Fetal Distress Classification Based on Cardiotocography

A PROJECT REPORT

Submitted by,

Mr. Akshay Menon - 20201CSD0033

Under the guidance of,

Dr. Harishkumar KS

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

ABSTRACT

This paper involves a comprehensive exploration into the development of a highly accurate machine learning-driven solution for intrapartum fetal distress classification based on cardiotocography (CTG). In traditional obstetric care, the diagnosis of fetal distress during labor has been subjective, relying heavily on the expertise of healthcare professionals. To overcome this limitation, our study proposes an objective approach utilizing advanced machine learning techniques, with a specific focus on a one-dimensional Convolutional Neural Network (CNN) model. The foundation of our research lies in the analysis of the CTU-CHB Intrapartum Cardiotocography Database, a dataset capturing detailed fetal heart rate patterns during labor. By employing state-of-the-art neural network architectures, our goal is to extract intricate patterns and relationships from CTG data, providing a more accurate and objective means of diagnosing intrapartum fetal distress. A significant challenge in this domain is the subjective nature of traditional diagnostic criteria. Our study addresses this challenge by identifying key combinations of neonatal criteria, such as umbilical cord pH and Apgar scores, and establishing optimal thresholds through rigorous analysis. This not only enhances the precision of our classification model but also aligns it with established medical standards. Furthermore, to overcome the challenges posed by the small size and class imbalance of the dataset, we implement data augmentation techniques. This involves creating synthetic variations of the existing data, thereby expanding the diversity of the dataset, and improving the model's ability to generalize to unseen samples. This step is crucial in ensuring that our model learns robust features representative of the broader population. The experimental results demonstrate the efficacy of our proposed approach. The one-dimensional CNN model achieves a remarkable validation accuracy of 98.3%, highlighting its ability to accurately classify intrapartum fetal distress cases. This high level of accuracy suggests not only the capability of our model to learn complex patterns from the CTG data but also its potential for generalization to real-world scenarios.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

Intrapartum fetal distress, a complex manifestation of the fetus's response to oxygen limitations during labor, manifests through irregularities in heart rate patterns. The diagnostic process is inherently subjective, relying on the nuanced expertise of healthcare professionals. The intricate nature of fetal distress emerges from a confluence of clinical factors, including neonatal parameters such as umbilical cord pH and Apgar scores, maternal risk factors like hypertension or diabetes, and other variables such as the progression of labor.

Cardiotocography (CTG), introduced in the mid-20th century, stands as a cornerstone in the evaluation of fetal health, assessing it through continuous monitoring of fetal heart rate (FHR) and uterine contractions (UC). The manual interpretation of CTG data adheres to the DR C BRAVADO approach, a comprehensive methodology involving the definition of risk (DR), study of contractions (C), assessment of baseline FHR (BRA) and its variability (V), identification of accelerations (A), decelerations (D), and culminating in an overall assessment of FHR and UC patterns (O).

1.1 Complexity Of Fetal Distress

Fetal distress is a multifaceted condition, influenced by an array of clinical factors that contribute to its manifestation. These factors include, but are not limited to, the health of the neonate, maternal well-being, and the dynamics of the labor process itself.

1.1.1 Factors Affecting Fetal Distress

In a nuanced exploration, consider the various factors that contribute to the intricate tapestry of fetal distress. Delve into the interplay of neonatal parameters, such as blood gas measurements and Apgar scores, maternal health conditions, including pre-existing hypertension or gestational diabetes, and other clinical variables that play pivotal roles in determining the occurrence and severity of fetal distress during the labor process.

1.2 Limitations in Current Diagnostic Methods

Despite the integration of CTG for fetal distress assessment, the prediction rates of abnormal patterns remain suboptimal. The diagnostic process is impeded by inherent challenges, ranging from the subjective nature of CTG interpretation to historical limitations in the accuracy of fetal distress assessments.

1.2.1 Subjectivity in CTG Interpretation

Embark on an in-depth examination of the challenges posed by the subjective interpretation of CTG data. Investigate how the unique perspectives of individual healthcare professionals may influence their perception and analysis of the same graph, leading to inconsistencies in the diagnosis of fetal distress.

1.2.2 Historical Perspective: Introduction of CTG

Offer a comprehensive historical backdrop on the introduction of CTG by E. Hon in the mid-20th century. Illuminate the pioneering role of CTG in continuous monitoring of FHR and UC, laying the foundation for contemporary intrapartum care practices and shaping the trajectory of fetal health assessment.

1.3 Towards an Objective Approach

Acknowledging the imperative to enhance accuracy and reduce false positives in fetal distress detection, a paradigm shift towards a highly objective approach becomes essential.

1.3.1 Importance of Objective Fetal Distress Detection

Delve into the critical significance of transitioning to an objective approach in fetal distress detection. Examine how such an approach can lead to more precise identification of cases, thereby reducing the risk of false positives and contributing to significant advancements in neonatal outcomes.

1.3.2 Challenges in Current Predictive Rates

Embark on an exploration of the existing challenges that impede improvements in predictive rates of abnormal patterns. Uncover the limitations of current diagnostic methods, emphasizing the need for innovative approaches to overcome these challenges and enhance the accuracy of fetal distress detection, thereby propelling advancements in intrapartum care.

CHAPTER-2

LITERATURE SURVEY

Machine learning-based fetal distress classification has witnessed significant advancements, with researchers employing diverse methodologies to enhance accuracy in identifying distress patterns during labor. In this literature survey, we delve into key studies that have contributed to this evolving field, exploring their approaches, methodologies, and outcomes.

2.1. Deep Learning Approaches in Fetal Distress Classification

2.1.1 Study by Jun Ogasawara et al. [8]

In their investigation, Ogasawara et al. leveraged deep learning concepts for fetal distress classification, utilizing a dataset from Keio University Hospital. The criteria for classification were based on Apgar score and umbilical cord arterial pH. Their implementation of CNN-based (CTG-Net) and LSTM-based models yielded notable success, surpassing the performance of traditional methodologies such as SVM and k-means clustering. This work emphasizes the efficacy of deep learning in enhancing the accuracy of fetal distress classification.

2.1.2 Investigation by Y. D. Daydulo et al. [12]

Daydulo and team employed a deep learning approach, emphasizing the labeling of samples as 'Normal' only if the pH was less than or equal to 7.15. Their methodology incorporated Morse wavelet for signal processing and transfer learning with a modified ResNet50 model. The study showcased high accuracies at different stages of labor, underscoring the robustness of their proposed methodology and its potential impact on accurate fetal distress classification.

2.1.3 Study Led by H. Liang et al. [13]

Following a similar labeling approach based on a pH threshold of less than 7.15, Liang et al. implemented 1D CNN coupled with bidirectional GRU. Their model achieved a commendable accuracy of 96%, further reinforcing the effectiveness of deep learning models in the context of fetal distress classification. The study contributes to the growing body of evidence supporting the role of advanced neural network architectures in improving diagnostic accuracy.

2.2 Variations in Labeling Criteria

2.2.1 Analysis by M. O’Sullivan et al. [14]

O'Sullivan and colleagues adopted a unique labeling criterion of pH <= 7.0 and a low Apgar5 score for distress samples. Their focus on feature engineering, combined with machine learning algorithms like logistic regression and SVM, provided valuable insights into the distinctive characteristics of the CTU-CHB database. This work underscores the importance of nuanced criteria selection and dataset-specific considerations in achieving accurate fetal distress classification.

2.2.2 Dynamic Learning Rates by Anil Johny et al. [15]

While primarily focused on histopathologic image data, Johny et al. introduced the effectiveness of dynamic learning rates in improving model accuracy. Their experimentation highlighted the impact of adjusting learning rates during training, signaling potential advancements in training strategies for machine learning models. This insight contributes to the broader discussion on optimizing model performance in diverse healthcare applications.

2.3 Overview of Existing Criteria

2.3.1 Insight from V. Chuda´cek ˇ et al. [1]

V. Chuda´cek et al.'s foundational work presenting the CTU-CHB database showcased over 21 papers utilizing various distress classification criteria. Notably, pH and Apgar score emerged as the most frequently employed criteria, reflecting the diversity in approaches adopted by researchers. This comprehensive overview sets the stage for understanding the landscape of fetal distress classification and the prevalence of specific criteria in existing literature.

2.4 Towards an Optimal Model

In light of the rich insights gleaned from these studies, our research endeavors to contribute to fetal distress detection using deep learning methodologies. We draw inspiration from successful approaches, taking into account variations in labeling criteria. Our goal is to enhance the accuracy of fetal distress classification based on cardiotocography data, aligning with the evolving landscape of machine learning applications in healthcare.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

3.1 Subjectivity in Fetal Distress Diagnosis:

3.1.1 Historical Context:

Traditional methodologies have often placed a heavy burden on the expertise and experience of healthcare professionals, requiring them to interpret clinical indicators related to fetal distress based on their accumulated knowledge and judgment. This reliance on individual interpretation introduces a significant degree of variability, as different healthcare professionals may assess the same set of indicators differently. This historical context establishes a foundation for understanding the depth of subjectivity inherent in conventional approaches to fetal distress diagnosis.

3.1.2 Variability in Interpretation:

The introduction of variability in the interpretation of clinical indicators, such as fetal heart rate patterns and other relevant parameters, poses a substantial challenge. Healthcare professionals, drawing on their diverse experiences and training backgrounds, may approach the same diagnostic scenario with varying perspectives. This variability can lead to inconsistencies in diagnosis, hindering the establishment of a standardized and universally accepted diagnostic framework.

3.1.3 Gap in Achieving Standardization:

The inherent subjectivity in traditional methods creates a notable gap in the quest for achieving standardized and objective diagnostic practices. Standardization is critical not only for ensuring consistent and reliable diagnoses but also for facilitating effective communication and decision-making within the medical community. The lack of standardized approaches can impede collaborative efforts, potentially affecting patient outcomes and the overall quality of care.

3.1.4 Imperative for Objectivity:

Recognizing the challenges posed by subjectivity, there is a growing imperative for a paradigm shift towards objectivity in fetal distress diagnosis. The need for more standardized approaches becomes evident, emphasizing the importance of implementing objective tools and methodologies that can reduce reliance on individual interpretation. The imperative extends beyond individual cases to encompass the broader healthcare landscape, calling for a systemic transformation in diagnostic practices.

3.1.5 Enhancing Diagnostic Accuracy:

The primary motivation behind advocating for a shift towards objectivity is the enhancement of diagnostic accuracy. By minimizing the influence of subjective factors, the medical community can foster a more reliable and consistent evaluation process for fetal distress. This shift is essential for ensuring that diagnoses are based on standardized criteria, leading to more accurate and timely interventions when necessary.

3.1.6 Comprehensive Reforms:

Recognizing this gap not only underscores the need for immediate improvements but also calls for comprehensive reforms in fetal distress diagnosis methodologies. These reforms should encompass the development and integration of standardized, quantifiable tools that can provide a more objective assessment of fetal well-being. The goal is to create a more consistent and reliable evaluation process that contributes to improved patient outcomes and reinforces trust in diagnostic practices.

3.2 Optimization of Neonatal Criteria:

3.2.1 Identifying Key Combinations:

The optimization of neonatal criteria, specifically focusing on identifying key combinations such as umbilical cord pH and Apgar scores, brings to light a critical gap in current fetal distress diagnostic practices. The traditional diagnostic landscape has often lacked a systematic approach to defining and utilizing neonatal criteria effectively. By honing in on these key combinations, there's an acknowledgment that these parameters play a pivotal role in accurately characterizing the health of the neonate during childbirth.

3.2.2 Lack of Standardized Parameters:

The absence of precisely defined and standardized parameters for accurate classification introduces a level of ambiguity that impedes the progress towards establishing a universally accepted diagnostic framework. Without clear guidelines on how specific neonatal criteria should be utilized and interpreted, there is room for variability in clinical practice. This variability not only impacts the consistency of diagnoses but also poses challenges in ensuring that healthcare professionals are aligned in their understanding of these critical parameters.

3.2.3 Ambiguity and Diagnostic Framework:

Addressing the identified gap involves not only determining the optimal thresholds for neonatal criteria but also delving into the intricacies of how these criteria interact with each other. The ambiguity in the current diagnostic landscape necessitates a meticulous examination of the nuanced relationships between various neonatal parameters. Understanding how these parameters influence each other and contribute to the overall assessment of fetal well-being is crucial for developing a standardized and universally accepted diagnostic framework.

3.2.4 Optimization Process:

The optimization process emerges as a key aspect of refining diagnostic accuracy in fetal distress cases. This goes beyond merely identifying the individual parameters; it involves a systematic and data-driven approach to determining the optimal combinations, thresholds, and interdependencies. By optimizing neonatal criteria, healthcare professionals can enhance the precision of their diagnoses, leading to more informed and timely interventions when necessary.

3.2.5 Nuanced Relationships:

The meticulous examination of the nuanced relationships between various neonatal parameters is vital for establishing a robust foundation for classification. It requires a deep understanding of how changes in one parameter may influence the interpretation of another. For instance, understanding how variations in umbilical cord pH may correlate with specific Apgar scores contributes to a more comprehensive assessment of fetal distress. This nuanced approach ensures that diagnostic criteria are not isolated but considered in a holistic manner.

3.3 Dataset Size and Imbalance:

3.3.1 Recognition of Dataset Challenges:

The challenges associated with the dataset's small size and class imbalance present significant hurdles in the development of machine learning models geared towards achieving high predictive accuracy. The limited availability of diverse and well-represented samples within the dataset can give rise to biased models that struggle to generalize effectively across a broader spectrum of scenarios.

3.3.2 Expansion Strategies:

The recognition of this gap underscores the pivotal role of data augmentation strategies in mitigating the challenges arising from the dataset's constraints. Data augmentation involves the creation of synthetic samples through various transformations of existing data, thereby enhancing the model's exposure to a more extensive range of patterns and variations.

3.3.3 Comprehensive Dataset Representation:

Expanding the dataset size emerges as a primary strategy for closing the identified gap. However, this expansion is not merely about increasing the quantity of data; it necessitates a focus on incorporating diverse instances that accurately reflect the complexity of real-world scenarios. This comprehensive representation is essential for training a model that can adapt and generalize effectively, ensuring its applicability in various clinical contexts.

3.3.4 Implementing Sophisticated Data Augmentation:

Implementing sophisticated data augmentation techniques becomes integral to achieving a more nuanced and diverse dataset. These techniques may involve transformations such as rotation, scaling, or introducing noise to the existing data, simulating a broader array of conditions. By exposing the model to this augmented dataset, it gains a more robust understanding of potential variations in fetal distress patterns, improving its ability to make accurate classifications.

3.3.5 Pivotal Role in Enhancing Model Capacity:

Effectively addressing the challenges posed by limited dataset size and class imbalance is pivotal for enhancing the model's capacity to generalize and accurately classify instances of fetal distress. A model trained on a more extensive, diverse, and balanced dataset is better equipped to recognize patterns and make informed predictions, contributing to increased diagnostic precision. In the context of fetal distress classification, where the stakes are high, this enhancement in accuracy can lead to more timely and appropriate clinical interventions, ultimately improving patient outcomes.

3.4 Lack of Objective Tools:

The absence of objective tools for fetal distress detection represents a substantial gap in the healthcare practitioner's arsenal. Reliance on subjective interpretations and traditional methods without standardized, quantifiable tools introduces challenges to progress in accurate and consistent diagnosis.

3.4.1 Reliance on Subjective Interpretations:

The historical reliance on subjective interpretations for fetal distress diagnosis has perpetuated a lack of objectivity in clinical assessments. Healthcare practitioners, guided by their experience and expertise, may introduce variability in their evaluations. This subjectivity not only hampers diagnostic consistency but also limits the precision needed for timely and effective interventions.

3.4.2 Impeding Progress in Diagnosis:

The absence of standardized, quantifiable tools impedes progress in accurate and consistent diagnosis. Without objective metrics, healthcare professionals face challenges in precisely characterizing and quantifying fetal distress, potentially leading to delays in appropriate medical responses. Bridging this gap is crucial for improving diagnostic accuracy and, subsequently, patient outcomes.

3.4.3 Development of Objective Tools:

Addressing this gap necessitates the development and integration of reliable, objective tools that leverage advanced technologies, including machine learning. These tools can bring a systematic and data-driven approach to fetal distress diagnosis, introducing a level of precision that transcends subjective interpretations. The integration of machine learning algorithms enables the identification of intricate patterns in large datasets, contributing to more nuanced and accurate diagnoses.

3.4.4 Systematic and Standardized Approach:

The introduction of a more systematic and standardized approach through objective tools is paramount for minimizing diagnostic subjectivity. By employing consistent criteria and quantifiable measures, these tools offer a level of standardization that is essential for effective communication among healthcare professionals. This, in turn, fosters a collaborative and informed decision-making process in the clinical setting.

3.4.5 Empowering Healthcare Professionals:

The integration of reliable and objective tools empowers healthcare professionals with precise, data-driven insights. This empowerment is transformative, as it enables practitioners to base their decisions on a foundation of quantifiable data rather than subjective interpretations. It enhances the overall quality of care by providing clinicians with a more accurate understanding of fetal distress conditions, facilitating targeted interventions and improving patient outcomes.

3.4.6 Integral for Advancing Diagnosis:

Addressing the gap in the lack of objective tools is integral for advancing fetal distress diagnosis. As medical technology evolves, embracing objective tools becomes imperative for staying at the forefront of diagnostic capabilities. The integration of these tools ensures optimal care for expectant mothers and their infants, aligning with the broader goal of enhancing maternal and neonatal healthcare.

3.5 Robustness and Generalization:

Demonstrating the model's robustness and generalization through a high validation accuracy underscores a notable gap in previous approaches to fetal distress classification. The historical lack of models that can consistently perform well across diverse datasets and scenarios has limited the reliability of diagnostic tools.

3.5.1 Historical Lack of Robust Models:

Historically, there has been a lack of models that consistently exhibit robustness and generalization across diverse datasets and scenarios. This deficiency in model performance has posed challenges in providing reliable diagnostic tools for healthcare practitioners, as the ability to adapt to variations in data distribution is paramount for effective clinical utility.

3.5.2 Limitations in Reliability:

The historical lack of robust and generalized models has, in turn, limited the reliability of fetal distress diagnostic tools. Inconsistent performance across different datasets and clinical scenarios can lead to a lack of trust in the accuracy of the models, hindering their practical application in real-world clinical settings.

3.5.3 Necessity for Robust and Generalized Models:

This situation highlights the necessity for more robust and generalized models that can adapt effectively to variations in data distribution. Such models are essential for ensuring consistent and accurate performance in real-world clinical settings, where the diversity of patient populations and healthcare practices demands a high level of adaptability.

3.5.4 Bridging the Gap:

Bridging the identified gap involves refining machine learning algorithms to enhance their adaptability and generalization capabilities. This refinement may include optimizing model architectures, fine-tuning hyperparameters, and incorporating advanced techniques such as transfer learning to leverage knowledge from related domains. The goal is to create models that not only perform well in controlled environments but also maintain their accuracy when faced with the complexities of real-world clinical data.

3.5.5 Incorporating Diverse Datasets:

An essential component of bridging the gap is incorporating diverse datasets into the model training process. Exposure to a wide range of data ensures that the model learns patterns that are representative of different clinical scenarios and patient demographics. This diversity enhances the model's ability to generalize beyond the specifics of the training data.

3.5.6 Comprehensive Validation Strategies:

Deploying comprehensive validation strategies is crucial to ensuring the model's reliability across different contexts. Rigorous validation, including cross-validation and testing on diverse datasets, provides evidence of the model's ability to generalize and maintain robust performance beyond the training environment.

3.5.7 Advancements in Model Development:

Recognizing this gap signals a call for advancements in model development to enhance the practical utility of fetal distress classification models. Continuous innovation in machine learning methodologies, coupled with a commitment to addressing real-world challenges, is essential for closing this gap and ensuring that diagnostic tools meet the high standards required in clinical practice.

CHAPTER-4

PROPOSED METHODOLOGY

Convolutional Neural Network (CNN) stands as the centerpiece of our proposed methodology for intrapartum fetal distress classification. In this chapter, we delve into the intricacies of utilizing CNNs to handle time series data, automate feature extraction, address class imbalance, and optimize the model architecture. Each facet contributes to a comprehensive and effective approach for fetal distress classification based on cardiotocography data.

Fig. 4. Represents the architecture of our proposed methodology.

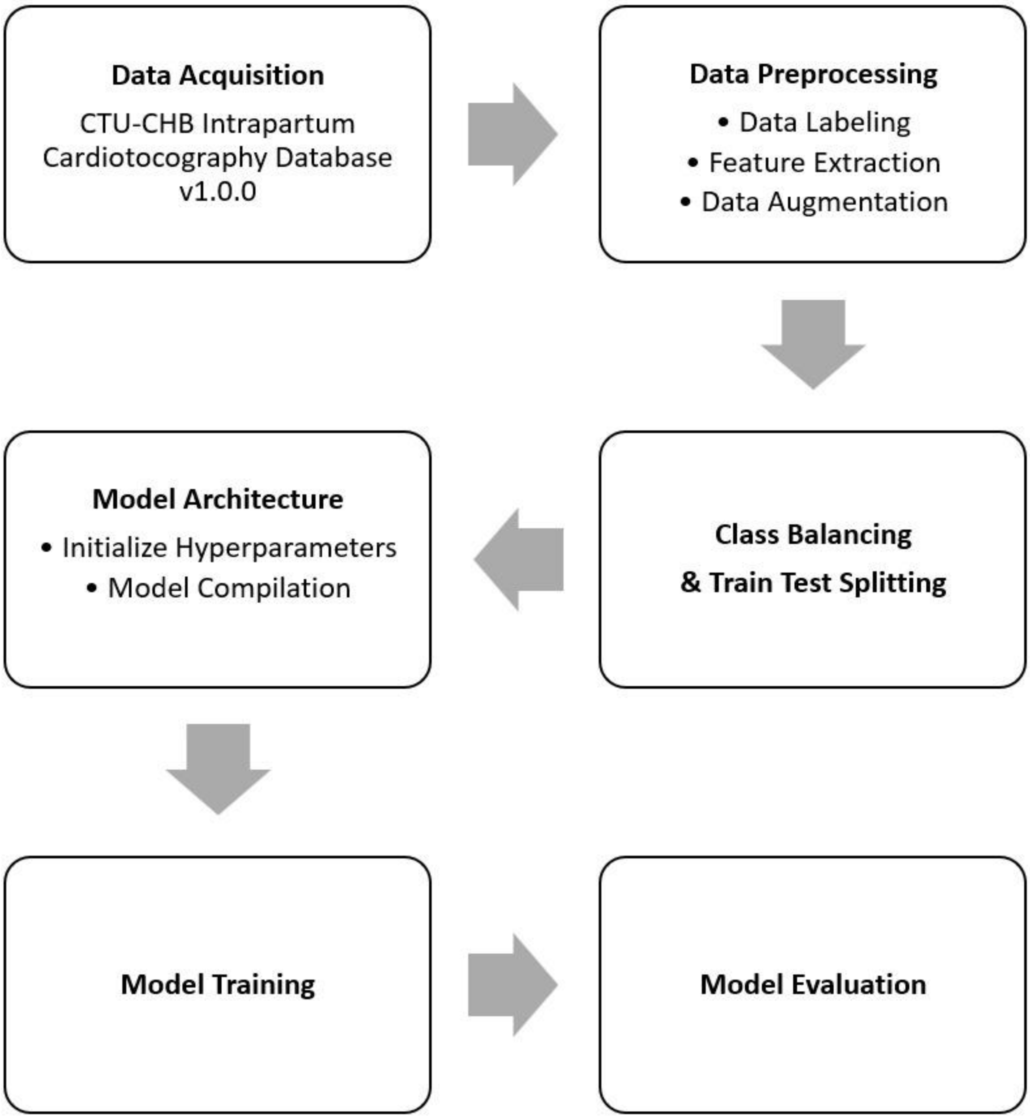


Fig.4.1. Architecture of Proposed Classification Methodology

4.1 Time Series Data Handling:

4.1.1 Significance in Fetal Distress Classification:

CNNs excel in processing time series data, particularly FHR and UC signals. These signals contain complex temporal patterns crucial for fetal distress classification. By leveraging the inherent capabilities of CNNs in capturing sequential dependencies, this approach enhances the model's ability to discern subtle variations indicative of fetal distress.

4.1.2 Adaptive Temporal Pattern Recognition:

The utilization of CNNs enables adaptive temporal pattern recognition, allowing the model to dynamically adjust its understanding of sequential information. This adaptability is crucial for accurately interpreting variations in FHR and UC signals across different stages of labor, contributing to more precise classifications.

4.2 Automated Feature Extraction:

4.2.1 CNN's Inherent Feature Extraction Capability:

CNNs bring automated feature extraction to the forefront of the methodology. Through convolutional layers and hierarchical feature learning, the model autonomously identifies and extracts key features from FHR and UC signals. This automation streamlines the differentiation between 'Distress' and 'Normal' instances, reducing the reliance on manually engineered features.

4.2.2 Enhanced Discriminative Power:

Automated feature extraction enhances the model's discriminative power by focusing on relevant patterns indicative of fetal distress. This empowers the CNN to discern intricate variations in the input signals, contributing to a more nuanced and accurate classification process.

4.3 Data Augmentation and Variability:

4.3.1 Augmentation for Dataset Expansion:

The incorporation of data augmentation techniques plays a pivotal role in dataset expansion. Signal variations introduced through augmentation contribute to a more diverse dataset, preventing the model from becoming overly specialized and enhancing its robustness.

4.3.2 Diversification to Prevent Overfitting:

Data augmentation not only expands the dataset but also introduces signal diversification to prevent overfitting. This diversification is essential for ensuring that the model generalizes well to unseen data, a critical factor in real-world clinical scenarios.

4.4 Addressing Class Imbalance:

4.4.1 Importance of Balanced Datasets:

The methodology addresses class imbalance by balancing 'Normal' and 'Distress' samples. This strategic adjustment improves training dynamics, creating a more equitable dataset distribution that aids the model in effectively learning patterns from both classes.

4.4.2 Equitable Representation for Improved Learning:

Balancing the dataset ensures equitable representation, preventing the model from being skewed towards the majority class. This balanced representation contributes to a more nuanced understanding of both 'Normal' and 'Distress' instances, fostering improved learning outcomes.

4.5 Optimized Architecture and Regularization:

4.5.1 Convolutional Layers and ReLU Activation:

The CNN's architecture is optimized, incorporating convolutional layers and ReLU activation functions. These elements enhance the model's ability to capture hierarchical features within the time series data, facilitating more effective discrimination between different fetal health conditions.

4.5.2 Regularization Techniques:

To prevent overfitting, regularization techniques such as dropout are implemented in the architecture. These techniques contribute to model generalization by introducing controlled randomness during training, enhancing the CNN's ability to perform well on diverse datasets.

CHAPTER-5

OBJECTIVES

In this chapter, we outline the primary and secondary objectives guiding the development of an automated system for the early detection and classification of fetal distress. The overarching goal is to address the inherent subjectivity in traditional diagnosis methods, introducing a more objective and data-driven approach based on machine learning algorithms. By achieving these objectives, we aim to revolutionize fetal distress diagnosis, enhance patient safety, and contribute to the broader healthcare goals of improving maternal and neonatal care.

5.1 Primary Objective:

Development of an Automated System for Fetal Distress Detection

5.1.1 Purpose:

The primary objective centers on the creation of an automated system utilizing machine learning for the early detection and classification of fetal distress using Fetal Heart Rate (FHR) and Uterine Contractions (UC) data. The primary purpose is to significantly reduce subjectivity in fetal distress diagnosis, striving for the highest level of objectivity and accuracy.

5.1.2 Mitigating Subjectivity:

This objective aims to mitigate the inherent subjectivity in traditional fetal distress diagnosis, offering a standardized and data-driven methodology. By leveraging machine learning algorithms, the goal is to establish an automated framework that minimizes the variability introduced by individual healthcare professionals' interpretations.

5.2 Secondary Objectives:

5.2.1 Timely Detection:

A secondary objective focuses on enabling healthcare professionals to detect fetal distress in real-time during pregnancy and labor. Emphasizing timely detection ensures that interventions can be initiated promptly when necessary, addressing the critical need for proactive measures.

5.2.2 Reduce Subjectivity:

Another secondary objective is to minimize the reliance on subjective assessments by providing an objective and data-driven classification of fetal distress. This involves leveraging machine learning models to analyze FHR and UC data objectively, reducing the influence of individual interpretations and enhancing the consistency of diagnoses.

5.2.3 Enhance Maternal and Neonatal Outcomes:

Improving maternal and neonatal healthcare outcomes is a key secondary objective. Early and accurate detection of fetal distress contributes to timely interventions, potentially preventing complications and ensuring the well-being of both the mother and the baby.

5.2.4 Align with Healthcare Goals:

The final secondary objective involves aligning with broader healthcare goals by enhancing patient safety. This includes reducing healthcare costs and improving the overall quality of maternal and neonatal care. The automated system aims to contribute to these overarching healthcare objectives.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 Hardware Configuration:

6.1.1 Experiments Execution:

The model's development and experimentation unfolded on Google Colaboratory, harnessing the computational power of the Tesla T4 GPU runtime environment. Tesla T4, a high-performance GPU, excels in parallel processing tasks and proves optimal for the swift training of deep neural network models, such as the CNN model proposed. The utilization of Google Colab's free Tesla T4 GPU provided substantial resources, boasting 12.7 GB of System RAM, 15 GB of GPU RAM, and approximately 78.2 GB of disk space.

6.1.2 Optimized for Deep Learning:

The Tesla T4 GPU's design is particularly tailored for deep learning tasks, ensuring expedited model training times. This hardware configuration played a pivotal role in the efficient execution of the experiments, facilitating the development and fine-tuning of the proposed CNN model for fetal distress classification.

6.2 Software Environment:

6.2.1 Programming Language:

Python served as the programming language for the development of the fetal distress classification model. This choice facilitated seamless integration with a myriad of data processing and deep neural network libraries, enhancing the model's flexibility and extensibility.

6.2.2 Libraries and Tools:

Several libraries and tools were instrumental in crafting the model. Data loading, preprocessing, and augmentation were executed using various libraries, including standard ones such as os for path definition and NumPy for array operations. The wfdb library played a crucial role in operations on the CTU-CHB database, extracting FHR and UC signals from .dat files and reading contents of .hea files. Data augmentation, a pivotal step in dataset expansion, was achieved through the tsaug library. The scikit-learn library contributed to preprocessing tasks, encoding class labels with LabelEncoder, providing metrics like classification reports and confusion matrices for model evaluation, and facilitating data splitting operations such as the train-test split function.

6.2.3 Deep Learning Frameworks:

TensorFlow and Keras formed the core of the deep learning architecture. TensorFlow was utilized for constructing the CNN architectures and related operations, while Keras served as a high-level neural networks API, offering convenient functions like Conv1D, MaxPooling1D, Flatten, Dense, Dropout, Adam, and BatchNormalization.

6.2.4 Visualization Libraries:

For enhanced visualization of results, Seaborn and Matplotlib were employed. These libraries played a crucial role in presenting the model's performance metrics, aiding in the interpretation and assessment of the proposed CNN model's efficacy.

6.3 Implementation:

In our strategic decision-making for distress classification, the choice of a 1D Convolutional Neural Network (CNN) over traditional machine learning models reflects a deliberate approach to address the unique characteristics of the CTU-CHB intrapartum database. While machine learning models are often favored for smaller datasets, our rationale for opting for a 1D-CNN stems from its inherent ability to excel in scenarios where complex time series data, such as Fetal Heart Rate (FHR) and Uterine Contractions (UC) signals, demand intricate pattern recognition (Fig. 6.1). The critical factor guiding this decision lies in the augmentation and scaling of the dataset. Recognizing the limitations posed by the relatively small dataset, we undertook a comprehensive data augmentation strategy. By substantially expanding the dataset to a sufficiently large number of samples, we aimed to provide the 1D-CNN with ample data to learn from. Deep learning models, and particularly CNNs, exhibit superior performance in scenarios where they can generalize intricate patterns effectively. This strategic augmentation ensures that the model's capacity for pattern recognition remains robust without succumbing to overfitting. In the proposed methodology, each stage is meticulously orchestrated to contribute to the overall efficacy of the distress classification model. Starting with the essential task of labeling samples, we move on to the extraction and preprocessing of signals stored in the .dat files. The subsequent phase involves a critical step in data augmentation, where the dataset is artificially enriched to overcome the constraints of its initial size. Finally, the CNN model is trained, employing a thoughtful selection of hyperparameters and optimization techniques, aimed at maximizing its performance. This staged approach underscores the importance of each step in the implementation process, setting the groundwork for the development of a high-performing and robust distress classification model tailored to the intricacies of the CTU-CHB intrapartum database.

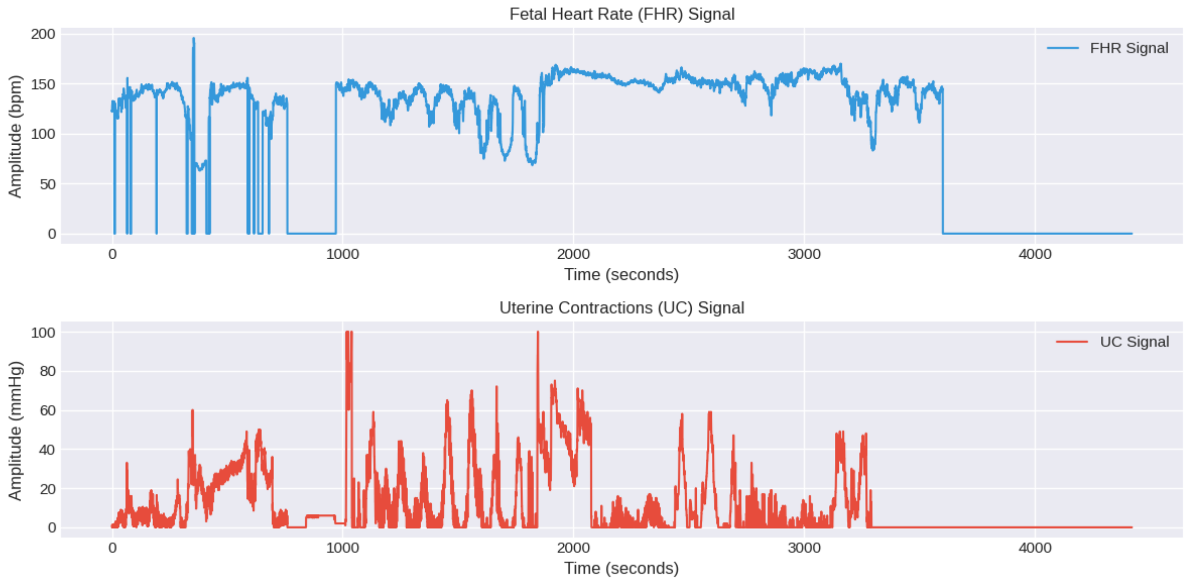


Fig. 6.1. Fetal Heart Rate and Uterine Contractions of a Distressed Sample (pH = 7.01, Apgar1 = 2 and Apgar5 = 4)

6.3.1 Data Labeling:

The initial phase of the implementation process involves the crucial step of assigning labels to the 552 samples within the CTU-CHB database. Given that the dataset arrives unlabeled, our approach is to meticulously categorize each sample based on the most significant parameters indicative of fetal distress. To achieve this, we initiate the labeling process by extracting and analyzing information from the .hea files associated with each sample.

The key parameters selected for distress categorization include the umbilical arterial pH level and Apgar scores at 1 and 5 minutes after birth (Table 6.1). These parameters are recognized as critical physiological markers providing insights into the fetal condition. Our predefined distress thresholds guide the labeling process: if the pH level is below 7.15 (pH < 7.15) or the Apgar scores at 1 or 5 minutes are less than 7 (Apgar1/Apgar5 < 7), the corresponding

Cardiotocography (CTG) recording is labeled as 'Distress.' Conversely, if these criteria are not met, the CTG recording is labeled as 'Normal.'

Sign \ Score	2	1	0
Appearance (skin color)	Pink all over	Pink with Blue extremities	Cyanotic / Pale all over
Pulse (heart rate)	> 100 bpm	< 100 bpm	Absent
Grimace Response (reflex irritability)	Crying, Sneezing, Coughing	Grimacing	No response to stimulation
Activity (muscle tone)	Active	Arms and legs flexed	Absent
Respiration	Good, crying	Slow and irregular, weak, gasping	Absent

Table. 6.1. The Apgar score [7] [9], ranging from 0 to 10, is derived by summation of scores assigned to five neonatal criteria, each scored on a scale of 0 to 2.

This meticulous labeling strategy stems from a well-established justification, focusing on capturing instances where the chosen physiological markers unequivocally signal fetal distress. The stringent criteria applied during the labeling process ensure that the subsequent model training is rooted in a precise and clinically relevant classification of 'Distress' and 'Normal' instances.

As a result of this labeling criteria application, the class distribution within the dataset manifests as follows: 415 samples are labeled as 'Normal,' while the remaining 136 samples are identified as 'Distress' (Fig. 6.2). This class distribution lays the foundation for subsequent stages of the implementation process, contributing to the creation of a well-defined and appropriately labeled dataset for the training of the distress classification model.

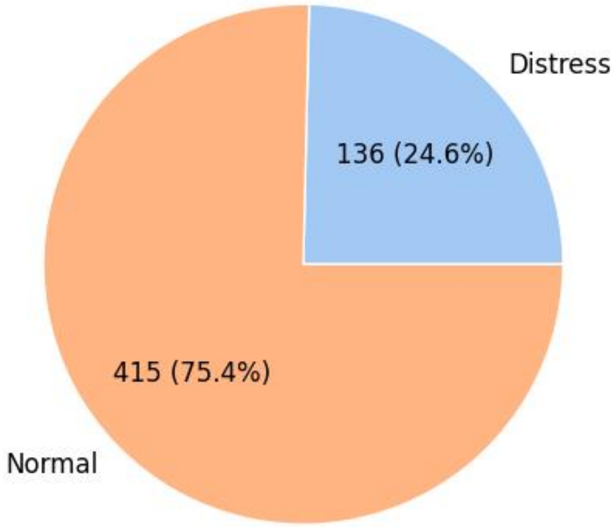


Fig. 6.2. Distribution of Classes after Labeling

6.3.2 Data Preprocessing:

The data preprocessing phase is a pivotal step in preparing the raw signals extracted from the CTU-CHB database for subsequent model training. In this stage, we focus on extracting and refining the Fetal Heart Rate (FHR) and Uterine Contraction (UC) signals, crucial components for accurate fetal distress classification. Leveraging the capabilities of the wfdb Python library, we systematically process each sample to ensure the consistency and homogeneity of the dataset.

The initial step involves the extraction of the FHR and UC signals from the raw data of each sample. The time series nature of the data necessitates resampling to 1000 features for both the FHR and UC signals. This standardization in the length of signals ensures uniformity and facilitates streamlined processing during subsequent stages.

It is imperative to acknowledge that FHR and UC are inherently different in their measurement units, with FHR being quantified in beats per minute and UC representing uterine pressure measured in mmHg. To eliminate potential biases arising from disparate units, we employ Min-Max normalization. This normalization process involves scaling each value in the FHR signal by the maximum FHR value (maxFHR), set at 293.0, and each value in the UC signal by the maximum UC value (maxUC), set at 127.5. Consequently, all values are transformed to a standardized range of [0, 1].

This normalization procedure is iteratively applied to each sample, resulting in the creation of

2000-feature one-dimensional arrays for both FHR and UC. Subsequently, these normalized features are concatenated into a cumulative array for all 552 samples. This comprehensive array, with dimensions (552x2000), signifies the successful completion of the data preprocessing phase.

The meticulous preprocessing of FHR and UC signals ensures that the dataset is well-structured and ready for further analysis and model training. This standardized representation forms the foundation for subsequent stages, contributing to the development of a robust and effective distress classification model.

6.3.3 Data Augmentation:

In addressing the inherent challenge of a relatively small dataset within the CTU-CHB database [1] [2], consisting of only 552 samples, the adoption of data augmentation emerges as a strategic approach to bolster the capacity for effective deep neural network model training, particularly Convolutional Neural Networks (CNNs). Recognizing the limitations posed by the dataset size, we opted for the implementation of data augmentation to artificially amplify the dataset size by generating additional samples based on the existing ones.

To execute the augmentation process, we leveraged the tsaug Python library, a versatile tool designed for time series data augmentation. Specifically, we applied the AddNoise transformation with a scaling factor of 0.01. This transformative technique introduces controlled jittering to the original signals, thereby incorporating subtle variations that contribute to the model's ability to generalize effectively. Figure 3 visually represents the impact of this augmentation, showcasing the discernible variations introduced through the AddNoise transformation.

It is noteworthy that the application of the augmentation transformation is executed with a 90% probability on each sample, ensuring a balanced augmentation strategy. With the overarching goal of constructing a sufficiently large dataset, 20 augmented samples were generated for each original sample, resulting in a substantial increase in the total number of samples for subsequent model training.

Post-augmentation, the dataset underwent a final reshaping process, involving the vertical stacking of the original and augmented data. This strategic restructuring aims to enhance variability and sparsity in the dataset, mitigating the risk of potential biases before the commencement of model training. Additionally, the augmented dataset underwent a shuffling process, further contributing to increased diversity and preventing any systematic patterns from influencing the training process (Fig. 6.3).

Fig. 6.3. Comparison of preprocessed FHR and UC features of a sample with its two most distinctly augmented samples.

Upon the successful completion of the augmentation stage, the total number of samples surged from the initial 552 to 11592, encompassing both the original 552 samples and the newly generated augmented samples as shown in Fig. 6.4.

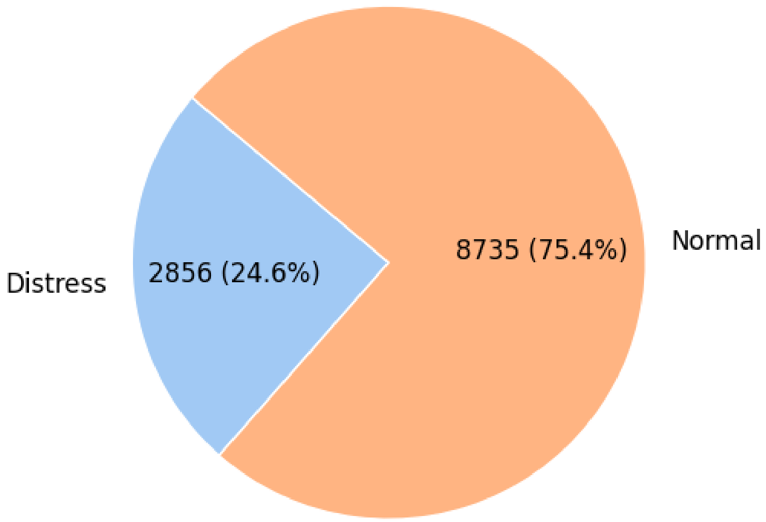


Fig. 6.4. Distribution of Classes (after Augmentation)

This augmentation strategy plays a pivotal role in overcoming the dataset's inherent limitations, empowering the subsequent stages of model development and fostering improved generalization capabilities.

6.3.4 Data Balancing, Splitting & Further Preprocessing:

Following the successful augmentation and expansion of the dataset, a critical consideration in the subsequent stages of the model development process is the inherent imbalance in the class distribution. As illustrated in Figure 5, the 'Distress' samples amount to 2856, a notably smaller subset compared to the 8735 'Normal' samples. Addressing this class imbalance is essential to ensure the effectiveness of model training and to yield improved validation results.

To rectify the class imbalance, a strategic decision was made to select 2856 random 'Normal' samples, aligning with a 50:50 class distribution. This approach draws inspiration from the work of H. Liang et al. [13], aiming to create a balanced dataset that facilitates optimal model training. The resulting balanced dataset comprised a total of 5712 samples, striking a harmonious equilibrium between 'Normal' and 'Distress' classes.

Subsequently, the balanced dataset underwent a meticulous splitting process into training and testing sets, maintaining a 70:30 ratio. This division resulted in 3998 training samples and 1714 validation samples, establishing a comprehensive dataset for the training and evaluation of the CNN model, respectively.

In preparation for model input, a crucial step involved encoding the categorical 'Normal' and 'Distress' labels into numerical values, ensuring compatibility with the CNN model's requirements. Additionally, a reshaping process was implemented for both the training and testing sets. This involved adding a third dimension, transforming the previously two-dimensional matrix into a three-dimensional tensor input. This reshaping ensures the proper alignment of data structures, facilitating the subsequent stages of model architecture definition and training. The thoroughness of these preprocessing steps lays the foundation for a well-structured and balanced dataset, essential for robust model development and evaluation.

6.3.5 Model Architecture and Hyperparameters:

The architecture of the proposed model is meticulously crafted to harness the power of Convolutional Neural Networks (CNNs) for the effective classification of fetal distress. Comprising one-dimensional convolutional layers with 32 and 64 filters of size 3, these layers are pivotal for capturing intricate temporal patterns within the Fetal Heart Rate (FHR) and Uterine Contractions (UC) signals. The activation functions employed are Rectified Linear Units (ReLU), known for their efficiency in learning complex features. To downsample spatial dimensions, one-dimensional Max Pooling over 2-unit windows is applied, contributing to the model's capacity to discern critical features. Table 6.2 summarizes this model architecture and the output shapes after each layer.

Layer	Component	Output Shape
0	Input (Conv1D + ReLU)	(None, 1998, 32)
1	MaxPooling1D	(None, 999, 32)
2	BatchNormalization	(None, 999, 32)
3	Dropout	(None, 999, 32)
4	Conv1D + ReLU	(None, 997, 64)
5	MaxPooling1D	(None, 498, 64)
6	BatchNormalization	(None, 498, 64)
7	Dropout	(None, 498, 64)
8	Flatten	(None, 31872)
9	Dense + ReLU	(None, 64)
10	Dropout	(None, 64)
11	Dense + Sigmoid	(None, 1)
Total params		2,046,657
Trainable params		2,046,465
Non-trainable params		192

Table. 6.2. Model Architecture

The integration of Batch Normalization enhances training stability and expedites convergence. Notably, Dropout layers are strategically incorporated after each batch normalization, mitigating overfitting by randomly deactivating 50% of neurons during training. The subsequent flattening operation transforms the output into a one-dimensional vector.

To further counteract overfitting, L2 regularization is applied with a penalty of 0.01 to the kernel weights. Additionally, the dropout rate is increased to 0.7 before the second fully connected layer, reinforcing the model's ability to generalize effectively.

The model is compiled using the Adam optimizer, emphasizing a dynamic learning rate configuration. The initial learning rate is set to 1.11x10^(-4), recognizing the necessity for adaptability throughout training. The implementation of a dynamic learning rate, facilitated by TensorFlow's callback mechanism, monitors the validation loss. If no improvement is observed for two consecutive epochs, the learning rate is adjusted by a factor of 0.5, ensuring optimal convergence.

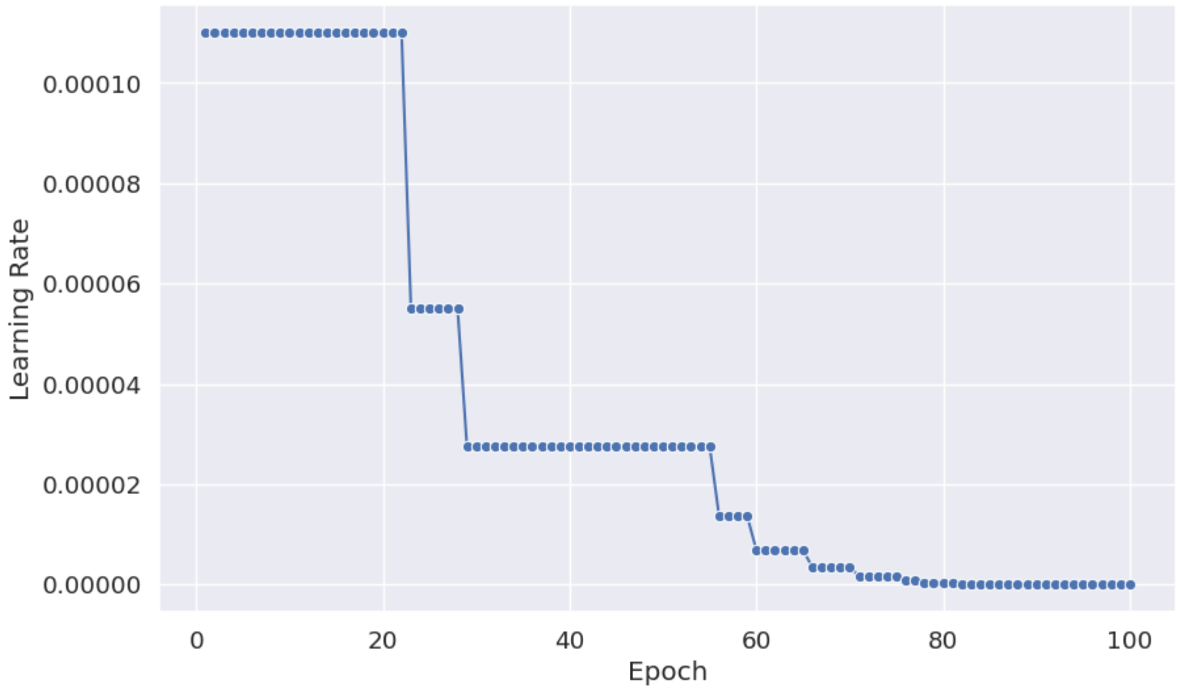


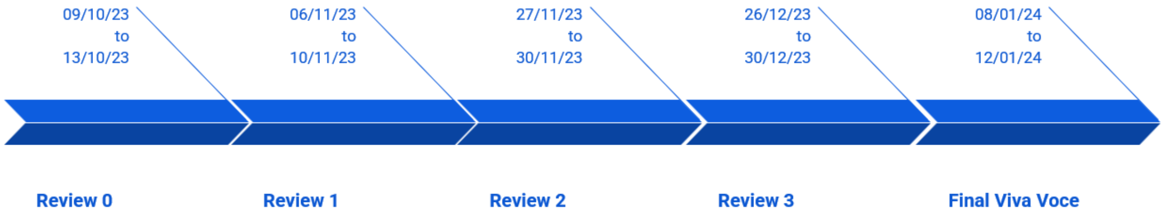
Fig. 6.5. Learning Rate over Epochs

With a focus on hyperparameter initialization, a batch size of 64 is chosen to optimize computational efficiency. The model undergoes training for a total of 100 epochs, reflecting a balance between achieving convergence and avoiding potential overfitting. This meticulous selection of architectural elements and hyperparameters underscores the thoughtful design of the CNN model, poised for effective fetal distress classification.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)



CHAPTER-8
OUTCOMES

The outcomes of the proposed 1D-CNN model for fetal distress classification demonstrate its efficacy in leveraging Fetal Heart Rate (FHR) and Uterine Contractions (UC) signals. Rigorous evaluation metrics and analyses underscore the model's performance, generalization, and robustness, providing valuable insights into its practical utility.

8.1. High Validation Performance:

The validation phase of the 1D-CNN model yielded outstanding results, showcasing its exceptional ability to accurately classify instances of fetal distress. The model achieved an impressive accuracy rate of 98.3%, a testament to its proficiency in effectively utilizing features extracted from Fetal Heart Rate (FHR) and Uterine Contractions (UC) signals for precise classification. The concurrent low validation loss, quantified at 0.2613, further accentuates the model's efficacy.

The high accuracy reflects the model's capability to correctly identify and differentiate between normal and distress cases, indicating a robust learning process. The low validation loss, a measure of the dissimilarity between predicted and actual values, underscores the model's accuracy in predicting the outcomes of previously unseen data. This combination of high accuracy and low loss signifies the model's discriminative power and generalization to diverse instances of fetal distress.

The success of the 1D-CNN model in achieving these metrics not only highlights its effectiveness in accurate classification but also suggests its potential for reliable application in clinical settings. The model's ability to leverage complex temporal patterns within FHR and UC signals emphasizes its utility as a valuable tool for enhancing fetal distress diagnosis, contributing to improved maternal and neonatal care outcomes.

The proposed CNN model was trained for 100 epochs and exhibited promising results in our fetal distress classification task. One of the reasons for concatenating the resampled FHR and UC signal features was to qualitatively study and understand if an inherent correlation exists between them, and in turn identifying if the model can capture it to produce an accurate classification of a sample. To justify this, we obtained an exceptionally high validation accuracy of 98.3%, and a smoothly converging model with a final validation loss of 0.2613

(Fig. 8.1, 8.2).

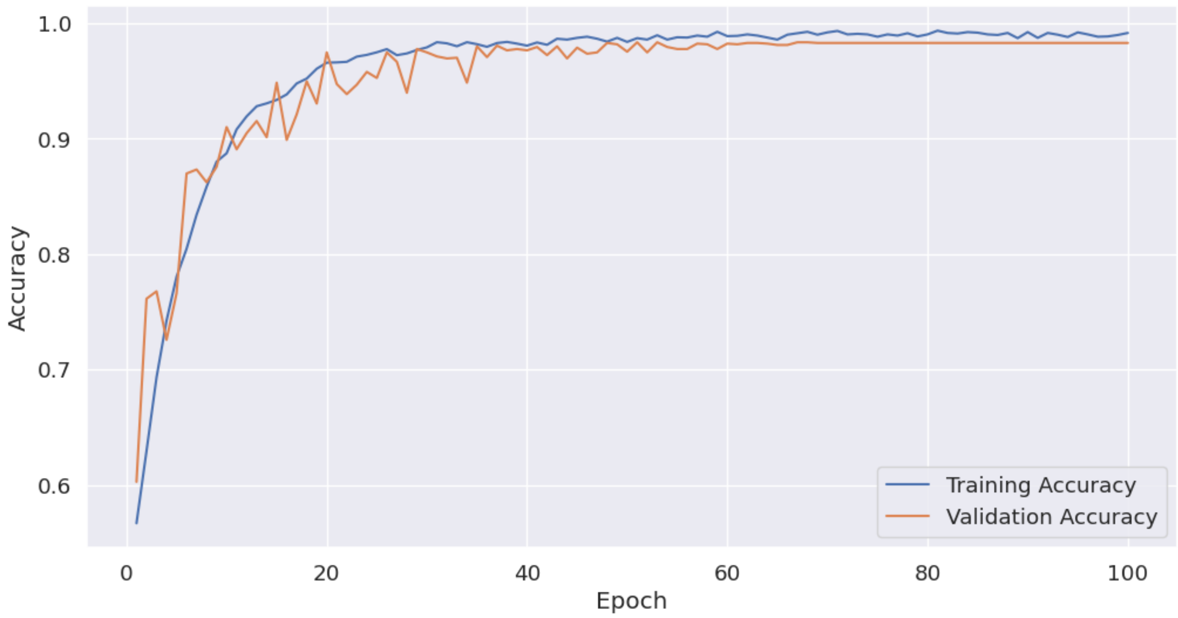


Fig. 8.1. Training and Validation Accuracy

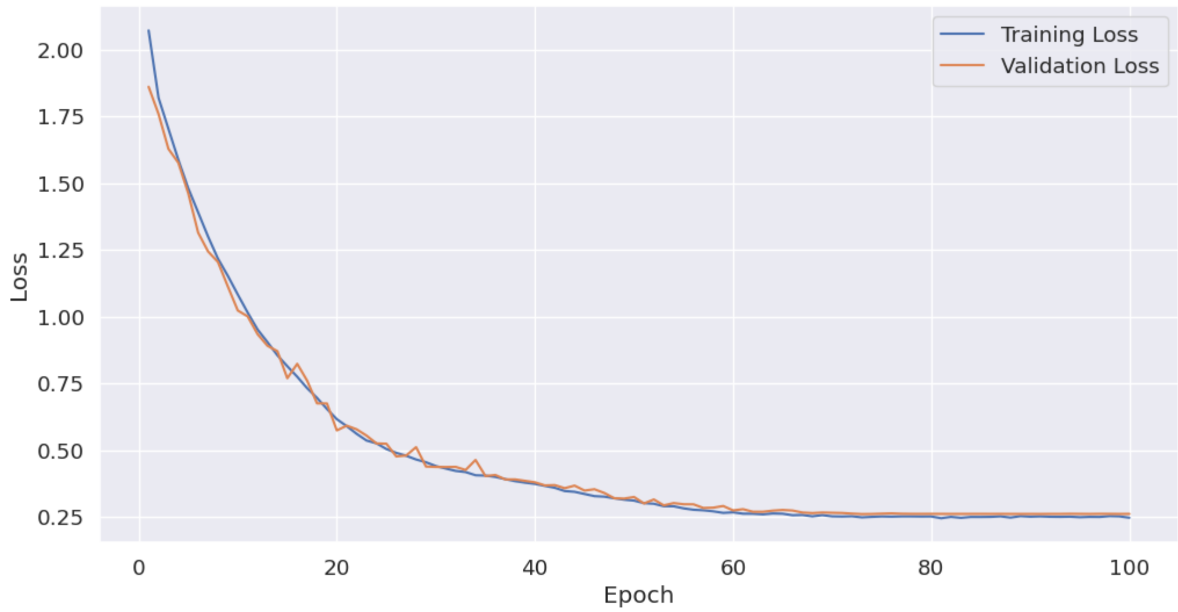


Fig. 8.2. Training and Validation Loss

8.2 No Overfitting:

The training dynamics of the 1D-CNN model provide valuable insights into its robustness and avoidance of overfitting, a critical aspect of model performance.

During the training process, the model exhibited a steady increase in accuracy, reaching a plateau at around 99%. This gradual ascent indicates the model's capacity to learn from the training data, continually improving its ability to classify instances of fetal distress. Importantly, the plateauing of accuracy at less than 100% is a key observation. This phenomenon signifies that the model refrained from memorizing the intricacies of the training data, avoiding overfitting.

Overfitting occurs when a model becomes too tailored to the training data, capturing noise and idiosyncrasies that may not generalize well to new, unseen data. In this case, the model's decision to plateau before complete convergence to 100% accuracy demonstrates its ability to generalize effectively to new instances beyond the training set.

The trends observed in both training and validation losses further support the absence of overfitting. The model's ability to minimize both training and validation losses in tandem indicates its adaptability to complex patterns present in the data without compromising its performance on new and unseen instances.

This robustness against overfitting enhances the model's reliability and applicability in real-world scenarios. The avoidance of overfitting ensures that the model's learned features are representative of genuine patterns in fetal distress cases, contributing to its overall effectiveness as a diagnostic tool in clinical settings.

8.3 Robustness and Generalization:

The robustness and generalization capabilities of the 1D-CNN model are thoroughly examined through a meticulous analysis of the confusion matrix, specifically focusing on the validation dataset consisting of 1714 samples. This in-depth evaluation provides crucial insights into the model's adaptability and reliability in real-world scenarios.

The confusion matrix serves as a powerful tool for assessing the model's predictive performance, offering a detailed breakdown of its classification outcomes. In the context of fetal distress diagnosis, the matrix provides a visual representation of the model's ability to accurately categorize instances across different classes, notably 'Distress' and 'Normal.'

The model's outstanding performance on the validation dataset, comprising previously unseen samples, is indicative of its robustness. Achieving accuracy in diverse and unfamiliar scenarios demonstrates the model's capacity to generalize well beyond the training data. This generalization is crucial for ensuring the model's practical utility in clinical settings, where it may encounter a wide spectrum of fetal distress cases. On analyzing the confusion matrix (Fig 10) of the model on the validation data (1714 samples), we can also conclude that the model generalized well to unseen validation data. These results reflect the robustness and consistency of the model that we have developed.

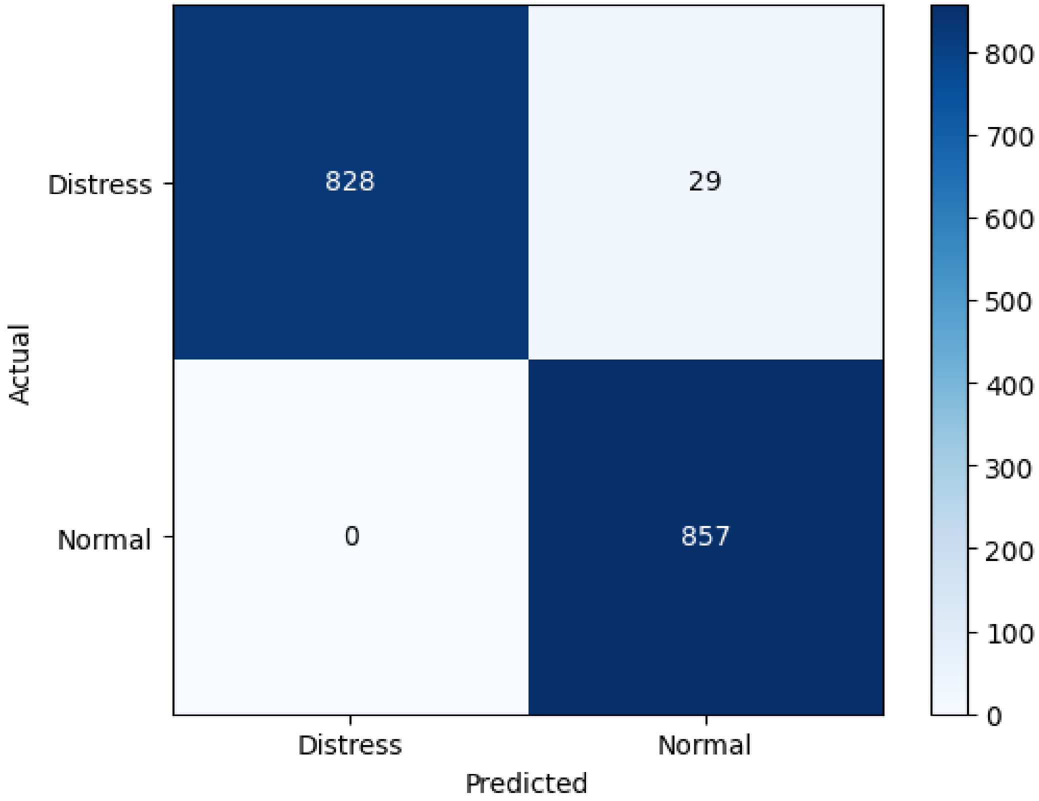


Fig. 8.3. Confusion Matrix

The confusion matrix not only highlights the model's accuracy but also reveals its proficiency in distinguishing between various classes, minimizing false positives and false negatives. This nuanced understanding of the model's behavior enhances its reliability in making critical diagnostic decisions.

These outcomes collectively affirm the model's robustness and generalization, reinforcing its practical applicability in real-world clinical contexts. The ability to perform effectively on previously unseen data positions the developed 1D-CNN model as a promising tool for improving the accuracy and objectivity of fetal distress diagnosis, ultimately contributing to enhanced maternal and neonatal care.

CHAPTER-9
RESULTS AND DISCUSSIONS

9.1 Results:

The culmination of our extensive training efforts, spanning 100 epochs, yielded highly promising outcomes for the fetal distress classification task using the proposed 1D-CNN model. This innovative approach harnessed the distinctive features derived from resampled Fetal Heart Rate (FHR) and Uterine Contractions (UC) signals. Our intentional concatenation of these features was driven by the pursuit of unraveling potential inherent correlations that significantly contribute to the accurate classification of fetal distress. This strategic decision exemplifies the depth of our investigative approach, aiming not only for predictive accuracy but also for a comprehensive understanding of the interplay between different physiological signals.

During the validation phase, the model demonstrated exceptional precision, achieving an impressive accuracy of 98.3%. This metric signifies the model's adeptness in distinguishing between 'Distress' and 'Normal' instances, a critical aspect in ensuring reliable clinical outcomes. The complementing validation loss of 0.2613 further underscores the model's proficiency in making accurate predictions while maintaining a high level of generalization. The validation loss serves as a crucial indicator of how well the model extrapolates its learning to previously unseen data, and the low value obtained reflects the robustness of the model.

Analyzing the training accuracy revealed a steady upward trajectory throughout the epochs, plateauing at around 99%. Importantly, the intentional design choice to prevent the model from reaching 100% accuracy during training was a proactive measure against overfitting. Overfitting occurs when a model memorizes the training data but fails to generalize well to new, unseen data. By maintaining a slight gap between the training and validation accuracies, our model demonstrated an ability to generalize effectively without merely memorizing the training set.

The consistent trends observed in both training and validation losses further validate the model's generalization capabilities. The model learned intricate patterns from the training data without compromising its ability to adapt to diverse scenarios. This balanced performance across training and validation sets is indicative of the model's robustness and reliability, essential qualities for its practical application in clinical settings.

The Confusion Matrix (Fig. 8.3) serves as a pivotal visual tool providing detailed insights into the model's performance during the validation phase, specifically on a dataset comprising 1714 samples. This matrix categorizes the predictions made by the model into four distinct categories: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Each cell in the matrix represents the count of instances falling into these categories.

- **True Positive (TP):** This represents the number of instances correctly classified as 'Distress.' In the context of fetal distress detection, a higher TP count indicates the model's effectiveness in correctly identifying cases when distress is present.
- **True Negative (TN):** These are instances accurately classified as 'Normal.' A higher TN count demonstrates the model's proficiency in correctly identifying situations when fetal distress is absent.
- **False Positive (FP):** Instances incorrectly classified as 'Distress' when they are, in fact, 'Normal.' A lower FP count is desirable, as it signifies fewer instances of misclassification, which is crucial for ensuring the reliability of the model.
- **False Negative (FN):** Instances misclassified as 'Normal' when they are actually cases of 'Distress.' Minimizing FN is essential, as it reflects the model's ability to avoid overlooking instances of fetal distress, crucial for patient safety.

The confusion matrix provides a quantitative breakdown of these categories, enabling a nuanced evaluation of the model's strengths and areas for improvement. The goal is to have a high count in the diagonal elements (TP and TN) and low counts in the off-diagonal elements (FP and FN).

In our specific scenario, the confusion matrix visually communicates how well the 1D-CNN model navigated the complexity of fetal distress classification. The impressive validation accuracy of 98.3% aligns with a corresponding Confusion Matrix that substantiates the model's ability to make accurate predictions across both 'Distress' and 'Normal' classes. This analysis is paramount in affirming the model's robustness and generalization, reinforcing its adaptability to diverse clinical scenarios and its potential applicability in real-world healthcare settings.

9.2 Discussions:

In conclusion, the culmination of our research endeavors underscores the potential breakthrough presented by the developed 1D-CNN model in intrapartum fetal distress classification, leveraging the intricate information embedded in Fetal Heart Rate (FHR) and Uterine Contractions (UC) signals. The model's high validation accuracy, standing at an impressive 98.3%, coupled with an optimally low loss of 0.2613, manifests its capacity to be a valuable asset for healthcare professionals in accurately identifying cases of fetal distress. This achievement serves as a beacon for the application of machine learning approaches, particularly deep learning, in advancing the detection and diagnosis of fetal distress, thereby contributing significantly to ongoing efforts in the field.

A notable merit of the model lies in the objectivity it introduces, mitigating the inherent subjectivity associated with the manual interpretation of Cardiotocography (CTG) data. The model's ability to discern patterns and make predictions based on quantitative features extracted from FHR and UC signals provides a more consistent and unbiased approach, potentially revolutionizing the interpretation process.

While the attained results are promising, our vision extends beyond the current success, and we acknowledge the ample scope for improvement from an applicative standpoint. In future endeavors, we aspire to undertake clinical validation studies to assess the real-world efficacy and reliability of our model. This involves collaborating with healthcare professionals to evaluate its performance in diverse clinical settings and varying patient populations, ensuring its robustness and generalizability.

Further exploration into various facets is planned for future work. We aim to develop user-friendly software that incorporates our CNN model, facilitating medical professionals in objectively validating their CTG interpretations. This software could serve as an additional layer of analysis, offering insights and supporting clinical decision-making. Additionally, we intend to delve into advanced methodologies of data preprocessing for CTG signal data, refining and optimizing the feature extraction process to enhance the model's performance further.

Our aspirations also extend to exploring improved methods of data augmentation, as an enhanced dataset quality directly influences the model's training and subsequent predictive capabilities. By refining augmentation techniques, we aim to ensure the model's resilience and accuracy across a broader spectrum of scenarios and variations in fetal distress manifestations.

CHAPTER-10
CONCLUSION

In the culmination of our extensive research endeavors, the 1D-CNN model devised for intrapartum fetal distress classification has proven to be exceptionally effective, leveraging the nuanced features extracted from resampled Fetal Heart Rate (FHR) and Uterine Contractions (UC) signals. The intentional fusion of these features was not solely geared towards achieving a high classification accuracy but was also designed to unravel potential inherent correlations, thereby adding depth and comprehensiveness to our investigative approach.

The validation phase served as a critical validation of the model's robustness, revealing a remarkable accuracy of 98.3% accompanied by a validation loss of 0.2613. These metrics underline the model's precision in differentiating between instances of 'Distress' and 'Normal,' which is paramount for accurate clinical diagnoses. Noteworthy is the model's strategic approach during training, where it exhibited a steady increase in accuracy throughout epochs, reaching approximately 99%. Importantly, the deliberate avoidance of complete convergence to 100% is a prudent measure to prevent overfitting, affirming the model's ability to generalize effectively without sacrificing its capacity to discern intricate patterns.

Further bolstering the model's credibility is the synchronization of trends between training and validation losses. This congruence signifies the model's consistent learning trajectory, reinforcing its robustness in handling unseen data. The in-depth analysis of the confusion matrix on the validation dataset provides a nuanced understanding of the model's performance. This visual representation underscores the model's adaptability to diverse scenarios, showcasing its potential for seamless integration into real-world clinical applications.

While the current results indeed present a promising breakthrough, our research journey reveals new avenues for future exploration and refinement. The paramount objective of clinically validating our model and integrating it as a real-time tool underscores its potential practical impact in healthcare settings. Our aspirations extend beyond model deployment, aiming to develop software that collaborates harmoniously with medical professionals, augmenting the objectivity and reliability of Cardiotocography (CTG) interpretations. Additionally, our commitment to excellence is reflected in our ongoing efforts to explore advanced preprocessing techniques for CTG signal data, ensuring a more refined and accurate input for the model. Simultaneously, the refinement of methodologies for data augmentation remains an integral aspect of our future work, ensuring that the model continues to evolve and adapt to increasingly complex clinical scenarios. In essence, our research not only marks a significant stride in perinatal care but also lays the foundation for continuous innovation and improvement in leveraging artificial intelligence to address critical healthcare challenges.

REFERENCES

1. V. Chuda´cek, J. Spilka, M. Bur ˇ sa, P. Jank, L. Hruban, M. Huptych, and ˇ L. Lhotska, "Open access intrapartum ctg database," ´ BMC pregnancy and childbirth, vol. 14, pp. 1–12, 2014.
2. A. L. Goldberger et al., "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," Circulation [Online], vol. 101, no. 23, pp. e215–e220, 2000.
3. J. Parer and E. Livingston, "What is fetal distress?" American journal of obstetrics and gynecology, vol. 162, no. 6, pp. 1421–1427, 1990.
4. D. Ayres-de Campos, "Electronic fetal monitoring or cardiotocography, 50 years later: what's in a name?" American Journal of Obstetrics & Gynecology, vol. 218, no. 6, pp. 545–546, 2018.
5. E. H. Hon, "The electronic evaluation of the fetal heart rate: Preliminary report," American Journal of Obstetrics & Gynecology, vol. 75, no. 6, pp. 1215–1230, 1958.
6. R. E. Bailey, "Intrapartum fetal monitoring," American family physician, vol. 80, no. 12, pp. 1388–1396, 2009.
7. J. A. Spencer, "Clinical overview of cardiotocography," BJOG: An International Journal of Obstetrics & Gynaecology, vol. 100, pp. 4– 7, 1993.
8. J. Ogasawara, S. Ikenoue, H. Yamamoto, M. Sato, Y. Kasuga, Y. Mitsukura, Y. Ikegaya, M. Yasui, M. Tanaka, and D. Ochiai, "Deep neural network-based classification of cardiotocograms outperformed conventional algorithms," Scientific reports, vol. 11, no. 1, p. 13367, 2021.
9. V. Apgar, "A proposal for a new method of evaluation of the newborn infant." Anesthesia & Analgesia, vol. 32, no. 4, pp. 260–267, 1953.
10. L. S. James, I. Weisbrot, C. Prince, D. Holaday, and V. Apgar, "The acid-base status of human infants in relation to birth asphyxia and the onset of respiration," The Journal of pediatrics, vol. 52, no. 4, pp. 379– 394, 1958.
11. L. V. Simon, M. F. Hashmi, and B. N. Bragg, "Apgar score," 2017.
12. Y. D. Daydulo, B. L. Thamineni, H. K. Dasari, and G. T. Aboye, "Deep learning based fetal distress detection from time frequency representation of cardiotocogram signal using morse wavelet: research study," BMC Medical Informatics and Decision Making, vol. 22, no. 1, p. 329, 2022.

13. H. Liang, Y. Lu, Q. Liu, and X. Fu, “Fully automatic classification of cardiotocographic signals with 1d-cnn and bi-directional gru,” in 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2022, pp. 4590–4594.

14. M. O’Sullivan, T. Gabruseva, G. B. Boylan, M. O’Riordan, G. Lightbody, and W. Marnane, “Classification of fetal compromise during labour: signal processing and feature engineering of the cardiotocograph,” in 2021 29th European Signal Processing Conference (EUSIPCO). IEEE, 2021, pp. 1331–1335.

15. A. Johny and K. Madhusoodanan, “Dynamic learning rate in deep cnn model for metastasis detection and classification of histopathology images,” Computational and Mathematical Methods in Medicine, vol. 2021, pp. 1–13, 2021.

16. L. N. Smith, “Cyclical learning rates for training neural networks,” in 2017 IEEE winter conference on applications of computer vision (WACV). IEEE, 2017, pp. 464–472.

APPENDIX-A

PSUEDOCODE

```
# Import necessary libraries
import os, numpy, wfdb, tensorflow, keras, matplotlib, seaborn, pandas, tsaug

# Mount Google Drive
drive.mount('/content/drive')

# Define data directory and retrieve .dat files
data_directory = '/content/drive/MyDrive/ctu-chb-intrapartum-cardiotocography-database-
1.0.0'
dat_files = list files in data_directory with extension '.dat'

# Define parameter thresholds for distress
parameter_thresholds = {"pH": (7.15, 9999), "Apgar1": (7, 9999), "Apgar5": (7, 9999)}

# Initialize empty lists for labels and preprocessed signals
labels = []
X_signals = []

# Process each .dat file
for each dat_file in dat_files:
    # Extract record name and load record
    record_name = remove extension from dat_file
    record = load record from data_directory with record_name

    # Check distress criteria based on parameter thresholds
    distress_flag = check_distress_criteria(record, parameter_thresholds)

    # Assign label based on distress flag
    label = "Distress" if distress_flag else "Normal"
    labels.append(label)

    # Extract and preprocess FHR and UC signals
    fhr_signal, uc_signal = extract_and_preprocess_signals(record)

    # Combine signals into a feature vector
    combined_signal = concatenate(fhr_signal, uc_signal)
    X_signals.append(combined_signal)

# Convert lists to NumPy arrays
original_labels = convert labels to NumPy array
X_signals = convert X_signals to NumPy array
```

```
# Augment data using a specified augmentation pipeline
augmented_X_signals, augmented_labels = augment_data(X_signals, original_labels)

# Visualize original and augmented signals
plot_signals(X_signals, augmented_X_signals)

# Prepare a balanced dataset by oversampling the minority class
X_balanced, labels_balanced = prepare_balanced_dataset(augmented_X_signals,
augmented_labels)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = split_dataset(X_balanced, labels_balanced)

# Encode labels using LabelEncoder
y_train_encoded, y_test_encoded = encode_labels(y_train, y_test)

# Reshape data for CNN input
X_train_cnn, X_test_cnn = reshape_for_cnn(X_train, X_test)

# Define CNN model architecture and compile it
cnn_model = define_and_compile_cnn_model()

# Train the CNN model with dynamic learning rate adjustment
training_history = train_model_with_dynamic_lr(cnn_model, X_train_cnn,
y_train_encoded, X_test_cnn, y_test_encoded)

# Save the trained model
save_model(cnn_model, '/content/drive/MyDrive/fetaldistress.h5')

# Evaluate the model on the test set
predicted_probabilities, predicted_labels = evaluate_model(cnn_model, X_test_cnn)

# Visualize learning rate changes over epochs
plot_learning_rate_changes(training_history)

# Visualize training history for accuracy and loss
plot_training_history(training_history)

# Display confusion matrix and classification report
display_confusion_matrix_and_classification_report(y_test_encoded, predicted_labels)
```

APPENDIX-B

SCREENSHOTS

```
!pip install numpy matplotlib wfdb ipywidgets tsaug --quiet


#Importing libraries
import os
import numpy as np
import wfdb
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import tensorflow as tf
from keras import layers, models, Model
from keras.regularizers import l2
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,
ConfusionMatrixDisplay
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.signal import resample
from scipy.spatial.distance import euclidean
from keras.optimizers import Adam
from sklearn.metrics import f1_score
from ipywidgets import interact, widgets
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense, Dropout, Input, Conv1D,
BatchNormalization, Activation, Add, GlobalAveragePooling1D, Dense, MaxPooling1D, Dropout
from keras.optimizers import Adam
import tsaug


#Mount Google Drive
from google.colab import drive
drive.mount(                )

# Define the directory where the data is stored
data_directory =

# List all .dat files in the directory
dat_files = [f for f in os.listdir(data_directory) if f.endswith(      )]

parameter_thresholds = {
        : (7.15,9999),
            : (7, 9999),
            : (7, 9999)
}
```

```
#Visualizing a sample (most distressed case)

# Specify the record name
record_name = "2024"  # Replace with the desired record name

# Load the record
record = wfdb.rdrecord(os.path.join(data_directory, record_name))

# Extract FHR and UC signals
fhr_signal = record.p_signal[:, record.sig_name.index('FHR')]
uc_signal = record.p_signal[:, record.sig_name.index('UC')]

# Print information about the record
print("Record Information:")
print("Signals:", record.sig_name)
print("Units:", record.units)

# Get the metadata to obtain the correct units
fhr_units = record.units[record.sig_name.index('FHR')]
uc_units = record.units[record.sig_name.index('UC')]

# Calculate the time vector based on the sampling frequency
time_vector = (1 / record.fs) * np.arange(len(fhr_signal))

# Set up a clean and professional plot with a professional color scheme
plt.style.use('seaborn-darkgrid')
plt.figure(figsize=(12, 6))

# Define professional colors
fhr_color = '#3498db'  # Blue
uc_color = '#e74c3c'   # Red

# Plot FHR signal
plt.subplot(2, 1, 1)
plt.plot(time_vector, fhr_signal, color=fhr_color, label='FHR Signal', linewidth=1.5)
plt.title('Fetal Heart Rate (FHR) Signal')
plt.xlabel('Time (seconds)')
plt.ylabel('Amplitude ({})'.format(fhr_units))
plt.legend()

# Plot UC signal
plt.subplot(2, 1, 2)
plt.plot(time_vector, uc_signal, color=uc_color, label='UC Signal', linewidth=1.5)
plt.title('Uterine Contractions (UC) Signal')
plt.xlabel('Time (seconds)')
plt.ylabel('Amplitude (mmHg)')
plt.legend()

# Fine-tune layout for a professional look
plt.tight_layout()

# Display the plot
plt.show()



# Initialize lists to corresponding labels
labels = []

# Process the uploaded .dat files and assign labels based on parameter thresholds
for dat_file in dat_files:
    record_name = os.path.splitext(dat_file)[0]  # Remove the .dat extension
    record = wfdb.rdrecord(os.path.join(data_directory, record_name))

    # Load the corresponding .hea file to access header information
    hea_file_path = os.path.join(data_directory, record_name + '.hea')
    with open(hea_file_path, 'r') as hea_file:
        hea_content = hea_file.read()

    # Check distress criteria based on thresholds
    distress_flag = False
    for param, (low, high) in parameter_thresholds.items():
        param_value = float(hea_content.split(f"#{param}")[-1].split()[0])
        if param_value < low:
          distress_flag = True
          break

    # Assign labels based on distress flag
    label = "Distress" if distress_flag else "Normal"
    labels.append(label)
```

```
# Convert binary labels to original format ("Normal" and "Distress")
original_labels = np.array(labels)

# Calculate label distribution
unique_labels, label_counts = np.unique(original_labels, return_counts=True)

# Create a pie chart
fig, ax = plt.subplots()
ax.pie(label_counts, labels=unique_labels, autopct='%1.1f%%', startangle=90)
ax.axis('equal')

plt.title('Distribution of Data')

plt.show()


# Initialize variables to store maximum values
max_fhr_bpm = 0
max_uc_value = 0

for dat_file in dat_files:
    record_name = os.path.splitext(dat_file)[0]  # Remove the .dat extension
    record = wfdb.rdrecord(os.path.join(data_directory, record_name))

    # Extract FHR and UC signals
    fhr_signal = record.p_signal[:, record.sig_name.index('FHR')]
    uc_signal = record.p_signal[:, record.sig_name.index('UC')]

    # Find maximum FHR BPM and UC value in the current record
    max_fhr_bpm = max(max_fhr_bpm, np.max(fhr_signal))
    max_uc_value = max(max_uc_value, np.max(uc_signal))

# Initialize an empty list to store preprocessed signals
X_signals = []

for dat_file in dat_files:
    record_name = os.path.splitext(dat_file)[0]  # Remove the .dat extension
    record = wfdb.rdrecord(os.path.join(data_directory, record_name))

    # Extract FHR and UC signals
    fhr_signal = record.p_signal[:, record.sig_name.index('FHR')]
    uc_signal = record.p_signal[:, record.sig_name.index('UC')]

    # Resample signals to a common length
    common_length = 1000
    fhr_signal_resampled = resample(fhr_signal, common_length)
    uc_signal_resampled = resample(uc_signal, common_length)

    # Normalize FHR signal
    fhr_signal_resampled /= max_fhr_bpm

    # Normalize UC signal
    uc_signal_resampled /= max_uc_value

    # Combine FHR and UC signals into a single feature vector
    combined_signal = np.concatenate((fhr_signal_resampled, uc_signal_resampled))

    # Append the preprocessed signal to the list
    X_signals.append(combined_signal)

# Convert the list to a NumPy array for further processing
X_signals = np.array(X_signals)

print(max_fhr_bpm)
print(max_uc_value)


# Convert features and labels lists to NumPy arrays
X_signals = np.array(X_signals)
labels = np.array(labels)
X_signals.shape
labels.shape
```

```
# Define the data augmentation pipeline
augmenter = (
    tsaug.AddNoise(scale=0.01) @ 0.9  # with 90% probability, introduce random jittering
)

# Increase the number of augmentations per sample to achieve a larger dataset
num_augmentations_per_sample = 20  # Adjust this number as needed

# Apply data augmentation to each sample in X_signals
augmented_X_signals = []
augmented_labels = []

for i in range(X_signals.shape[0]):
    original_signal = X_signals[i, :]

    # Apply augmentation to the original signal multiple times
    for _ in range(num_augmentations_per_sample):
        augmented_signal = augmenter.augment(original_signal)
        augmented_X_signals.append(augmented_signal)
        augmented_labels.append(labels[i])

# Reshape augmented data to match the original data shape
augmented_X_signals_reshaped = np.array(augmented_X_signals).reshape(-1, X_signals.shape[1])

# Concatenate original and augmented data
X_signals_augmented = np.vstack((X_signals, augmented_X_signals_reshaped))
labels_augmented = np.concatenate((labels, augmented_labels))

# Shuffle the augmented dataset
shuffle_indices = np.random.permutation(X_signals_augmented.shape[0])
X_signals_augmented = X_signals_augmented[shuffle_indices]
labels_augmented = labels_augmented[shuffle_indices]



# Select a specific sample index (wrap around if out of bounds)
sample_index = 2018 % X_signals.shape[0]

original_sample = X_signals[sample_index, :]
augmented_samples = augmented_X_signals_reshaped[
    sample_index * num_augmentations_per_sample : (sample_index + 1) * num_augmentations_per_sample, :
]

# Calculate Euclidean distances between original and augmented signals
distances = [euclidean(original_sample, augmented_samples[i, :]) for i in
range(num_augmentations_per_sample)]

# Select the indices of the most distinct signals (e.g., top 3)
most_distinct_indices = sorted(range(num_augmentations_per_sample), key=lambda i: distances[i],
reverse=True)[:2]

# Set Seaborn style
sns.set(style="darkgrid")

# Plot the original and the most distinct augmented signals with different line styles and colors
plt.figure(figsize=(20, 6))

# Plot original signal
plt.plot(original_sample, label='Original Signal', linewidth=2, color='black')

# Different line styles and colors for augmented signals
line_styles = ['--', '-.']
colors = ['blue', 'orange']

for i, idx in enumerate(most_distinct_indices):
    plt.plot(augmented_samples[idx, :], label=f'Augment {i + 1}', linestyle=line_styles[i],
color=colors[i], alpha=1)

plt.xlabel('Feature Index')
plt.ylabel('Signal Value')
plt.legend()

# Use Seaborn to enhance the plot
sns.despine()
plt.show()


X_signals_augmented.shape
labels_augmented.shape


# Calculate class counts
unique_labels, label_counts = np.unique(labels_augmented, return_counts=True)

# Total number of samples in the augmented dataset
total_samples = len(labels_augmented)

# Plot pie chart with count values
plt.figure(figsize=(4,4))
plt.pie(label_counts, labels=unique_labels, autopct=lambda p: '{:.0f}'.format(p * total_samples / 100),
startangle=140, textprops={'color': "black"})
plt.title('Class Counts in Augmented Dataset')
plt.show()
```

```
# Create a DataFrame for easy manipulation
data = pd.DataFrame(data=X_signals_augmented)
data['labels'] = labels_augmented

# Separate distress and normal samples
distress_data = data[data['labels'] == 'Distress']
normal_data = data[data['labels'] == 'Normal']

# Sample random normal samples
balanced_normal_data = normal_data.sample(n=len(distress_data), random_state=42)

# Concatenate distress and balanced normal samples
balanced_data = pd.concat([distress_data, balanced_normal_data])

# Shuffle the balanced dataset
balanced_data = balanced_data.sample(frac=1, random_state=42)

# Split the balanced dataset into features and labels
X_balanced = balanced_data.drop(columns=['labels']).values
labels_balanced = balanced_data['labels'].values

# Split the balanced dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_balanced, labels_balanced, test_size=0.3,
random_state=42)
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)


# Encode labels to numerical values
le = LabelEncoder()
y_train_encoded = le.fit_transform(y_train)
y_test_encoded = le.transform(y_test)

# Reshape the data for CNN input
X_train_cnn = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test_cnn = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)


# Define hyperparameters
initial_learning_rate = 0.00011
batch_size = 64

# Define dynamic learning rate callback
reduce_lr =tf.keras.callbacks.ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.5,
    patience=2,
    verbose=1,
    mode='auto'
)
# Modified model architecture
model = Sequential()

model.add(Conv1D(32, kernel_size=3, activation='relu', input_shape=(X_train_cnn.shape[1],
X_train_cnn.shape[2])))
model.add(MaxPooling1D(pool_size=2))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Conv1D(64, kernel_size=3, activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Flatten())
model.add(Dense(64, activation='relu', kernel_regularizer=l2(0.01)))  # Reduced the number of units
model.add(Dropout(0.7))  # Increased dropout rate
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(loss='binary_crossentropy', optimizer=Adam(learning_rate=initial_learning_rate), metrics=
['accuracy'])



# Display the model summary
model.summary()

# Fit the model with dynamic learning rate
history = model.fit(X_train_cnn, y_train_encoded,
                    epochs=100,
                    batch_size=batch_size,
                    validation_data=(X_test_cnn, y_test_encoded),
                    callbacks=[reduce_lr])


#Save the trained model
model.save('/content/drive/MyDrive/fetaldistress.h5')
```

```
# Extract training history
training_loss = history.history['loss']
training_acc = history.history['accuracy']
val_loss = history.history['val_loss']
val_acc = history.history['val_accuracy']


# Extract learning rates from the history object
learning_rates = history.history['lr']

# Create a dataframe for Seaborn
import pandas as pd
df_lr = pd.DataFrame({
    'Epoch': range(1, len(learning_rates) + 1),
    'Learning Rate': learning_rates
})

# Plot learning rate changes
plt.figure(figsize=(10, 6))
sns.lineplot(data=df_lr, x='Epoch', y='Learning Rate', marker='o')
plt.title('Learning Rate over Epochs')
plt.xlabel('Epoch')
plt.ylabel('Learning Rate')

plt.show()



# Plot training history for accuracy
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.legend()
plt.title('Training and Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')

# Plot training history for loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.legend()
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')

plt.show()


# Evaluate the model
y_pred_prob = model.predict(X_test_cnn)
y_pred = (y_pred_prob > 0.5).astype(int)

# Assuming 'le' is your LabelEncoder
y_test_decoded = le.inverse_transform(y_test_encoded)
y_pred_decoded = le.inverse_transform(y_pred.reshape(-1))


# Confusion Matrix
conf_mat = confusion_matrix(y_test_decoded, y_pred_decoded)

# Plot Confusion Matrix
disp = ConfusionMatrixDisplay(confusion_matrix=conf_mat, display_labels=le.classes_)
disp.plot(cmap='Blues', values_format='d')
plt.title('Confusion Matrix')
plt.show()


# Generate classification report
classification_rep = classification_report(y_test_decoded, y_pred_decoded)
print("Classification Report:\n", classification_rep)
```

APPENDIX-C

ENCLOSURES

The project work carried out here is mapped to SDG-3 Good Health and Well-Being.

The project directly contributes to reducing maternal and newborn mortality. This aligns with SDG 3's focus on ensuring healthy lives and access to quality healthcare, particularly for mothers and infants during childbirth. The innovative approach of utilizing technology such as machine and deep learning for early detection of fetal distress for timely interventions supports the global goal of improving health outcomes and promoting overall well-being. Lastly, the project's potential impact on reducing adverse birth outcomes underscores its relevance to achieving SDG 3's broader aspirations for global health improvement.