Machine Learning-Assisted Approach for Fetal Health Status Prediction using Cardiotocogram Data

Mahmoud Abouelyazid
University of Evansville
Chen Xiang
University of Evansville

Abstract

Objective: This study aims to develop and compare machine learning models for predicting fetal health status using Cardiotocogram (CTG) data, while addressing class imbalance and feature scaling issues.

Methods: CTG data from 2126 records, classified by expert obstetricians into Normal, Suspect, and Pathological classes, was used. Synthetic Minority Over-sampling Technique (SMOTE) was applied to handle class imbalance, and Robust Scalar was used for feature scaling. Logistic Regression, Decision Tree, and Random Forest classifiers were trained and evaluated using various cross-validation techniques, including K-Fold, Stratified K-Fold, and 10-fold cross-validation. Hyperparameter optimization was performed using GridSearch. Model performance was assessed using accuracy, precision, recall, and F1-score metrics.

Results: The Decision Tree classifier achieved the best performance with an accuracy of 93.2%, precision of 94%, recall of 93.2%, and F1-score of 93.4% using SMOTE and Robust Scalar on the 10-fold cross-validation set. The Random Forest classifier obtained an accuracy of 86.9%, precision of 90.3%, recall of 86.9%, and F1-score of 87.8% under similar conditions. Logistic Regression showed an accuracy of 84%, precision of 88.3%, recall of 84%, and F1-score of 85.4% using SMOTE, Robust Scalar, and Stratified K-Fold cross-validation. Feature importance analysis revealed that histogram mean, % time with abnormal long-term variability, and abnormal short-term variability were the most influential features across the models.

Conclusion: This study demonstrates the potential of machine learning models in predicting fetal health status from CTG data. Data balancing with SMOTE and feature scaling with Robust Scalar proved beneficial in improving model performance. The Decision Tree classifier outperformed other models, indicating its suitability for clinical decision support in assessing fetal well-being. Further research is needed to validate these findings and explore the integration of such models into clinical practice.

Keywords: *Machine Learning, Fetal Health Prediction, Cardiotocogram Data, Class Imbalance, Feature Scaling, Decision Tree Classifier*

Introduction

Fetal health refers to the overall well-being and development of an unborn baby during pregnancy [1]. It encompasses various aspects, such as the fetus's growth, organ development, and the absence of congenital abnormalities or infections. Fetal health is

a critical component of a healthy pregnancy, as it lays the foundation for the baby's future physical and cognitive development [2].

Understanding fetal development is necessary for both expectant parents and healthcare providers. Parents can better appreciate the remarkable changes occurring within the womb, by knowing the key milestones and stages of fetal growth. This knowledge also helps them make informed decisions about prenatal care, nutrition, and lifestyle choices that can positively impact their baby's health. For healthcare professionals, a deep understanding of fetal development allows them to monitor the pregnancy effectively, identify potential issues, and provide timely interventions when necessary [3], [4].

The fetal environment plays a role in shaping an individual's future health, including their risk of developing chronic diseases later in life. Exposure to certain factors during pregnancy, such as maternal stress, poor nutrition, and environmental toxins, can have long-lasting effects on the fetus's health. These early exposures can influence the development of vital organs and systems, such as the brain, heart, and immune system, potentially setting the stage for chronic health issues in adulthood.

For example, study [5] has linked maternal obesity and gestational diabetes to an increased risk of obesity, type 2 diabetes, and cardiovascular disease in the offspring. Similarly, exposure to air pollution during pregnancy has been associated with a higher risk of respiratory problems, such as asthma, in children. Healthcare providers can work with expectant mothers to optimize prenatal care and minimize potential risks by understanding the impact of the fetal environment on future health. This may include providing guidance on nutrition, stress management, and reducing exposure to harmful substances, ultimately promoting better health outcomes for the child throughout their life.

Infant mortality has been a significant concern in healthcare systems around the world for many decades. Despite advances in developing tools to evaluate fetal well-being, interpreting cardiotocography (CTG) data remains challenging, especially in regions without expert obstetricians. Even in areas with medical professionals, individually diagnosing fetuses based on CTG measurements is time-consuming and inefficient.

Artificial intelligence (AI) and machine learning (ML) have emerged as promising tools in fetal health prediction, offering automated analysis of ultrasound images and the integration of multiple data sources for risk assessment. AI-powered systems can analyze ultrasound images with remarkable speed and accuracy, detecting subtle abnormalities that may be missed by the human eye. These systems can identify markers for chromosomal disorders, congenital heart defects, and other structural anomalies, providing valuable insights for early intervention and treatment planning [6].

The integration of multiple data sources, such as maternal health records, genetic information, and environmental factors, further enhances the predictive capabilities of AI and ML in fetal health assessment. ML algorithms can process vast amounts of data,

identifying patterns and risk factors that may not be apparent through traditional analysis methods. This holistic approach to risk assessment enables healthcare providers to develop more comprehensive and personalized care plans, taking into account the unique circumstances of each pregnancy. As AI and ML continue to advance, their role in fetal health prediction is expected to grow, offering increasingly sophisticated tools for ensuring the well-being of both the mother and the unborn child [7], [8].

Machine learning models offer a solution to these challenges by allowing fetal health classifications to be made efficiently and without the presence of obstetricians. These models have shown high accuracy in their predictions, making them viable solutions to the difficulties surrounding fetal health assessment. Machine learning techniques are crucial in extracting knowledge and uncovering hidden insights from system data. They contribute to the development of efficient medical decision-making systems by utilizing various tools and technologies to construct algorithms for this purpose. To effectively address these challenges, implementing an explainable model is considered the most efficient approach. An explainable model not only achieves accurate predictions but also provides insights into its decision-making process. This enables scientists and researchers to understand the model's reasoning.

Machine learning models equip with the knowledge to communicate specific abnormal metrics to their patients, leading to improved patient care. For example, if the model predicts a pathological case for a fetus and indicates that the prediction is based on a low frequency of uterine contractions per second, a doctor can advise the patient on appropriate measures. Machine learning models provide transparency and accountability in the decision-making process, which is essential in a field where lives are at stake. These models also help to build trust between healthcare providers and patients, as patients can understand the reasoning behind the decisions made about their health.

Prenatal screening, diagnostic tests, and monitoring techniques have undergone significant advancements, leading to accurate and comprehensive predictions of fetal health. These advancements allow healthcare providers to detect potential issues early on and intervene promptly for optimizing fetal outcomes. Early detection is necessary, as it enables medical professionals to provide timely treatment, modify prenatal care plans, and offer appropriate support to expectant parents.

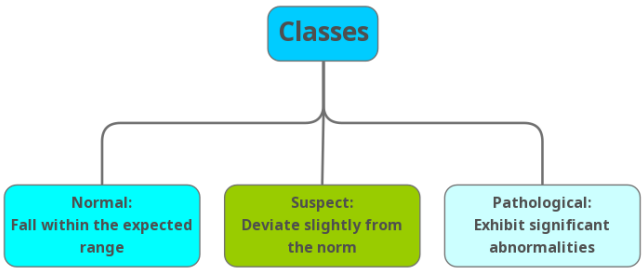
Innovations in prenatal screening, such as non-invasive prenatal testing (NIPT) and high-resolution ultrasounds, have contributed the field of fetal health assessment. These techniques give detailed look into the fetus's genetic makeup, anatomy, and overall development without posing significant risks to the pregnancy. Healthcare providers can now identify chromosomal abnormalities, structural defects, and other potential health concerns with greater precision. This allows for more informed decision-making and personalized care.

Data

According to [9], [10], Cardiotocography (CTG) is a non-invasive and affordable technique used to monitor fetal well-being during pregnancy and labor. CTG machines use ultrasound technology to provide valuable insights into fetal heart rate (FHR) patterns, fetal movements, and uterine contractions. This information enables healthcare professionals to identify potential complications and intervene promptly for reducing the risk of maternal and infant morbidity and mortality.

The dataset under consideration comprises 2,126 samples of features derived from Cardiotocogram examinations. A panel of three experienced obstetricians evaluated and categorized these features into three distinct classes: 1. Normal: CTG readings that fall within the expected range and indicate a healthy fetal state. 2. Suspect: CTG measurements that deviate slightly from the norm, warranting closer monitoring and potential further investigation. 3. Pathological: CTG results that exhibit significant abnormalities, suggesting the presence of fetal distress or other complications that require immediate medical attention.

Figure 1. Target classes



Methods

Data Preprocessing

The 'fetal_health' feature contains the classes we want to predict. This problem is a multiclass classification task with three categories:

- 1: Normal
- 2: Suspect
- 3: Pathological

The class distribution is imbalanced, which can pose challenges for the predictive model. If left unaddressed, the model might learn to perform better on class 1, which has a higher number of instances, compared to the other classes. One strategy to mitigate this issue is to balance the data and evaluate the benefits.

Data Balancing with SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) is an over-sampling method that aims to balance an imbalanced dataset by generating synthetic observations for the minority class [11]. These artificial minority class records are created based on similarity in the predictor space and added to the existing dataset.

It is to note that SMOTE should be applied only to the training set, leaving the test set untouched for final prediction and evaluation [12]. This approach ensures that the model's performance is assessed on real, unseen data.

Feature Selection

Multiple features display correlations with the target variable 'fetal_health'. While it is often advisable to consider removing features that are highly correlated with others, as they may not substantially contribute to the predictive analysis and could potentially be a linear combination of the remaining features, this problem does not impose significant computational constraints. Therefore, the decision has been made to retain all features for subsequent analysis/

Data Scaling with Robust scalar

Data Scaling

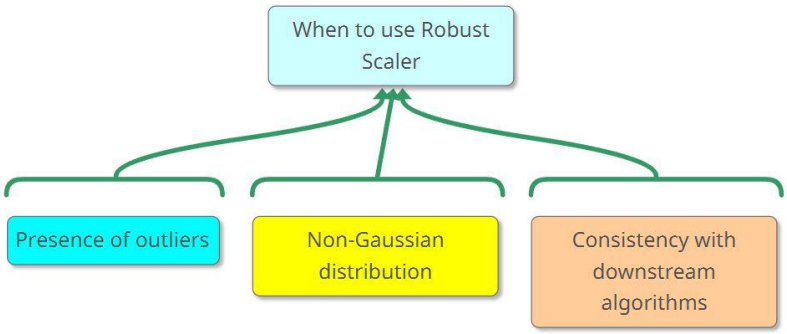
The features in the dataset have varying scales and orders of magnitude. To ensure that all features contribute equally to the predictive model and to avoid bias towards features with larger values, data scaling is necessary. In this case, we will employ the Robust Scaler, which is less sensitive to outliers compared to other scaling techniques like StandardScaler or MinMaxScaler.

Robust Scaler is a data scaling technique that is less sensitive to outliers compared to other scaling methods, such as StandardScaler or MinMaxScaler. It is useful when the dataset contains outliers or when the distribution of the features is not Gaussian.

Table 1. **Characteristic of Robust Scaler**

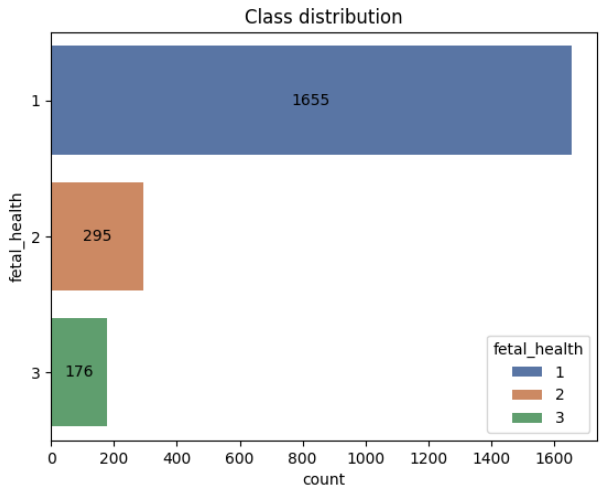
Characteristic	Description
Scaling based on percentiles	Robust Scaler scales the features using the interquartile range (IQR), which is the range between the 25th and 75th percentiles. This reduces the influence of extreme values or outliers [13].
Centering based on the median	Instead of using the mean, Robust Scaler centers the data using the median value. The median is less sensitive to outliers, making it a robust measure of central tendency.
Robust to outliers	Robust Scaler is designed to minimize the impact of outliers on the scaled features by using the IQR for scaling and the median for centering. This ensures better model performance in the presence of outliers.
Preserves the original distribution	Unlike some other scaling techniques, Robust Scaler aims to maintain the original distribution of the data. This can be advantageous when the original distribution holds important information or when the model assumes a specific distribution.

Figure 2. use of Robust scalar



It is necessary to maintain consistency between the training and test sets during the scaling process. The Robust Scaler should be fitted (i.e., the scaling parameters should be calculated) on the training set only. Subsequently, the same scaling parameters should be applied to both the training and test sets [14]. This approach guarantees that the model is evaluated on data that has the same scale as the data it was trained on, preventing any data leakage from the test set into the training process [15].

Figure 3. Class distribution



The dataset exhibits class imbalance, meaning that the number of instances in each class is not evenly distributed. In this case, class 1 has a significantly higher number of instances compared to the other classes. Class imbalance can pose challenges for predictive models, as they may learn to favor the majority class over the minority classes.

When a model is trained on an imbalanced dataset, it may achieve high overall accuracy by simply predicting the majority class most of the time. However, this does not necessarily mean that the model is performing well on the minority classes, which are often the classes of interest in real-world scenarios.

Figure 4. Distribution of the features

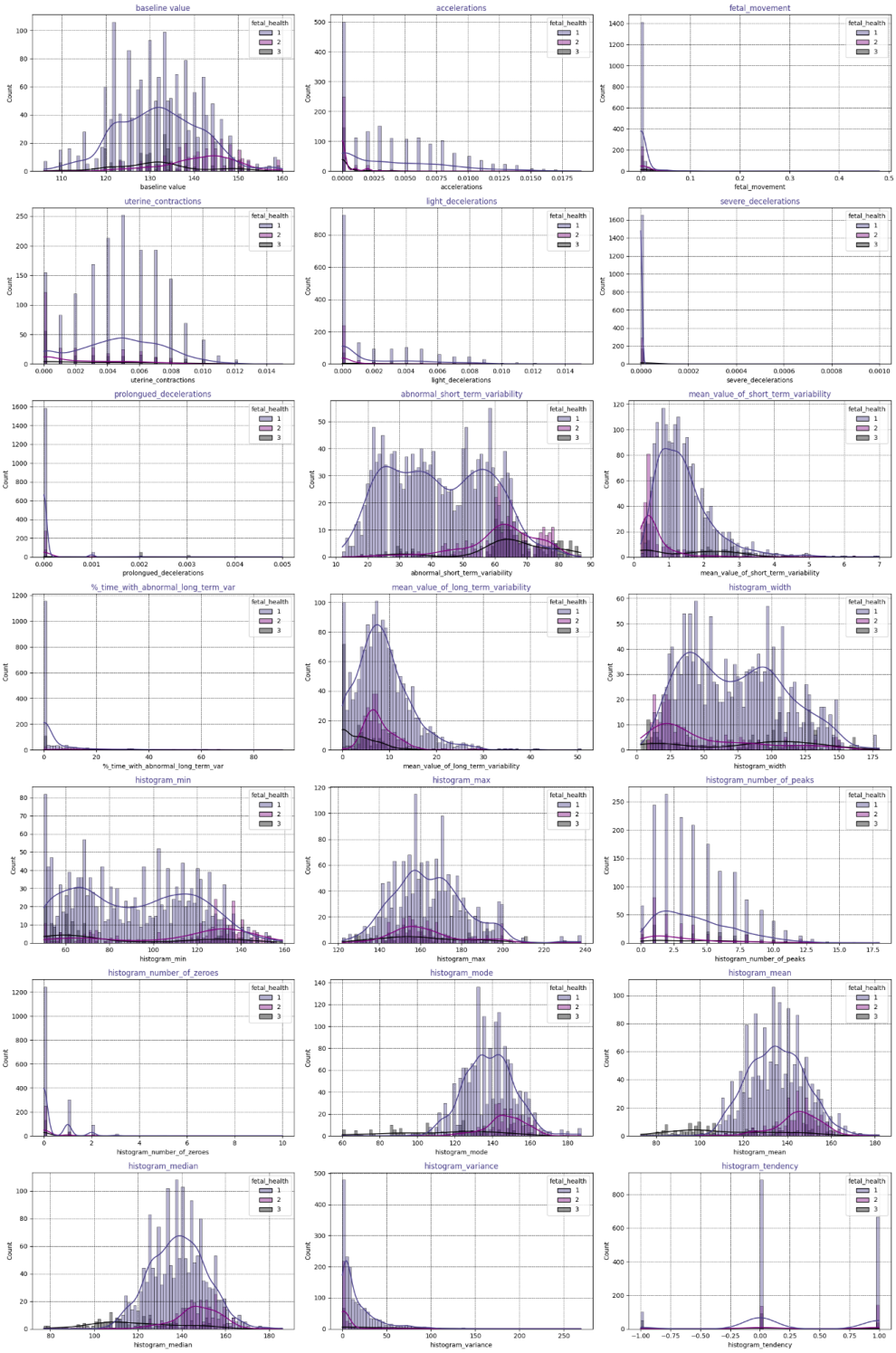

baseline value
accelerations
fetal_movement
uterine_contractions
light_decelerations
severe_decelerations
prolongued_decelerations
abnormal_short_term_variability
mean_value_of_short_term_variability
%_time_with_abnormal_long_term_var
mean_value_of_long_term_variability
histogram_width
histogram_min
histogram_max
histogram_number_of_peaks
histogram_number_of_zeroes
histogram_mode
histogram_mean
histogram_median
histogram_variance
histogram_tendency
fetal_health
1
2
3
Count

Table 2. Consequences and techniques for solving Class Imbalance	
Consequences of Class Imbalance	**Techniques to Address Class Imbalance**
Biased model performance: The model may exhibit better performance metrics (e.g., accuracy, precision, recall) for the majority class compared to the minority classes.	Oversampling the minority classes (e.g., using SMOTE)
	Undersampling the majority class
Poor generalization: The model may struggle to generalize well on unseen data, especially when it comes to correctly classifying instances from the minority classes.	Adjusting class weights during model training
	Using evaluation metrics (e.g., precision, recall, F1-score) that take class imbalance into account

Figure 5. Correlation coefficient heatmap

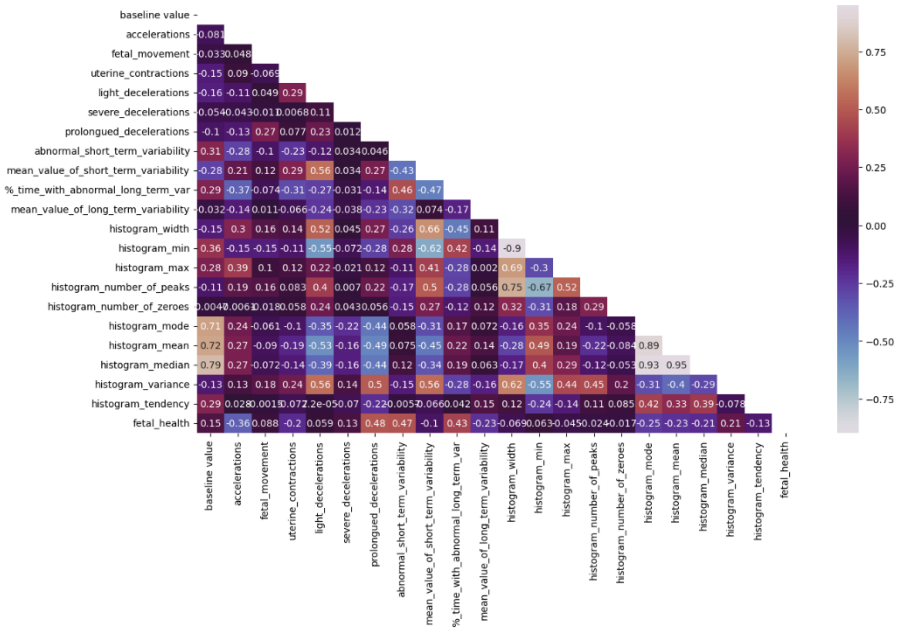
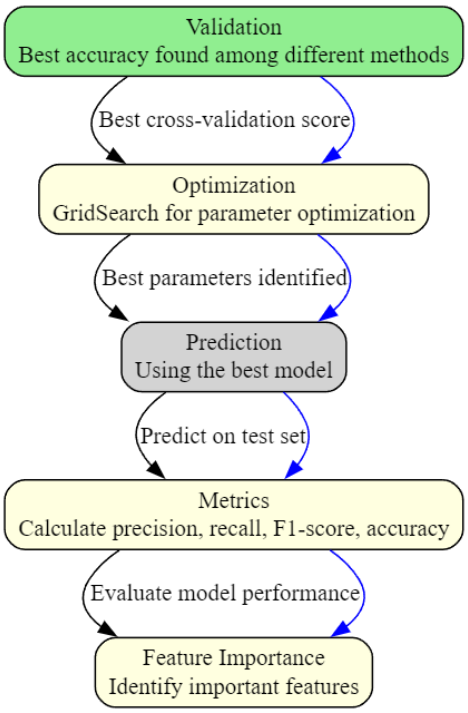


Table 3. Dataset split across train and test		
	Fetal Health class	**Count**
Train	1	1322
Train	2	231
Train	3	147
Test	1	333
Test	2	64
Test	3	29

Prediction

The analysis progressed through several stages, each designed to enhance the model's performance and understanding of the underlying data. Initially, a validation phase was conducted to identify the most suitable cross-validation method among cv=10, K-Fold, and Stratified K-Fold for normal, balanced, and scaled+balanced datasets. This step aimed to determine the approach that yielded the highest accuracy.

Figure 6. prediction workflow of this study



Once the optimal dataset and cross-validation method were identified, the focus shifted to optimizing the model's hyperparameters using GridSearch. This process involved systematically exploring various combinations of hyperparameters to find the configuration that maximized the model's performance.

With the best hyperparameters determined, the model was then employed to predict the test variables, providing insights into its generalization capabilities. To assess the model's performance comprehensively, several metrics were calculated, including Precision, Recall, F1-Score, and Accuracy for both the training and test sets. Finally, an analysis of feature importance was conducted to identify the variables that had the most significant impact on determining the class labels. This information can be valuable for understanding the underlying factors driving the classification outcomes and can guide future data collection and feature engineering efforts.

Predictive Models

The models compared are:

- **Logistic Regression**

- **Decision Tree**
- **Random Forest**

Logistic Regression is a statistical method used for classification tasks. Despite its name, logistic regression is actually a linear model, where the dependent variable is transformed using the logistic function to ensure predictions fall within the range. Mathematically, it models the probability that a given input belongs to a certain class, based on linear combinations of the input features. The model is trained by minimizing a loss function such as cross-entropy, using optimization techniques like gradient descent.

A Decision Tree is a non-parametric supervised learning method used for classification and regression tasks. It recursively partitions the input space into regions, where each partition is determined by the value of a certain feature. At each node of the tree, a decision is made based on a feature value, leading to a split that maximizes information gain or minimizes impurity. Decision Trees are intuitive and easily interpretable, making them for understanding feature importance and explaining predictions. They are prone to overfitting, especially with deep trees, which can be mitigated using techniques like pruning or ensemble methods.

Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive performance and reduce overfitting. It operates by constructing a multitude of decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Each tree is built using a random subset of the training data and a random subset of the features, introducing randomness that helps to decorrelate the trees and reduce variance. Random Forest is known for its robustness and high accuracy across a variety of datasets.

Validation Techniques for the predictive model

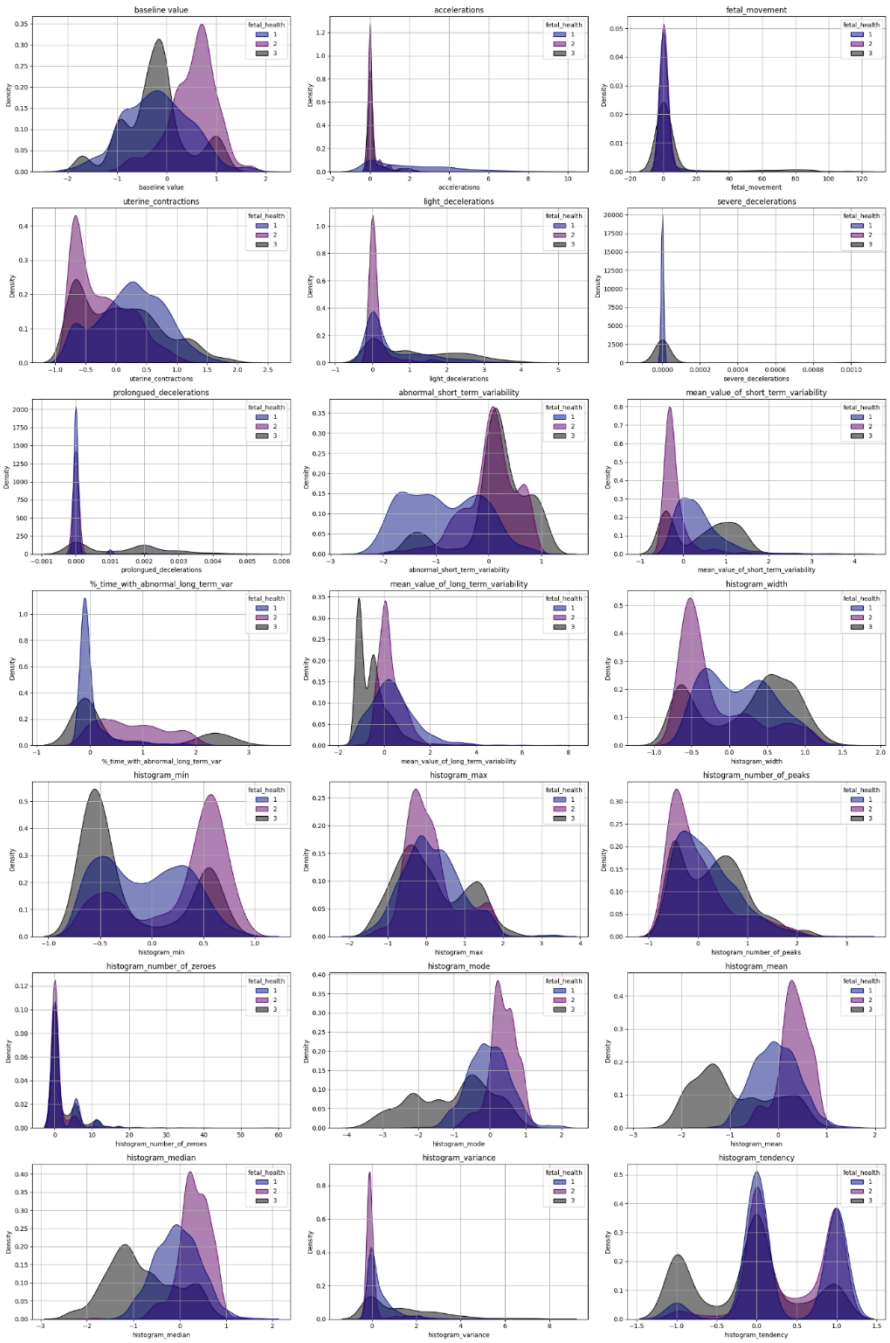
Two primary validation techniques were utilized to evaluate the model's ability to generalize to unseen data. The first approach, K-Fold Cross-Validation, involved partitioning the dataset into k subsets, with each fold containing a balanced distribution of classes. This method ensures that the model is trained and tested on different portions of the data, providing a more reliable estimate of its performance.

The second technique, Stratified K-Fold Cross-Validation, is particularly suitable for imbalanced datasets. It operates similarly to K-Fold but maintains the class distribution within each fold, ensuring that the model is exposed to a representative sample of each class during training and testing.

Both K-Fold and Stratified K-Fold Cross-Validation were implemented with 10 splits, striking a balance between computational efficiency and robustness. Additionally, the data was shuffled before splitting to mitigate any potential bias introduced by the original ordering of the instances. This randomization helps to ensure that the model's performance is not unduly influenced by any specific patterns or trends present in the dataset.

Data Preparation

Figure 7. Feature distribution After SMOTE AND Robust Scalar



To assess the influence of different data preprocessing approaches on the model's performance, three distinct strategies were implemented. The first approach involved

training and evaluating the model using the original, unmodified dataset. This served as a baseline to gauge the model's performance without any data manipulation.

The second strategy employed the Synthetic Minority Over-sampling Technique (SMOTE) to address the class imbalance present in the dataset. SMOTE generates synthetic samples for the minority classes, effectively balancing the class distribution. This approach aims to prevent the model from being biased towards the majority class and to improve its ability to recognize and classify instances from the minority classes.

The third approach combined feature scaling with data balancing. The features were scaled using the Robust Scalar, which is less sensitive to outliers compared to other scaling methods. Scaling ensures that all features contribute equally to the model, preventing features with larger magnitudes from dominating the learning process. Subsequently, SMOTE was applied to the scaled dataset to address class imbalance. This combined approach tackles both feature scale discrepancies and class imbalance simultaneously.

Results

Logistic regression

The effect of data scaling on the performance of the logistic regression model appears to be generally positive across all validation strategies. Scaling the data seems to help the algorithm converge faster and achieve better generalization, leading to improved performance. However, the impact of data balancing using SMOTE is mixed. In some cases, such as when using SMOTE with K-Fold and STK-Fold validation, the performance decreases compared to the original or scaled data. On the other hand, when using SMOTE with cv=10 validation, there is a marginal improvement in performance. This variability suggests that the effectiveness of SMOTE may be dependent on the specific characteristics of the dataset and the chosen validation strategy.

Table 4. Results of Logistic Regression

	Logistic Regression	**Logistic Regression Robust Scaled**	**Logistic Regression Smote**	**Logistic Regression after SMOTE and Robust Scaler**
K-Fold	0.8800	0.8941	0.8276	0.8923
STK-Fold	0.8776	0.8906	0.8268	0.8936
cv=10	0.8824	0.8953	0.8255	0.8911

When comparing the validation strategies, both K-Fold and Stratified K-Fold (STK-Fold) exhibit similar performance trends across different data preprocessing techniques. This consistency indicates that the choice between these two strategies may not have a significant impact on the observed performance differences in this particular scenario. The decision to use Stratified K-Fold suggests that maintaining class balance within each fold was considered important in this context.

Table 5. Accuracy scores	
Metric	**Value**
Accuracy (Train)	0.899
Accuracy (Test)	0.843
Precision (Test)	0.883
Recall (Test)	0.843
F1-Score	0.856

Table 6. Accuracy scores across classes				
	Precision	**Recall**	**F1-Score**	**Support**
Class 1	0.98	0.86	0.92	333
Class 2	0.54	0.77	0.63	64
Class 3	0.54	0.76	0.63	29

Table 7. Accuracy scores after cross validation	
Metric	**Value**
Accuracy	0.843
Precision	0.883
Recall	0.843
F1-Score	0.856
Validation	STK-Fold
Data	Smote+Scaled

The results in tables 5, 6, 7 showcase the performance of a classification model evaluated using various metrics and a Stratified K-Fold (STK-Fold) validation strategy on a dataset that has been preprocessed with SMOTE oversampling and feature scaling.

The model achieves an accuracy of 0.899 on the training set and 0.843 on the test set. This indicates that the model is able to correctly classify approximately 84% of the instances in the test set. The relatively small difference between the training and test accuracies suggests that the model is not severely overfitting to the training data.

The precision score of 0.883 indicates that when the model predicts a positive class, it is correct about 88% of the time. The recall score of 0.843 means that the model correctly identifies around 84% of the actual positive instances. The F1-score, which is the harmonic mean of precision and recall, is 0.856, providing a balanced measure of the model's performance.

It is evident that the model performs exceptionally well on Class 1, with a high precision (0.98), recall (0.86), and F1-score (0.92). This suggests that the model is highly accurate in identifying instances of Class 1 and has a low false positive rate. However, the performance on Class 2 and Class 3 is comparatively lower, with precision, recall, and F1-scores around 0.54, 0.77, and 0.63, respectively. This indicates that the model may struggle more with correctly classifying instances of these classes, potentially due to class imbalance or the complexity of distinguishing between these classes.

The use of SMOTE oversampling and feature scaling as preprocessing techniques aims to address potential class imbalance and ensure that the features are on a similar scale. The STK-Fold validation strategy helps assess the model's performance by maintaining the class distribution across the folds, providing a more reliable estimate of the model's generalization ability.

Table 8. Top 20 features along with their corresponding coefficients in Logistic regression

	Feature	Coefficient
0	accelerations	1.592702
1	histogram_mode	1.529701
2	uterine_contractions	0.802553
3	histogram_median	0.778095
4	light_decelerations	0.509800
5	histogram_width	0.427798
6	mean_value_of_short_term_variability	0.214358
7	mean_value_of_long_term_variability	0.208817
8	histogram_number_of_zeroes	0.042682
9	severe_decelerations	0.000371
10	prolongued_decelerations	-0.047194
11	fetal_movement	-0.056091
12	histogram_number_of_peaks	-0.207527
13	histogram_mean	-0.511047
14	histogram_tendency	-0.524506
15	%_time_with_abnormal_long_term_var	-0.735621
16	histogram_min	-0.800067
17	histogram_max	-0.867143
18	baseline value	-1.225771
19	histogram_variance	-1.628323

Decision tree

Table 9. Decision tree results

	Decision Tree	Decision Tree scaled	Decision Tree SMOTE	Decision Tree SMOTE+Robust Scaled
K-Fold	0.9112	0.9112	0.9581	0.9581
STK-Fold	0.9118	0.9135	0.9592	0.9592
cv=10	0.9118	0.9135	0.9597	0.9594

The results provided show the performance of a Decision Tree classifier under different data preprocessing techniques and validation strategies. The preprocessing techniques include scaling, SMOTE oversampling, and a combination of SMOTE and robust scaling. The validation strategies used are K-Fold, Stratified K-Fold (STK-Fold), and 10-fold cross-validation (cv=10).

Table 10. Performance scores across classes

	precision	recall	f1-score	support
1	0.98	0.93	0.96	333
2	0.75	0.89	0.81	64
3	0.82	0.97	0.89	29
accuracy			0.93	426
macro avg	0.85	0.93	0.89	426
weighted avg	0.94	0.93	0.93	426

Applying SMOTE oversampling significantly improves the model's performance. The accuracy scores increase from around 0.91 without SMOTE to approximately 0.96 with SMOTE, regardless of the validation strategy. This suggests that addressing the class

imbalance through oversampling helps the Decision Tree classifier to better learn and distinguish between the different classes.

The impact of scaling alone (without SMOTE) on the model's performance is minimal, as the accuracy scores remain nearly the same with or without scaling. However, when scaling is applied in combination with SMOTE (SMOTE+Robust Scaled), the model maintains its high performance, indicating that scaling can be beneficial in conjunction with oversampling.

The choice of validation strategy does not seem to have a significant impact on the model's performance, as the accuracy scores are consistent across K-Fold, STK-Fold, and cv=10 for each preprocessing technique. This suggests that the model's performance is robust and not highly sensitive to the specific validation approach used.

In class-wise performance metrics (precision, recall, and f1-score), the model performs exceptionally well for Class 1, with high scores across all metrics. The performance for Class 2 and Class 3 is also strong, with precision scores of 0.75 and 0.82, recall scores of 0.89 and 0.97, and f1-scores of 0.81 and 0.89, respectively. These results indicate that the model is able to effectively distinguish between the different classes, despite the potential challenges posed by class imbalance.

After performing hyperparameter optimization, the best model achieved an impressive accuracy score of 94.98%. The optimal hyperparameters found through this process include using the entropy criterion, a maximum depth of 9, a minimum of 1 sample per leaf, and a minimum of 4 samples required to split an internal node. These settings strike a balance between model complexity and generalization ability, allowing the Decision Tree classifier to capture the underlying patterns in the data while avoiding overfitting.

Table 11. **Feature importance in decision tree**

Feature	Importance
histogram_mean	0.235199
mean_value_of_short_term_variability	0.199540
%_time_with_abnormal_long_term_var	0.197643
abnormal_short_term_variability	0.112131
prolongued_decelerations	0.075840
histogram_median	0.054215
accelerations	0.025330
baseline value	0.014523
histogram_number_of_peaks	0.014520
histogram_mode	0.014317
fetal_movement	0.013112
histogram_max	0.009530
histogram_variance	0.008829
histogram_tendency	0.007238
uterine_contractions	0.006143
light_decelerations	0.004501
histogram_width	0.003507

mean_value_of_long_term_variability	0.002500
histogram_number_of_zeroes	0.001380

Random forest

Table 12. Random forest results

Method	**RFC**	**RFC scaled**	**RFC smote**	**RFC smote+scaled**
K-Fold	0.9400	0.9394	0.9796	0.9796
STK-Fold	0.9435	0.9441	0.9786	0.9786
Cross-Validation	0.9347	0.9347	0.9793	0.9793

Table 13. Accuracy scores across classes

	precision	**recall**	**f1-score**	**support**
1	0.98	0.86	0.92	333
2	0.59	0.88	0.70	64
3	0.68	0.90	0.78	29
accuracy			0.87	426
macro avg	0.75	0.88	0.80	426
weighted avg	0.90	0.87	0.88	426

The tables 12 and 13 show the performance of a Random Forest Classifier (RFC) under different data preprocessing techniques and validation strategies. The preprocessing techniques include scaling, SMOTE oversampling, and a combination of SMOTE and scaling. The validation strategies used are K-Fold, Stratified K-Fold (STK-Fold), and cross-validation. Applying SMOTE oversampling significantly improves the model's performance. The accuracy scores increase from around 0.94 without SMOTE to approximately 0.98 with SMOTE, regardless of the validation strategy. This suggests that addressing the class imbalance through oversampling helps the Random Forest Classifier to better learn and distinguish between the different classes.

The impact of scaling alone (without SMOTE) on the model's performance is minimal, as the accuracy scores remain nearly the same with or without scaling. However, when scaling is applied in combination with SMOTE (SMOTE+Scaled), the model maintains its high performance, indicating that scaling can be beneficial in conjunction with oversampling.

The choice of validation strategy has a slight impact on the model's performance, with STK-Fold and K-Fold yielding slightly higher accuracy scores compared to cross-validation. However, the differences are relatively small, suggesting that the model's performance is robust across different validation approaches.

After performing hyperparameter optimization, the best-performing Random Forest Classifier model achieved an accuracy score of approximately 90.4%. The optimal hyperparameters found include using the entropy criterion, a maximum depth of 9, selecting the square root of the total number of features at each split (max_features='sqrt'), limiting the maximum number of leaf nodes to 9, and using 1000 decision trees in the ensemble (n_estimators=1000).

Table 14. Feature Importance in random forest	
Feature	**Importance**
% Time with Abnormal Long Term Var	0.136700
Histogram Mean	0.126976
Abnormal Short Term Variability	0.117263
Histogram Median	0.095905
Prolonged Decelerations	0.095337
Accelerations	0.091215
Mean Value of Short Term Variability	0.076574
Mean Value of Long Term Variability	0.062427
Histogram Mode	0.061983
Baseline Value	0.033225
Histogram Variance	0.032504
Uterine Contractions	0.015113
Histogram Min	0.014982
Histogram Width	0.014270
Light Decelerations	0.009128
Histogram Max	0.006681
Fetal Movement	0.003883
Histogram Tendency	0.003112
Histogram Number of Peaks	0.002169
Severe Decelerations	0.000553

It can be observed that the Random Forest Classifier achieves an accuracy of 0.869, precision of 0.903, recall of 0.869, and an F-score of 0.878 using cross-validation and SMOTE+Robust Scaled data. These scores are slightly lower than the Decision Tree model but higher than the Logistic Regression model.

The model performs exceptionally well for Class 1, with high precision (0.98), recall (0.86), and F1-score (0.92). The performance for Class 2 and Class 3 is also good, with precision scores of 0.59 and 0.68, recall scores of 0.88 and 0.90, and F1-scores of 0.70 and 0.78, respectively. These results indicate that the model is able to effectively distinguish between the different classes, although there is room for improvement in the precision scores for Class 2 and Class 3.

Table 15. performance of the 3 models						
Model	**Accuracy**	**Precision**	**Recall**	**F-score**	**Validation**	**Data**
Logistic Regression	0.84	0.883	0.84	0.854	STK-Fold	Smote+Scaled
Decision Tree	0.932	0.94	0.932	0.934	CV=10	Smote+Scaled
Random Forest	0.869	0.903	0.869	0.878	K-Fold	Smote+Scaled

In the logistic regression model, the top 20 features and their corresponding coefficients provide insights into the impact of each feature on the prediction. Features with positive coefficients, such as "accelerations," "histogram_mode," and "uterine_contractions," have a positive influence on the prediction, meaning that higher values of these features are associated with a higher probability of the positive class. On the other hand, features with negative coefficients, such as "histogram_variance," "baseline value," and "histogram_max," have a negative impact on the prediction, indicating that higher values of these features are associated with a lower probability of the positive class.

The decision tree model provides feature importance scores, which represent the relative importance of each feature in making predictions. The top features in the decision tree model include "histogram_mean," "mean_value_of_short_term_variability," "%_time_with_abnormal_long_term_var," and "abnormal_short_term_variability." These features play a significant role in the decision-making process of the tree, with higher importance scores indicating a greater influence on the final predictions.

Similarly, the random forest model also provides feature importance scores. The top features in the random forest model are "% Time with Abnormal Long Term Var," "Histogram Mean," "Abnormal Short Term Variability," "Histogram Median," and "Prolonged Decelerations." These features are considered the most informative and have the highest impact on the predictions made by the ensemble of decision trees in the random forest.

Comparing the feature importance across the three models, we can observe some commonalities and differences. "histogram_mean" appears as a top feature in both the decision tree and random forest models, indicating its significance in making predictions. "%_time_with_abnormal_long_term_var" is highly important in the decision tree and random forest models but has a negative coefficient in the logistic regression model. "accelerations" has a high positive coefficient in logistic regression and is also considered important in the random forest model. "mean_value_of_short_term_variability" is ranked high in the decision tree model but has a relatively lower coefficient in logistic regression. "prolongued_decelerations" has a negative coefficient in logistic regression but is considered important in both the decision tree and random forest models.

These differences in feature importance across models can be attributed to the different ways each model handles and interprets the features. Logistic regression considers the linear relationship between the features and the log-odds of the outcome, while decision trees and random forests capture non-linear relationships and interactions between features.

Conclusion

Infant mortality remains a significant concern in healthcare systems worldwide, and machine learning models offer a solution to the challenges surrounding fetal health assessment. Explainable models, in particular, provide transparency and accountability in the decision-making process, helping to build trust between healthcare providers and patients.

This study demonstrates the potential of machine learning models, particularly the Decision Tree classifier, in predicting fetal health status using Cardiotocogram (CTG) data. By applying feature scaling with Robust Scalar, the models achieved promising performance metrics, with the Decision Tree classifier outperforming the Random Forest and Logistic Regression models. The identification of influential features, such as histogram mean, % time with abnormal long-term variability, and abnormal short-

term variability, provides insights into the key factors contributing to fetal health assessment.

The class imbalance problem is a common challenge in machine learning, particularly in medical datasets where certain conditions or outcomes may be relatively rare compared to others. In the case of the CTG dataset used in this study, the original class distribution was imbalanced, meaning that the number of instances in each class (Normal, Suspect, and Pathological) was not equal. Imbalanced class distribution can lead to biased models that favor the majority class and perform poorly in predicting the minority classes, which are often of greater interest in clinical settings.

To address the class imbalance issue, the researchers employed the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is a popular oversampling method that generates synthetic examples of the minority classes by interpolating between existing minority instances. By creating additional synthetic examples, SMOTE aims to balance the class distribution and provide the machine learning models with a more representative dataset for training.

While SMOTE has been widely used and has shown success in mitigating class imbalance, it is important to recognize its limitations. SMOTE generates synthetic examples based on the existing minority instances, assuming that the interpolation between them captures the underlying distribution of the minority classes. However, this assumption may not always hold true, especially if the minority classes have complex or non-linear decision boundaries. In such cases, SMOTE may generate synthetic examples that do not accurately represent the true distribution of the minority classes, leading to unrealistic or noisy instances.

Moreover, SMOTE's interpolation approach may introduce some bias into the dataset. By creating synthetic examples, SMOTE may amplify certain patterns or characteristics present in the existing minority instances, potentially overemphasizing certain regions of the feature space. This bias can affect the model's learning process and lead to overoptimistic performance on the minority classes during training and evaluation.

While SMOTE can help improve the model's performance on the minority classes, it does not guarantee a perfect representation of the true underlying distribution. The generated synthetic examples may not capture all the intricacies and variability present in real-world data. Therefore, the results obtained using SMOTE should be interpreted with caution, and the models should be validated on independent, unseen data to assess their generalization ability.

To mitigate the potential biases introduced by SMOTE, studies can consider using other advanced oversampling techniques, such as Adaptive Synthetic (ADASYN) or Generative Adversarial Networks (GANs), which aim to generate more realistic and diverse synthetic examples. Additionally, using a combination of oversampling and undersampling techniques, such as SMOTE with Tomek Links or SMOTE with Edited Nearest Neighbors (ENN), can help balance the class distribution while removing noisy or overlapping instances.

The interpretation of feature importance rankings is a crucial aspect of understanding the results of machine learning models. However, it is important to recognize that the interpretation requires domain knowledge to assess the relevance and practicality of the identified features in the context of the problem being addressed.

Domain expertise plays a big role in determining whether the top features identified by the models make sense and align with the underlying principles and mechanisms of the specific domain. For example, in a healthcare-related problem, a medical expert would be best suited to evaluate if the top features, such as "accelerations," "histogram_mode," or "uterine_contractions," have a meaningful connection to the outcome being predicted. They can assess whether these features are clinically relevant, have a plausible biological or physiological basis, and align with established medical knowledge.

Without sufficient domain expertise, it may be challenging to make such assessments. A data scientist or machine learning practitioner who lacks the necessary domain knowledge may struggle to determine if the identified features are truly informative or if they are merely statistical artifacts. They may not have the background to evaluate the feasibility or practicality of using certain features in real-world decision-making processes.

Domain expertise can help identify potential confounding factors or variables that may influence the relationship between the features and the outcome. A domain expert can provide closer look into whether the top features are likely to be causally related to the outcome or if they are simply correlated due to other underlying factors. They can also help assess if the identified features are practical to measure or collect in real-world settings, considering factors such as cost, accessibility, and reliability.

In cases where the top features identified by the models do not align with domain knowledge or expectations, it may indicate the need for further investigation. It could suggest the presence of hidden biases, data quality issues, or limitations in the modeling approach. Domain experts can guide the refinement of the models, suggest alternative features or data sources, and provide valuable context for interpreting the results.

References

[1] R. Woods, “Death before birth: Fetal health and mortality in historical perspective,” Aug. 2009.

[2] J. Lalor and C. Begley, “Fetal anomaly screening: what do women want to know?,” *J. Adv. Nurs.*, vol. 55, no. 1, pp. 11–19, Jul. 2006.

[3] P. D. Gluckman, *The Fetal Matrix: Evolution, Development and Disease*. Cambridge University Press, 2004, p. 272.

[4] D. Almond and J. Currie, “Killing Me Softly: The Fetal Origins Hypothesis,” *J. Econ. Perspect.*, vol. 25, no. 3, pp. 153–172, Summer 2011.

[5] K. Tenenbaum-Gavish and M. Hod, “Impact of maternal obesity on fetal health,” *Fetal Diagn. Ther.*, vol. 34, no. 1, pp. 1–7, Jun. 2013.

[6] J. B. Josimovich, B. Kosor, L. Boccella, D. H. Mintz, and D. L. Hutchinson, "Placental lactogen in maternal serum as an index of fetal health," *Obstet. Gynecol.*, vol. 36, no. 2, pp. 244–250, Aug. 1970.
[7] K. M. Godfrey and D. J. Barker, "Fetal programming and adult health," *Public Health Nutr.*, vol. 4, no. 2B, pp. 611–624, Apr. 2001.
[8] W. D. Rees, C. J. McNeil, and C. A. Maloney, "The Roles of PPARs in the Fetal Origins of Metabolic Health and Disease," *PPAR Res.*, vol. 2008, p. 459030, 2008.
[9] A. Pinas and E. Chandraharan, "Continuous cardiotocography during labour: Analysis, classification and management," *Best Pract. Res. Clin. Obstet. Gynaecol.*, vol. 30, pp. 33–47, Jan. 2016.
[10] C. Pehrson, J. L. Sorensen, and I. Amer-Wåhlin, "Evaluation and impact of cardiotocography training programmes: a systematic review," *BJOG*, vol. 118, no. 8, pp. 926–935, Jul. 2011.
[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *jair*, vol. 16, pp. 321–357, Jun. 2002.
[12] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem," in *Advances in Knowledge Discovery and Data Mining*, 2009, pp. 475–482.
[13] R. D. Martin and R. H. Zamar, "Bias Robust Estimation of Scale," *aos*, vol. 21, no. 2, pp. 991–1017, Jun. 1993.
[14] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognit.*, vol. 38, no. 12, pp. 2270–2285, Dec. 2005.
[15] A. B. Kahng, S. Mantik, and I. L. Markov, "Min-max placement for large-scale timing optimization," in *Proceedings of the 2002 international symposium on Physical design*, San Diego, CA, USA, 2002, pp. 143–148.