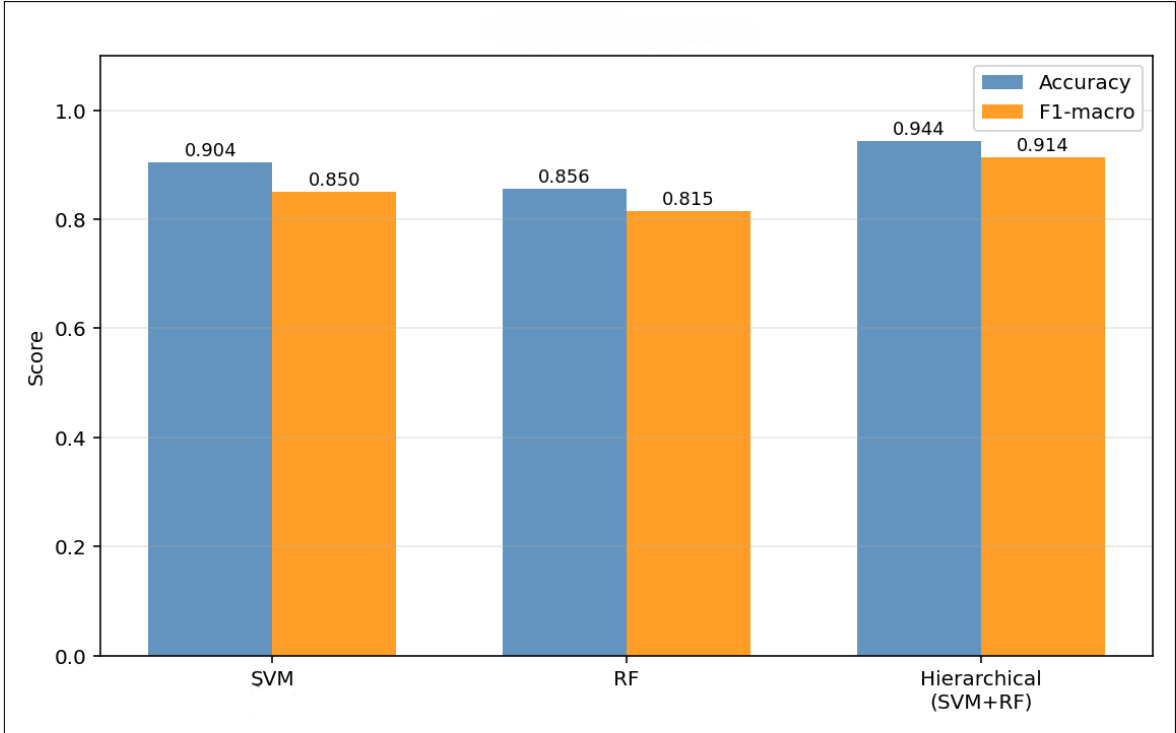


Figure 4.7: Model comparison bar charts .

4.5.2 Discussion of Results

The experimental results confirm that the hierarchical SVM + RF pipeline significantly outperforms both s across all evaluation metrics. The hierarchical approach achieved a 6.4 percentage point improvement in accuracy over the SVM (94.4% vs. 90.0%) and a 6.0 percentage point improvement in F1-macro (0.91 vs. 0.85). Several factors contribute to this substantial gain.

First, the hierarchical decomposition reduces the decision complexity at each stage by separating the acoustically distinct hungry class from the remaining four classes at Stage 1, allowing Stage 2 to focus on a more homogeneous, acoustically similar sub-group. Second, the use of SMOTE specifically calibrated for each stage (with appropriate k-neighbor settings) ensures that both classifiers are trained on balanced representations without introducing artificial artifacts that could degrade performance. Third, the combination of SVM and Random Forest leverages the complementary strengths of both algorithms: SVM’s effectiveness in high-dimensional binary separation and Random Forest’s robustness for small-sample multi-class problems.

The Random Forest underperformed relative to the SVM, likely due to the high variance of individual decision trees when training data per class is limited. The `class_weight='balanced'` setting partially compensated for imbalance but could not fully overcome the challenge of distinguishing five acoustically similar classes in a single decision space.

4.6 Web Application Implementation

4.6.1 Application Overview

A fully functional web application was developed using Streamlit to provide an accessible, real-world interface for the trained hierarchical classification system. The application is designed for use by parents and caregivers who wish to identify the reason behind their infant’s cry without requiring any technical expertise. The interface presents classification results in an intuitive, visually rich format accompanied by practical recommended caregiver actions for each predicted cry type.

4.6.2 User Interface Design

Figure 4.8 shows the application homepage, which prominently displays the system’s key performance metrics (94.4% accuracy, 0.91 F1-macro score, 5 cry categories, 2 pipeline stages) and a model status indicator confirming the successful loading of the hierarchical classifier.



Figure 4.8: Streamlit application homepage.

Figure 4.9 demonstrates the audio upload interface and the waveform visualization

panel. Upon uploading a WAV file, the application renders an interactive audio player and a time-domain waveform plot, allowing the user to preview the recording before classification.

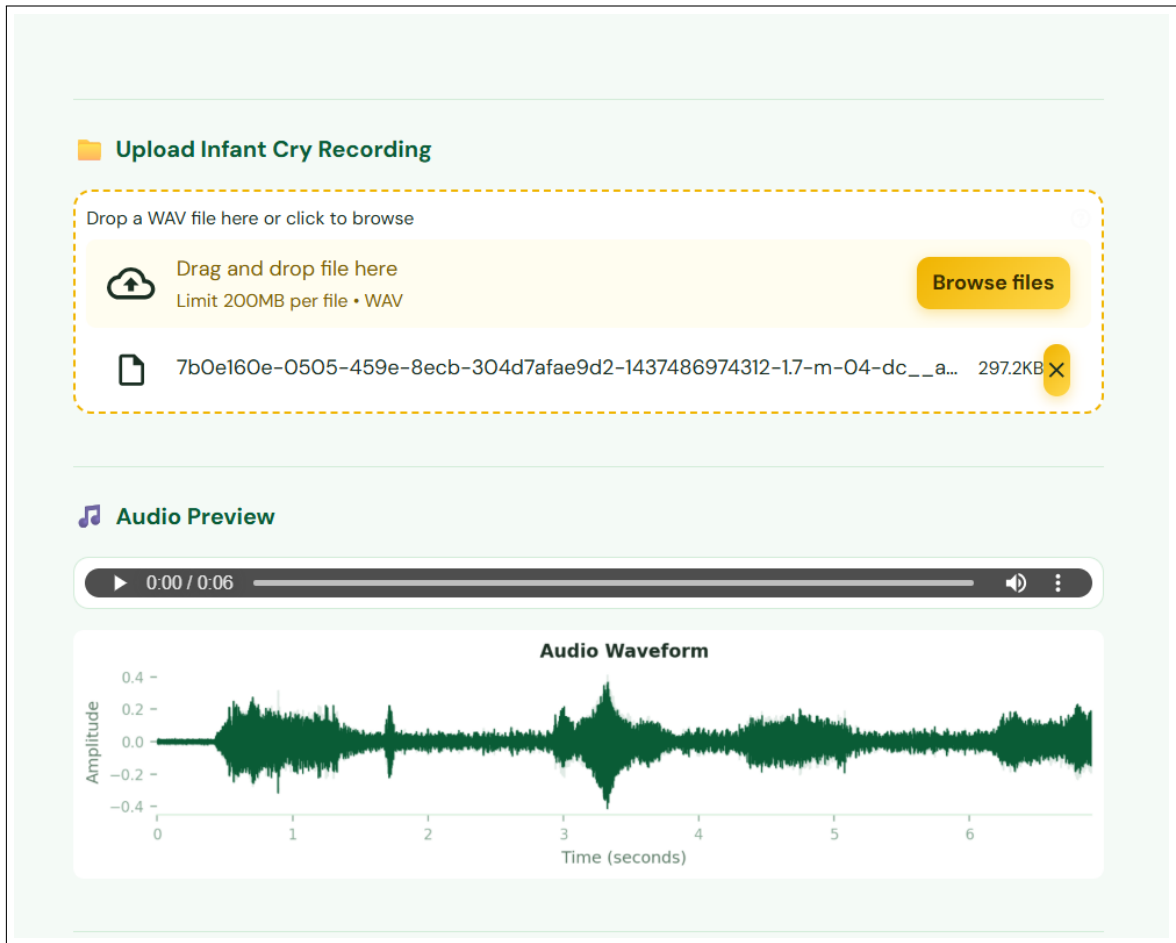


Figure 4.9: Audio upload interface and waveform visualization.

Figures 4.10 and 4.11 show the prediction result panel, which displays the predicted cry category with a descriptive explanation, the confidence score, the pipeline stage at which the prediction was made, and a horizontal bar chart of the full probability distribution across all five classes. Additionally, a “Recommended Actions” section provides actionable guidance for caregivers.

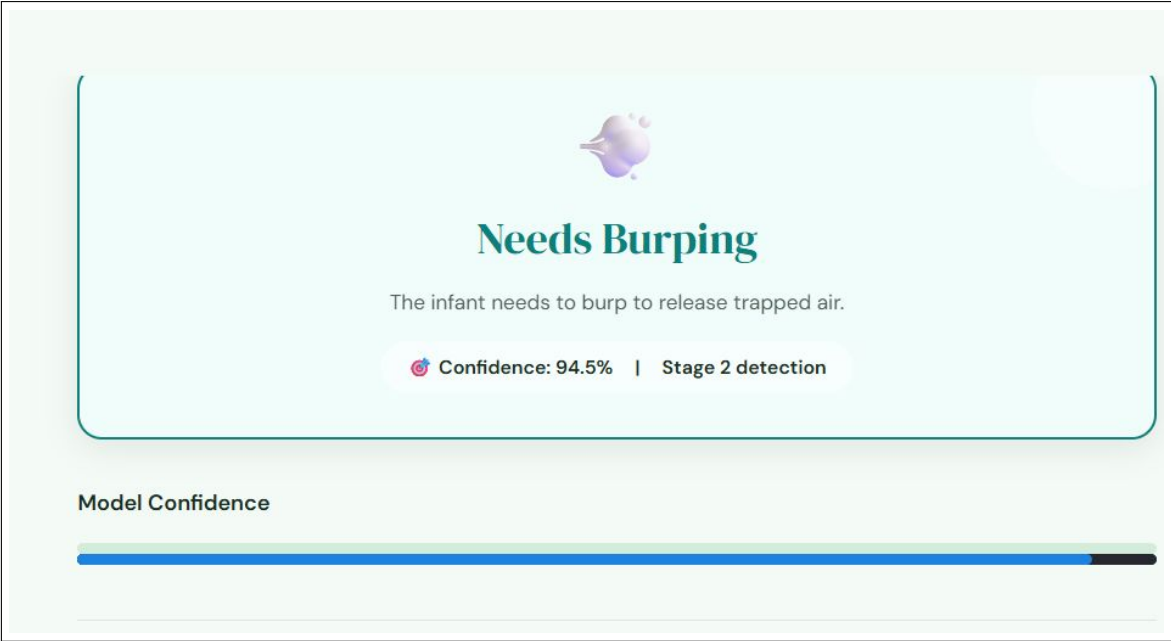


Figure 4.10: Prediction result panel.

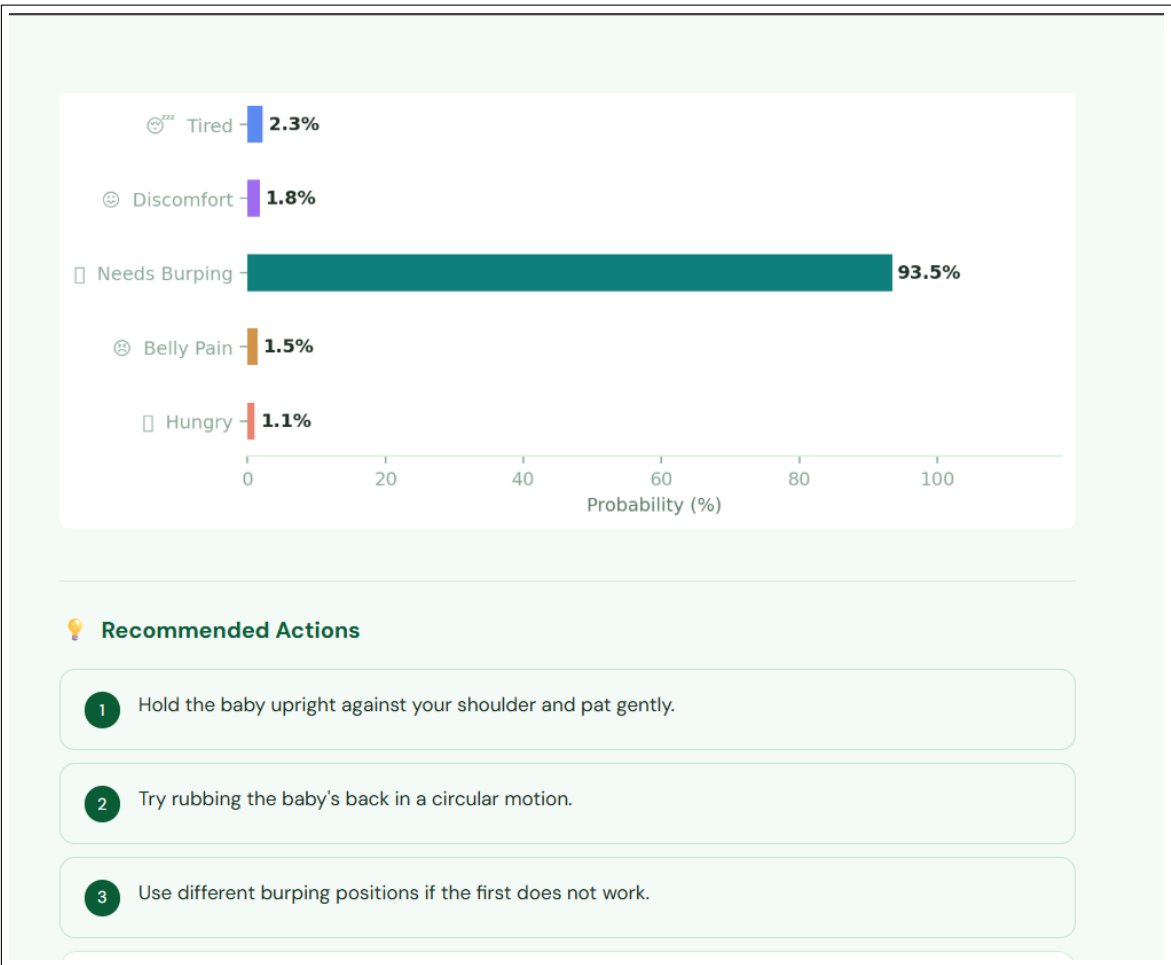


Figure 4.11: Probability distribution chart and recommended caregiver actions.

4.6.3 Backend Implementation

The backend of the Streamlit application performs the following sequence of operations upon receiving an uploaded audio file. The WAV file is read from memory using `librosa`, resampled to 22,050 Hz, and passed through the same 378-dimensional feature extraction pipeline used during model training. The extracted feature vector is then scaled using the pre-loaded Stage 1 `StandardScaler` and classified by the Stage 1 SVM binary classifier. If the prediction is `'hungry'`, the final result is returned immediately. Otherwise, the feature vector is scaled using the Stage 2 `StandardScaler` and passed to the Stage 2 Random Forest classifier, which outputs one of the four remaining class labels along with a class probability vector. Prediction latency is negligible, with the full inference pipeline completing in under one second for typical WAV files of 7–15 seconds duration.

4.7 Limitations and Future Work

4.7.1 Current Limitations

Despite the strong performance of the proposed system, several limitations should be acknowledged. The most fundamental constraint is the small size and limited diversity of the Donate-a-Cry Corpus dataset. With only approximately 8–382 recordings per class, the dataset cannot fully represent the acoustic variability of infant cries across different ages, health conditions, recording environments, and cultural backgrounds. The burping class in particular, with its small sample count, remains the most challenging to classify reliably. Additionally, the traditional machine learning approach adopted in this study, while effective on this dataset, may not scale to larger and more complex corpora as effectively as deep learning-based end-to-end approaches.

4.7.2 Future Improvements

Several promising directions exist for extending and improving this work. The most impactful future enhancement would be the adoption of deep learning architectures particularly Convolutional Neural Networks (CNNs) applied to mel-spectrogram representations, or Long Short-Term Memory (LSTM) networks for modeling temporal dependencies in cry sequences, which have demonstrated state-of-the-art performance in audio classification tasks. Real-time prediction capability could be added by integrating a microphone input interface directly into the Streamlit application. Expanding the dataset through additional crowdsourcing campaigns or collaboration with pediatric hospitals would significantly improve model robustness and generalizability. Finally, deployment on mobile platforms (iOS/Android) or

cloud-hosted infrastructure would extend the system's reach to a broader population of users.

4.8 Conclusion

In conclusion, this chapter presented a comprehensive evaluation of the proposed hierarchical infant cry classification system. The results demonstrated the effectiveness of the hierarchical approach in improving classification performance compared to traditional methods. In addition, the use of data augmentation, class balancing, and optimized feature extraction contributed significantly to enhancing the system's reliability and efficiency. The development of a Streamlit web application also highlighted the practical applicability of the proposed solution in real-world scenarios. Future work will focus on integrating deep learning techniques and expanding datasets to further advance automated infant cry analysis systems.

GENERAL CONCLUSION

This thesis investigated the problem of automatic infant cry classification using machine learning and audio signal processing techniques. The motivation behind this work stems from the important role that crying plays as the primary communication mechanism available to infants during the early stages of life. Developing intelligent systems capable of analysing cry signals can provide valuable assistance in understanding infant needs and supporting caregivers in their daily interactions with children.

To address this problem, a complete classification pipeline was designed and implemented using the Donate-a-Cry Corpus. The proposed system integrates several stages, including audio preprocessing, acoustic feature extraction, class balancing, machine learning classification, and real-time deployment through a web-based application.

A comprehensive preprocessing framework was developed to improve signal quality and ensure data consistency. Acoustic information was represented through a carefully engineered 378-dimensional feature vector combining MFCC coefficients, delta and delta-delta derivatives, chroma descriptors, and Zero Crossing Rate features. These representations successfully captured relevant characteristics of infant cry signals and provided informative inputs for machine learning algorithms.

Because the dataset exhibits a severe imbalance among cry categories, particular attention was devoted to data balancing strategies. Targeted data augmentation techniques were applied to minority classes, while SMOTE-based synthetic oversampling further improved class representation during training. These techniques contributed significantly to improving classifier robustness and reducing bias toward majority classes.

The principal contribution of this work is the proposed Hierarchical Classification Framework (HCF). Instead of addressing the problem directly as a single five-class

classification task, the framework decomposes it into two successive decision stages. The first stage isolates hungry cries from all remaining categories using a Support Vector Machine classifier, while the second stage employs a Random Forest classifier to discriminate among the four acoustically similar minority classes. This hierarchical structure proved particularly effective in handling the characteristics of the Donate-a-Cry dataset.

Experimental evaluation demonstrated that the proposed approach achieved an overall accuracy of **94.4%** and a macro-averaged F1-score of **0.91**. These results outperform the conventional classification models evaluated in this study and confirm the effectiveness of hierarchical decision-making for imbalanced infant cry classification tasks.

Beyond the machine learning contribution, a complete web application was developed using **Streamlit**. The application allows users to upload audio recordings and obtain real-time predictions together with confidence scores and detailed classification information. This practical implementation demonstrates the feasibility of transforming research outcomes into accessible tools that can support parents and caregivers.

Despite the encouraging results obtained, several limitations remain. The Donate-a-Cry Corpus is relatively small and may not fully represent the diversity of infant vocalisations encountered in real-world environments. Furthermore, certain categories remain under-represented and acoustically similar, making them inherently difficult to distinguish.

Future work may focus on collecting larger and more diverse datasets, exploring deep learning architectures such as Convolutional Neural Networks (CNN) and hybrid CNN-LSTM models, improving real-time processing capabilities, and developing adaptive systems capable of learning the vocal patterns of individual infants over time. Additional research may also investigate multimodal approaches that combine audio information with contextual or behavioural data to further improve classification performance.

In conclusion, the results obtained in this thesis demonstrate that classical machine learning techniques, when combined with appropriate feature engineering, class balancing strategies, and hierarchical decision structures, can achieve highly competitive performance in infant cry classification. The proposed Hierarchical Classification Framework provides an effective and interpretable solution for recognising infant needs from cry signals and represents a meaningful contribution to the growing field of AI-assisted infant communication understanding.

BIBLIOGRAPHY

- [1] S. World, “Sound,” 2022. [Online]. Available: <https://www.scienceworld.ca/resource/sound/>
- [2] C. Ji, T. B. Mudiyanse, Y. Gao, and Y. Pan, “A review of infant cry analysis and classification,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, no. 1, p. 8, 2021.
- [3] E. Koffi, “Is intensity (decibel levels) the most aversive correlate in infant cry?: Preliminary and exploratory results,” *Linguistic Portfolios*, vol. 11, no. 1, p. 2, 2022.
- [4] S. D. Cano-Ortiz, Y. Martinez-Canete, and L. Veranes-Vicet, “Evaluating new set of acoustical features for cry signal classification,” in *Mexican Conference on Pattern Recognition*. Springer, 2022, pp. 149–158.
- [5] M. Hammoud, M. N. Getahun, A. Baldycheva, and A. Somov, “Machine learning-based infant crying interpretation,” *Frontiers in Artificial Intelligence*, vol. 7, p. 1337356, 2024.
- [6] S. W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*, 2nd ed. California Technical Publishing, 1999. [Online]. Available: <https://www.dspguide.com/>
- [7] X. Qiao, S. Jiao, H. Li, G. Liu, X. Gao, and Z. Li, “Infant cry classification using an efficient graph structure and attention-based model,” *Kuwait Journal of Science*, vol. 51, no. 3, p. 100221, 2024.
- [8] P. C. Loizou, *Speech Enhancement: Theory and Practice*, 2nd ed. Boca Raton, FL: CRC Press, 2013.

- [9] T. Sainburg, “Noise reduction in python using spectral gating,” Dec 2022. [Online]. Available: <https://github.com/timsainb/noisereduce>
- [10] I. Stewart, “What is audio normalization?” May 2023. [Online]. Available: https://www.izotope.com/community/blog/audio-normalization?srsId=AfmBOoo18bROTpoXQ1Q9zlk6w8B4IFS-F-THzznaUNLzgpM8_92sixR8
- [11] Z. K. Abdul and A. K. Al-Talabani, “Mel frequency cepstral coefficient and its applications: A review,” *Ieee Access*, vol. 10, pp. 122 136–122 158, 2022.
- [12] R. Kumar, “Audio features,” 2024. [Online]. Available: https://ravinkumar.com/GenAiGuidebook/audio/audio_feature_extraction.html
- [13] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015, vol. 5.
- [14] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [15] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, “Music type classification by spectral contrast feature,” in *Proceedings. IEEE international conference on multimedia and expo*, vol. 1. IEEE, 2002, pp. 113–116.
- [16] C. Harte, M. Sandler, and M. Gasser, “Detecting harmonic change in musical audio,” in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, 2006, pp. 21–26.
- [17] IBM, “Feature selection,” Mar 2025. [Online]. Available: <https://www.ibm.com/think/topics/feature-selection>
- [18] S. Russell, P. Norvig, and A. Intelligence, “A modern approach,” *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, vol. 25, no. 27, pp. 79–80, 1995.
- [19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [20] S. M. H. Hashemi, H. Kolivand, W. Khan, and T. Saba, “Infant cry analysis: A survey of datasets, features, and machine learning techniques,” *IEEE Transactions on Affective Computing*, 2025.
- [21] Y. Zayed, A. Hasasneh, and C. Tadj, “Infant cry signal diagnostic system using deep learning and fused features,” *Diagnostics*, vol. 13, no. 12, p. 2107, 2023.
- [22] B. Bakirarar and A. H. Elhan, “Class weighting technique to deal with imbalanced class problem in machine learning: methodological research,” *Türkiye Klinikleri Biyoistatistik*, vol. 15, no. 1, pp. 19–29, 2023.

- [23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [24] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [25] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice Hall, 1994.
- [26] S. Kumar, *Neural Networks: A Classroom Approach*. New York, NY: Tata McGraw-Hill, 2004.
- [27] M. H. Sazlı, “A brief review of feed-forward neural networks,” *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 50, no. 01, 2006.
- [28] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [29] J. Chunyan, M. Chen, L. Bin, and Y. Pan, “Infant cry classification with graph convolutional networks,” in *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2021, pp. 322–327.
- [30] N. V. Dharwadkar, A. A. Dixit, A. K. Kannur, M. A. B. Kadampur, and S. Joshi, “Identification of reasons behind infant crying using acoustic signal processing and deep neural network for neonatal intensive care unit,” *International Journal of Information Retrieval Research (IJIRR)*, vol. 12, no. 1, pp. 1–17, 2022.
- [31] Y.-C. Liang, I. Wijaya, M.-T. Yang, J. R. Cuevas Juarez, and H.-T. Chang, “Deep learning for infant cry recognition,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 10, p. 6311, 2022.
- [32] V. R. Joshi, K. Srinivasan, P. D. R. Vincent, V. Rajinikanth, and C.-Y. Chang, “A multistage heterogeneous stacking ensemble model for augmented infant cry classification,” *Frontiers in Public Health*, vol. 10, p. 819865, 2022.
- [33] A. Ekinici and E. Küçükkülahlı, “Classification of baby cries using machine learning algorithms,” *Eastern Anatolian Journal of Science*, vol. 9, no. 1, pp. 16–26, 2023.
- [34] S. A. Younis, D. Sobhy, and N. S. Tawfik, “Evaluating convolutional neural networks and vision transformers for baby cry sound analysis,” *Future Internet*, vol. 16, no. 7, p. 242, 2024.

- [35] K. Rezaee, H. G. Zadeh, L. Qi, H. Rabiee, and M. R. Khosravi, “Can you understand why i am crying? a decision-making system for classifying infants’ cry languages based on deepsvm model,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 1, pp. 1–17, 2024.
- [36] Python Software Foundation, “Welcome to Python.org documentation,” 2019, accessed: 2026-06-04. [Online]. Available: <https://www.python.org/doc/>
- [37] Librosa Development Team, “Librosa documentation,” 2020, accessed: 2026-06-04. [Online]. Available: <https://librosa.org/doc/latest/index.html>
- [38] NumPy Developers, “Numpy documentation and manual,” 2022, accessed: 2026-06-04. [Online]. Available: <https://numpy.org/doc/stable/>
- [39] Scikit-learn Developers, “scikit-learn: Machine learning in Python,” 2024, accessed: 2026-06-04. [Online]. Available: <https://scikit-learn.org/stable/>
- [40] Imbalanced-learn Contributors, “Imbalanced-learn documentation,” 2021, accessed: 2026-06-04. [Online]. Available: <https://imbalanced-learn.org/stable/>
- [41] The Matplotlib Development Team, “Matplotlib: Python plotting documentation,” 2024, accessed: 2026-06-04. [Online]. Available: <https://matplotlib.org/stable/index.html>
- [42] Streamlit Developers, “Streamlit documentation,” 2026, accessed: 2026-06-04. [Online]. Available: <https://docs.streamlit.io/>