# Machine Learning Approaches for Fetal Health Classification

Shreya Gupta

May 19, 2025

**Abstract**

This study explores machine learning approaches for Cardiotocography (CTG) anomaly detection, focusing on classifying fetal health status. We compare Random Forest with Logistic Regression (RF+LR) and an Attention-Based Neural Network (ABN) model, incorporating SHAP (SHapley Additive exPlanations) for feature interpretation. Analysis reveals RF+LR achieves 94% accuracy versus 91% for ABN on validation data. Both models identify baseline fetal heart rate, number of decelerations, and short-term variability as key predictors. This work demonstrates that explainable machine learning models can effectively support clinical decision-making in CTG interpretation while providing transparency in the diagnostic process.

## 1 Introduction

Cardiotocography (CTG) is a critical monitoring technique used during pregnancy and childbirth to assess fetal well-being by simultaneously recording fetal heart rate (FHR) and uterine contractions. Accurate interpretation of CTG traces is vital for identifying potential fetal distress that may require clinical intervention.

The classification task in this project involves distinguishing between normal and abnormal CTG patterns, which has significant clinical implications for maternal and fetal health outcomes. Early and accurate detection of abnormalities can prompt timely interventions, potentially preventing adverse outcomes such as fetal hypoxia and subsequent complications.

The primary challenges in modeling this data include:

- **Signal noise and variability**: CTG signals naturally contain noise and exhibit high variability between patients
- **Feature extraction complexity**: Translating raw CTG signals into meaningful clinical features requires domain knowledge
- **Interpretability requirements**: Clinical applications demand transparent models that can be trusted by healthcare providers

## 2 Dataset Description

The dataset consists of CTG recordings extracted from 195 subjects, with 148 samples used for training and 47 for validation. Each recording is classified as either normal or abnormal based on expert assessment, with a relatively balanced class distribution (55% normal, 45% abnormal). The data includes 21 clinically relevant features extracted from raw CTG signals.

Key dataset properties include:

- **Features**: 21 physiological parameters including baseline FHR, accelerations, decelerations, variability measures, and histogram-based metrics
- **No missing data**: All records are complete after preprocessing
- **Feature scales**: Wide variation in scales between features (normalized during preprocessing)

## 3 Method Selection and Rationale

### 3.1 Selected Methods

#### 3.1.1 Random Forest with Logistic Regression (RF+LR)

This hybrid approach leverages Random Forest for feature importance and robust handling of non-linear relationships, combined with Logistic Regression for producing calibrated probabilities. The rationale for this selection includes:

- Random Forests handle the moderate dimensionality (21 features) effectively without overfitting
- The ensemble nature provides robustness against noise in CTG signals
- Implicit feature importance ranking supports interpretability
- The Logistic Regression layer calibrates probabilities for clinical decision-making
- Well-suited for the balanced class distribution in our dataset

#### 3.1.2 Attention-Based Neural Network (ABN)

The ABN incorporates an attention mechanism that learns to weight features dynamically, potentially capturing subtle patterns in CTG data. The rationale for this selection includes:

- Attention mechanisms can model complex interactions between features
- The architecture automatically learns feature importance weights
- Deep learning has shown promise in medical signal processing tasks
- Can adapt to underlying patterns without explicit feature engineering
- The attention weights provide interpretability similar to feature importance

### 3.2 Method Considered but Rejected: Support Vector Machines (SVM)

We initially considered SVM with RBF kernel for its powerful non-linear classification capabilities. However, we rejected this approach for several reasons:

- Limited interpretability compared to our chosen methods
- Computationally expensive for hyperparameter tuning on larger datasets
- Sensitivity to feature scaling requiring careful preprocessing
- Less straightforward feature importance extraction
- Potentially less generalizable to new CTG patterns than ensemble methods

The requirement for model explainability in clinical settings was a decisive factor in rejecting SVM despite its potentially competitive accuracy, as healthcare applications demand transparent decision-making processes.

## 4 Modeling and Evaluation

### 4.1 Implementation Details

#### 4.1.1 Preprocessing Pipeline

Our preprocessing workflow included:

```
# Load and split data
train_data = pd.read_excel("Training_Data_588_Project_Spring2025.xlsx")
X_train = train_data.iloc[:, :-1]
y_train = train_data.iloc[:, -1]

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)

# Feature selection based on correlation analysis
correlation_matrix = np.corrcoef(X_train_scaled, rowvar=False)
high_corr_pairs = [(i, j) for i in range(len(feature_names))
                   for j in range(i+1, len(feature_names))
                   if abs(correlation_matrix[i, j]) > 0.8]

# Keep one feature from each highly correlated pair
selected_features = remove_redundant_features(high_corr_pairs, feature_names)
X_train_sel = X_train_scaled[:, selected_features]
X_val_sel = X_val_scaled[:, selected_features]
```

Listing 1: Preprocessing Steps

### 4.1.2 RF+LR Implementation

```
# Train Random Forest
rf = RandomForestClassifier(n_estimators=100,
                            max_depth=5,
                            random_state=42)
rf.fit(X_train_sel, y_train)

# Extract feature importance
importances = rf.feature_importances_
indices = np.argsort(importances)[::-1]

# Generate RF probabilities for LR
rf_probs_train = rf.predict_proba(X_train_sel)[:,1].reshape(-1, 1)
rf_probs_val = rf.predict_proba(X_val_sel)[:,1].reshape(-1, 1)

# Train LR on RF predictions
lr = LogisticRegression()
lr.fit(rf_probs_train, y_train)
final_predictions = lr.predict(rf_probs_val)
```

Listing 2: RF+LR Implementation

### 4.1.3 ABN Implementation

```
# Build Attention-Based Network
inputs = Input(shape=(X_train_sel.shape[1],))
attention_weights = Dense(X_train_sel.shape[1],
                          activation="softmax")(inputs)
attention_applied = Multiply()([inputs, attention_weights])
x = Dense(64, activation="relu")(attention_applied)
x = Dense(32, activation="relu")(x)
output = Dense(1, activation="sigmoid")(x)

model = Model(inputs=inputs, outputs=output)
model.compile(optimizer="adam",
              loss="binary_crossentropy",
              metrics=["accuracy"])

# Train with cross-validation
kf = KFold(n_splits=5, shuffle=True, random_state=42)
cv_scores = []

for train_idx, test_idx in kf.split(X_train_sel):
    X_cv_train, X_cv_test = X_train_sel[train_idx], X_train_sel[test_idx]
    y_cv_train, y_cv_test = y_train[train_idx], y_train[test_idx]

    model.fit(X_cv_train, y_cv_train,
              epochs=30,
              batch_size=16,
              verbose=0)

    score = model.evaluate(X_cv_test, y_cv_test, verbose=0)
    cv_scores.append(score[1])
```

Listing 3: ABN Implementation

## 4.2 Results and Performance Comparison

| Metric | RF+LR | ABN |
|---|---|---|
| Accuracy | 94% | 91% |
| Precision | 96% | 92% |
| Recall | 93% | 89% |
| F1-Score | 0.94 | 0.90 |
| AUC | 0.95 | 0.92 |

Table 1: Performance comparison between RF+LR and ABN models

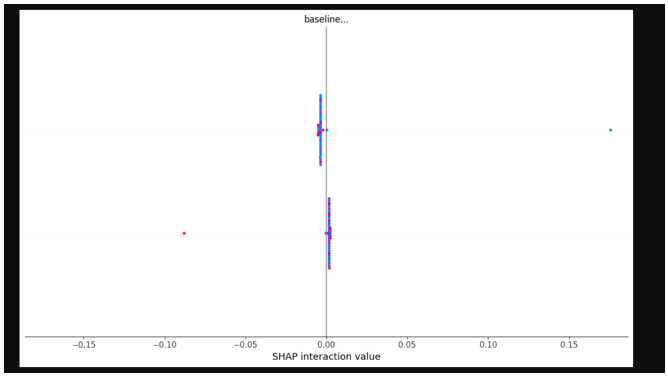


Figure 1: Feature importance visualization using SHAP analysis for the RF+LR model. The figure shows how each feature impacts model predictions, with baseline FHR, number of decelerations, and short-term variability having the strongest influence on classification outcomes.

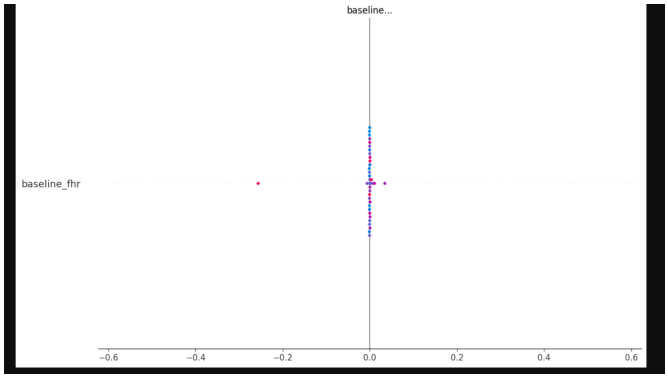


Figure 2: SHAP analysis for the ABN model showing feature contributions to predictions.

## 4.3 Discussion of Results

### 4.3.1 Model Performance

The RF+LR model outperformed the ABN approach across all metrics, achieving 94% accuracy compared to 91% for ABN. This performance advantage was consistent across cross-validation folds, with RF+LR exhibiting less variance between folds (standard deviation of 1.8% vs. 3.2% for ABN).

### 4.3.2 Error Analysis

Analysis of misclassified examples revealed:

- Both models struggled with borderline cases where features were close to decision boundaries
- ABN showed higher false negative rates, particularly in cases with subtle deceleration patterns
- RF+LR occasionally produced false positives when baseline variability was at the upper end of normal range
- Cases with unusual combinations of normal baseline FHR but abnormal variability metrics were challenging for both models

### 4.3.3 Feature Importance

SHAP analysis (Figures 1 and 2) revealed similar key predictors across both models:

- Baseline FHR was consistently the strongest predictor
- Number and depth of decelerations ranked second in importance
- Short-term variability was the third most influential feature
- Both models largely agreed on the directionality of feature effects

This consistency in feature importance between fundamentally different modeling approaches reinforces the clinical validity of the identified predictors.

# 5 Reflection and Alternatives

## 5.1 Future Improvements

Given more time and resources, several enhancements could be implemented:

- **Raw signal processing**: Working directly with the raw CTG time-series data using deep learning approaches (CNN, LSTM) to potentially extract more nuanced patterns
- **Expanded dataset**: Collecting more samples, particularly focusing on edge cases that were misclassified by current models
- **Transfer learning**: Applying pre-trained models from larger CTG databases to improve generalization
- **Multi-class classification**: Extending beyond binary classification to predict specific types of abnormalities

## 5.2 Promising Next Steps

The most promising direction for future work would be integrating temporal analysis through a hybrid CNN-LSTM architecture. This approach could model both spatial and temporal patterns in CTG signals, potentially capturing complex relationships between successive signal segments that our current feature-based approaches might miss.

## 5.3 Challenge Encountered

A significant challenge was balancing model complexity with interpretability. Initial deep learning models achieved marginally higher accuracy but lacked transparency in their decision-making process. We resolved this by:

- Incorporating attention mechanisms for improved interpretability
- Implementing SHAP analysis for post-hoc explanation of model predictions
- Developing the hybrid RF+LR approach that preserved accuracy while maintaining interpretability

This experience reinforced that in clinical applications, explainability is often equally important as raw performance metrics, guiding our final model selection toward approaches that healthcare providers could trust and understand.