# Fetal Health Classification

Angelina Xia[301322630], Mitchell Liu[301436305], Ryan Wu[301392875], Jerome Lau[301411165], and Jiachen Sun[301436357]

{axia,mhl36,rwa106,jla685,jsa356}@sfu.ca

**Abstract.** Monitoring fetal heart rate during labour and pregnancy is vital to ensuring the health of the fetus, as it can indicate possible hypoxia or other complications prior to birth. In this report, fetal heart rate data gathered using a technique called Cardiotocography (CTG) is preprocessed and analyzed with various machine learning methods to classify and predict fetal health.

**Keywords:** Cardiotocography (CTG) · Machine Learning (ML) · Fetal Heart Rate (FHR) · Convolutional Neural Network (CNN).

## 1 Introduction

Cardiotocography (CTG) is a method commonly used to assess fetal heart rate with ultrasound during labour, and is often used to determine if a fetus is at risk of hypoxia. We evaluated methods on two different datasets to create models that accurately predict the health of a fetus. Our models on the first dataset include testing several classification models to predict/classify the health of the fetus, and identify which model is most effective. On the second dataset, we preprocess biosignals into usable data, then represent as images and train a DCNN to predict fetal health. We also venture into feature extraction to test the efficacy of methods used in the first dataset on the second.

## 2 Materials

We primarily worked with two datasets - a simple tabulated cardiotocography from Kaggle, and a complex raw waveform/metadata dataset (CTU-CHB). We tested and performed feature analysis, extraction, and prediction on both datasets with different techniques and also reconciled the two datasets by comparing model performance between the two.

The tabulated CTG data [1] has 21 features across 2126 rows and a label column, 'fetal_health', as shown in table 1. It was tabulated by expert obstetricians into three health categories: Normal, Suspect, and Pathological.

The CTU-CHB dataset [2] is a collection of 552 CTGs from the Czech Technical University collected between 2010 and 2012 and is a raw waveform dataset with a

Table 1: Count of Tabulated CTG dataset

| Fetal Health | Count |
|---|---|
| Normal | 1655 |
| Suspect | 295 |
| Pathological | 176 |

fetal heart rate signal and a uterine contraction signal. The signal has a sampling frequency of 4 Hz and a maximum recording time of 90 minutes. Based on these studies [11][12][13][14], we decided to set one of our thresholds for determining hypoxia as the pH value being greater or equal to 7.15. Metadata annotates fetal clinical details, which are extracted and shown in Table 2.

Table 2: First 5 entries of the CTU-CHB dataset

| pH | BDecf | pCO2 | BE | Apgar1 | Apgar5 |
|---|---|---|---|---|---|
| 7.14 | 8.14 | 7.7 | -10.5 | 6 | 8 |
| 7.00 | 7.92 | 12.0 | -12.0 | 8 | 8 |
| 7.20 | 3.03 | 8.3 | -5.6 | 7 | 9 |
| 7.30 | 5.19 | 5.5 | -6.4 | 8 | 9 |
| 7.30 | 4.52 | 5.7 | -5.8 | 9 | 10 |

## 3 Methods

### 3.1 Machine Learning on Tabulated Dataset

We first constructed and tested our models on the tabulated dataset from Kaggle. We decided to start off using machine learning algorithms, as we were more familiar with these methods. We wanted to mainly use classification algorithms on our dataset, as we were attempting to classify the health of the infant as, normal, suspect, and pathological.

We decided to try 5 different methods to test which one would have the highest accuracy in predicting the condition of the infant. The first was the Decision Trees Classifier, which functioned by breaking the dataset into smaller subsets based on the most significant attribute at each node. Then we tested the Random Forest algorithm, which used a multitude of randomly generated decision trees to reach a single result. Next was the K-Nearest Neighbors method, which worked by classifying samples based on the classes of the k number of closest samples. We then tested the Gaussian Naive Bayes algorithm. Unlike the previous ones, this method only functions properly if the dataset falls under a normal distribution and each feature has an independent effect on the prediction of the target variable. Using Bayes' theorem, this algorithm predicts class probabilities by combining prior probabilities with likelihood estimates of observed features. Lastly, a neural network was trained, which has an input layer, several hidden layers, and an output layer with one node for each output category. Oversampling of the minority classes was performed to mitigate class imbalance.

Feature selection was also performed. Below are the results for a Chi-squared test, which determines the statistical significance of each feature on the output classifications. This was used to cull features dimensions for our machine learning methods.

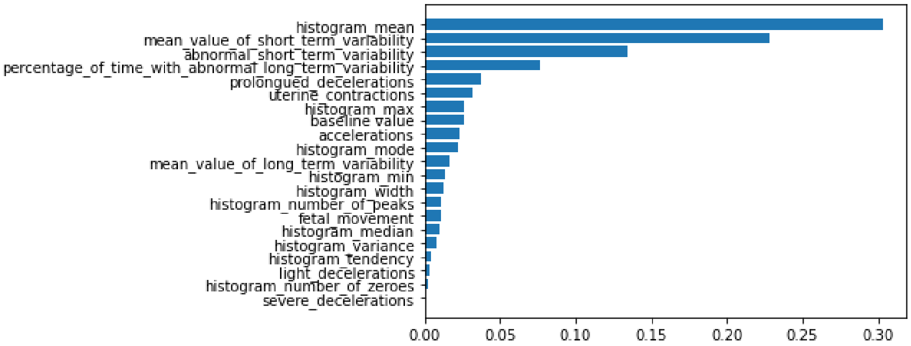


Fig. 1: Chi2 Feature Selection

### 3.2 CTU-CHB Preprocessing

A second dataset was introduced as well, this time with access to the original waveform biosignals. As our data entries are often large, waveforms can be represented as images, either through recurrence plots or spectrograms ([7][4]).

In clinical practice, fetal heart rate signals are measured by an electronic transducer attached to the abdomen of the mother during delivery. This signal may contain artifacts or spikes due to displacement of the transducer, or fetal or maternal movement. Thus, preprocessing was carried out by following several guidelines found in the paper ”A Comprehensive Feature Analysis of the Fetal Heart Rate Signal for the Intelligent Assessment of Fetal State”[5], as well as utilizing functions written by Doug Williams in their GitHub repository ”ML/DL analysis of Cardiotocography (CTG) traces using Recurrence Plot”[8].

Several methods were employed to clean the signals:

1. Values of over 200 or under 50 were considered data spikes, and removed.

2. Long gaps as defined by gaps of 10 seconds or longer were removed.

3. Short gaps were filled with linear interpolation.

4. Heart rate changes of over 25BPM from one sample to the next were filtered out as invalid data, and then replaced with linearly interpolated data.

5. The resulting data was then filtered to find valid segments, which were then spliced together to form the cleaned data for processing.

6. 7 records contained no valid segments at all due to the abundance of noise, and were removed from the dataset in use for processing. In addition, none of the papers cited studied the uterine contractions measured by the CTG, so this information was removed as well.

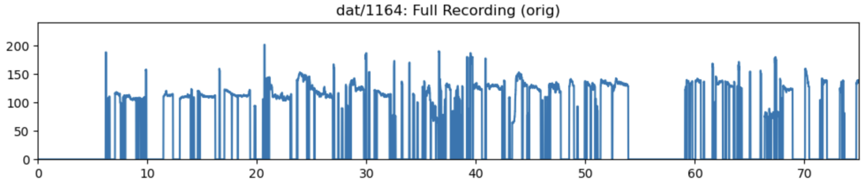


Fig. 2: Sample of Invalid Signal

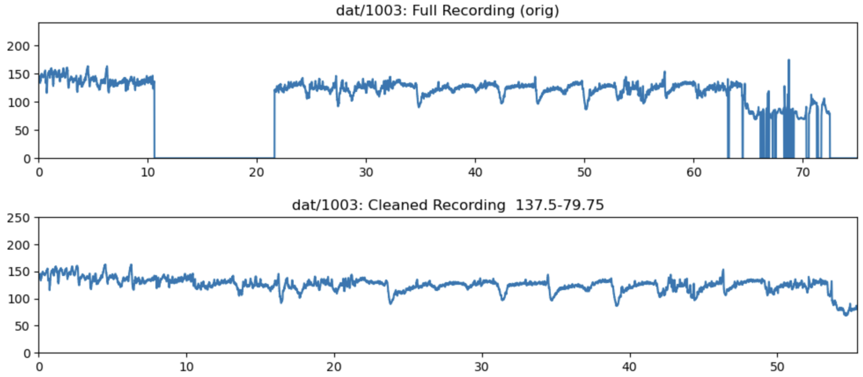


Fig. 3: Example of Signal Before and After Processing

### 3.3 CTU-CHB Image Representation

We experimented with several representations of the waveform as images:

**Spectrogram** (Fig 4a)

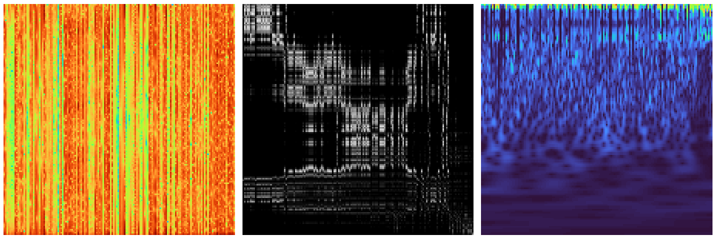
By visualizing the frequency of a waveform over time using a Short-time Fourier transform, a spectrogram displays how the frequency components of a signal change over time. These spatial changes can then be picked up as patterns by a model and used as features.

**Recurrence Plot** (Fig 4b)

Analyzing the recurrence of a system and capturing complex patterns in the time series data, a recurrence plot can generate images that are spatial representations of features that are close together in phase space.

**Continuous Wavelet Transform** (Fig 4c)

Instead of applying a Fourier transform, a CWT continuously translates and dilates across the time domain to represent the similarity between the signal and the wavelet at different times and frequencies.



(a) Spectrogram (b) Recurrence Plot (c) CWT

Fig. 4: 2D Data Representations

### 3.4 CTU-CHB Deep Convolutional Neural Network

CNNs, or convolutional neural networks, are robust neural network architectures designed to apply layers onto data to extract features and can be used for classification tasks as shown in Fig. 5. Deep CNNs utilize deeper layers for a more complex design capable of improved performance but are also a risk of overfitting. This type of architecture was similarly used in [7].

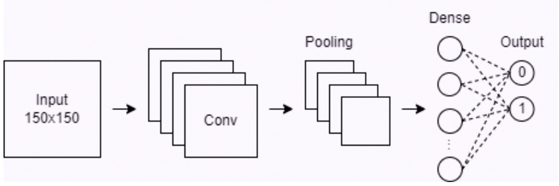


Fig. 5: CNN Diagram

A general equation for a single layer in our model can be expressed as follows:

$$\text{Conv}_i = \text{ReLU}\left(\sum_{j=1}^{n} (\text{input} * \text{filter}_{ij} + bias_{ij})\right) \tag{1}$$

Where $Conv_i$ is the output of the i-th convolution filter, $n$ is the number of filters, $*$ is the convolution operation, and the bias is added to each feature map. The whole output is passed through the ReLu activation function for a nonlinear transformation.

After generating the data into an image format, the DCNN was trained on the three sets of images to predict two classes, where an entry is labeled as possible hypoxia (class 1) if it meets any of the following criteria and class 0 otherwise:

- pH $<$ 7.15
- BDecf (Base Deficit) $>$ 10 mEQ/L
- pCO2 (Partial Pressure of Carbon Dioxide) $>$ 10 mmHg
- BE (Base Excess) $<$ -10 mEq/L
- Apgar Score (after 5 mins) $<$ 7

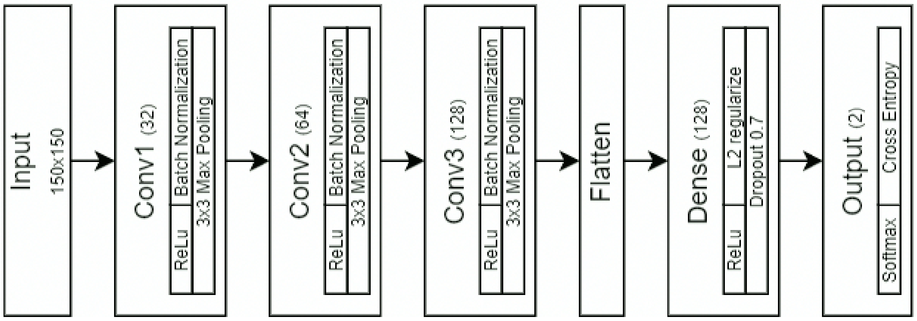


Fig. 6: Model Architecture

An Adam optimizer was chosen with $\eta = 0.0001$. Adam is a version of stochastic gradient descent and adapts to data characteristics well by tracking the moving average of the gradient of the objective function, where momentum circumvents convergence into suboptimal local minima.

The architecture in Fig. 6 first comprises of two convolutional layers and 32/64/128 filters each, followed by ReLu (Rectified Linear) activation functions. The first layer takes as input a 150x150 image. After flattening to a 1d vector, a dense layer with 128 nodes and ReLu is used, with a l2 loss regularizer = 0.001. A dropout rate of 0.7 is applied to the dense layer to combat overfitting, which randomly selects nodes to disregard.

The final output layer has a softmax activation layer with 2 classes and returns a probability distribution for each class. The softmax equation is as follows, where $x_i$ is the $i$th element of the vector:

$$\text{Softmax}(x_1, x_2) = \frac{e^{x_1}}{e^{x_1} + e^{x_2}}, \quad \text{Softmax}(x_2, x_1) = \frac{e^{x_2}}{e^{x_1} + e^{x_2}} \tag{2}$$

The class with the highest probability is chosen as the prediction. The DCNN was trained on the dataset in the form of spectrograms, recurrence plots, and continous wavelet transform, and the results are documented in the Results section.

### 3.5 Feature Extraction

Analyzing cardiotocograph (CTG) tracings is essential for monitoring fetal well-being throughout pregnancy and childbirth. In this practice, fetal heart rate (FHR) signals are measured and interpreted into quantifiable data to record and monitor so that fetal health is ensured. Therefore, in accordance with the Royal College of Obstetricians and Gynecologists (RCOG) guidelines, feature extraction was performed on the FHR data to tabulate the CTG waveforms.
The following features were extracted using their proposed methods in accordance with RCOG guidelines.

**Baseline**: The baseline measure of the FHR signal.
The Baseline value $R$ is measured as the mean signal value of the FHR signal

given by:

$$R = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{3}$$

where $N$ is the total number of signals and each $y_i$ is a given CTG signal.

**Accelerations**: An increase in FHR from the baseline.
An acceleration as defined by the RCOG, is "an improvement in FHR of at least (15 b.p.m.) from the baseline that is sustained at that level or greater for at least 15 s."[9] This implies that in a given segment between two intersections, $X_1$ and $X_2$, from the baseline and FHR signal, the time difference $X_a$, is at least 15 seconds.

$$X_a = X_2 - X_1 >= 15(seconds) \tag{4}$$

As well as the highest peak in the given segment $Y_{max}$ is at least 15 b.p.m more than the baseline R.

$$Y_a = Y_{max} - R >= 15(b.p.m) \tag{5}$$

where $Y_a$ is the difference between $Y_{max}$ and R.

**Decelerations**: An decrease in FHR from the baseline.
Similarly to accelerations, a deceleration as defined by the RCOG, is "Transient episodes of slowing of FHR below the baseline level of more than (15 b.p.m.) and lasting (15 s) or more."[9] This implies that in a given segment between two intersections, $X_3$ and $X_4$, from the baseline and FHR signal, the time difference $X_d$, is at least 15 seconds.

$$X_d = X_4 - X_3 >= 15(seconds) \tag{6}$$

As well as the lowest dip in the given segment $Y_{min}$ is at least 15 b.p.m less than the baseline R.

$$Y_d = R - Y_{min} >= 15(b.p.m) \tag{7}$$

where $Y_d$ is the difference between R and $Y_{min}$.
Should the deceleration last for 120 seconds or 2 minutes, this would classify the deceleration as a Prolonged Deceleration.

**Variability**: FHR fluctuations within a given timeframe.
As defined by the RCOG, "The baseline variability calculation is obtained by calculating the highest (Ymax) and lowest (Ymin) values of the FHR signal in a 2-min segment."[9]

$$V = Y_{max} - Y_{min} \tag{8}$$

where $V$ is the variability value of the FHR signal in any given timeframe. To align with the kaggle dataset, variability was split into 2 groups, short term variability and long term variability, where short term was defined as 1 minute intervals and long term was defined as 5 minute intervals.
Furthermore, each variability was taken into account of the percent of time the FHR signal sustained abnormal variability such that the variability $V$ was less than 5 b.p.m or greater than 25 b.p.m[10].

# 4 Results

## 4.1 CTU-CHB DCNN Results

The DCNN was trained on the CTU-CHB dataset recurrence plots at 20 epochs, and the results are shown in Table 4. Compared to the spectrogram and the cwt plots, the recurrence plots performed better on average, likely due to optimal extraction of frequency patterns of the fetal heart rate, resulting in stronger predictive power.

Table 3: Average training metrics per image type

| | Accuracy | f1_score | Precision | Recall |
|---|---|---|---|---|
| recurrence | 0.832 | 0.791 | 0.832 | 0.832 |
| spectrogram | 0.792 | 0.747 | 0.792 | 0.792 |
| cwt | 0.745 | 0.694 | 0.745 | 0.745 |

Table 4: Average validation metrics per image type

| | Accuracy | f1_score | Precision | Recall |
|---|---|---|---|---|
| recurrence | 0.754 | 0.485 | 0.754 | 0.754 |
| spectrogram | 0.730 | 0.548 | 0.730 | 0.730 |
| cwt | 0.560 | 0.462 | 0.560 | 0.560 |

It should be noted that with more epochs, validation and training loss differ substantially. This is likely due to overfitting as the model begins to memorize the dataset. In Fig. 7, the model was trained to 50 epochs to visualize the losses intersecting and diverging.

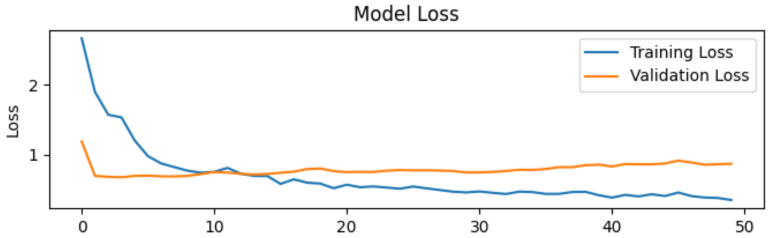


Fig. 7: Train and Test Loss per epoch

Table 5: Model Performance Metrics on Test Dataset

| Metric | Test Results |
|---|---|
| Accuracy | 0.7835 |
| F1 Score | 0.5453 |
| Precision | 0.6072 |
| Recall | 0.5460 |

Testing the DCNN on an unseen test dataset yielded the following results in table 2. Although achieving an accuracy of 78%, the model appears to often label false positives and false negatives, as indicated by the low f1-score, precision, and recall. This is possibly an adverse effect of rebalancing class weights and class imbalance.

We compared our results with [7], which followed a similar methodology and generally achieved higher results on average. We reason this is likely due to more preprocessing on separate frequency intervals, a more complex CNN model, and large input images allowing for more features.

### 4.2 CTU-CHB Feature Extracted Results

After extracting the features from the CTU-CHB data, we performed the same machine learning techniques as previously discussed in section 3.1 to classify the fetal health of each signal. However, we were not able to draw any reasonable conclusions about the data as the predicted classes of each signal were not consistent through each model and the mono classification of each method seemed incorrect as the average balanced accuracy of each method was around .8.

As we can see in Figure 4, the Gaussian Naive Bayes model predicted the entire dataset to be class 3, Pathological.

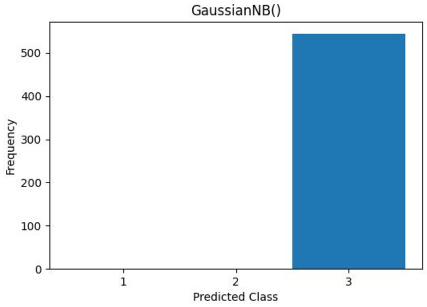


Fig. 8: Predictions made by GaussianNB model on CTU-CHB dataset

Whereas in Figure 5, we see that each respective model, Decision Trees, Random Forest, and K Nearest Neighbors, all classified every signal in the dataset to be class 1, Normal.

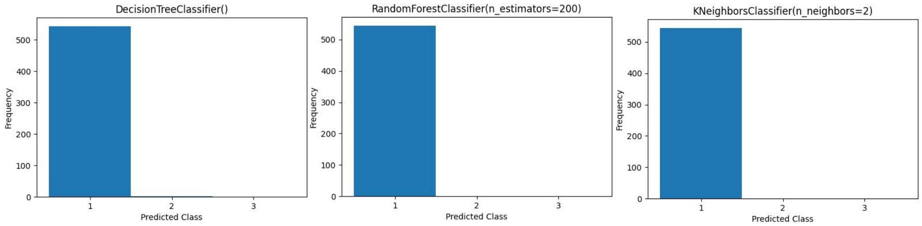


Fig. 9: Predictions made by Decision Trees Classifier, Random Forests Classifier, and K Nearest Neighbors Classifier on CTU-CHB dataset

A reason for these results is the data inconsistencies between the Kaggle data and the CTU-CHB data. When performing Feature Extraction as discussed in section 3.5, there was insufficient research to properly correspond the CTU-CHB data to the selected features in the Kaggle data. For example, the Kaggle data separates decelerations into two categories, light and severe. As we were unable to conclude how to categorize the decelerations into these categories, we opted to merge them into one category, decelerations, especially since the Kaggle data provided 0 records of any severe decelerations.

Furthermore, the variability parameters varied heavily. As normal variability stands between 5 and 25, which was represented in the CTU-CHB dataset, the Kaggle dataset had a mean short term variability trending towards 0.2. All these factors are reasons for why we think the results from the Machine Learning methods on classifying the CTU-CHB dataset is inconclusive.

## 5 Accomplishments

Through this project, we were able to gain a better understanding of Machine learning algorithms and how to utilize them for classification problems.

We initially had difficulties working with the CTU-CHB data, as it was in a waveform format that standard libraries could not read properly, but then we learned how to work with waveform data using the waveform toolkit.

Throughout the whole project, we worked on fetal health data and CTU-CHB data and found out the key factor that predicts the health of a fetus is its mean value of short term heart rate variability based on the Chi-squared test. Therefore we used different machine learning methods and built up models to detect the health status of fetuses. Besides, our DCNN model is also able to analyze fetuses' heart rate signals, generate recurrence plots, spectrograms or waveform signals, and give out the result of the fetus's health state. By completing our project, we make the fetal health detection technology come true and the beneficial influence of our project will cover millions of unborn babies across the world.

## 6 Contributions

- **Angelina Xia**: Installed TIGER library and figured out that was a project dead end, basic machine learning computations for tabulated data set, searched on PUBMED and google scholar for directions to take the project in, figured out waveform processing toolkit and data preprocessing by extending code found on williamsdoug repository, met with professor and TAs for project help.
- **Mitchell Liu**: Researched how to tabulate waveform data. Implemented random forest and feature selection code. Met with TA to discuss the project and find the williamsdoug repository.
- **Ryan Wu**: Trained neural network for kaggle data with chi-squared testing. Generated spectrogram and cwt from preprocessed waveforms. Created the DCNN model architecture, performed training on converted CTU-CHB

images to generate results, and optimized parameters and class reweighting to improve learning.

- **Jerome Lau**: Trained machine learning models on kaggle data. Recevied all machine learning methods to aggregate into one file. I received preprocessed fhr signals from Angelina to perform feature extraction. Used extracted features from CTU-CHB dataset to perform machine learning classification.
- **Jiachen Sun**: Training data with several models for machine learning purposes. Researched on the recurrece plot function to generate the recurrence plots with processed CTU-CHB data for DCNN model training. Update the group project github page.

## 7 Conclusion and Discussions

For tabulated data, the Random Forest machine learning model produced the best results compared with other machine learning models. We speculate that this is because Random Forest models are better at handling unbalanced data and higher dimensional data, which was a feature of the Kaggle dataset.

For our work on the CTU-CHB dataset, we found that generating recurrence plots produced an improvement in accuracy over spectrograms and continuous wavelet transforms, likely because recurrence plots are better at catching changes and correlations in temporal patterns over spectograms and continuous wavelet transforms. We were not able to achieve a high degree of accuracy compared with the papers that we studied, likely due to weaker preprocessing and CNN modeling.

We attempted to do feature extraction to compare the Kaggle dataset as well as the CTU-CHB dataset using machine learning methods, but because of the differing nature of the data we were not able to draw any meaningful conclusions from the CTU-CHB dataset, likely due to the much less annotated nature of the data.

## 8 Future Work

Should we continue to work on this project or for other students to pick up where we left off, we would like to implement a UI. This way, there would be an application for users to input their data in the form of tabular data or waveform signals and receive classification for the health of the fetus.

For the data preprocessing, it would be interesting to see if spline interpolation would make a noticeable difference versus linear interpolation on the data results. In addition, we mostly compared different forms of generated images in terms of using a CNN, but it would be useful to see for ourselves how different the results of machine learning versus deep learning methods would be on these images.

## Acknowledgements

Much of our research direction came from the paper ”Computer-Aided Diagnosis System of Fetal Hypoxia Incorporating Recurrence Plot With Convolutional Neural Network” [4], which pointed us in the direction of using recurrence plots and neural networks for our analysis. The github repository by Doug Williams ”ML/DL analysis of Cardiotocography (CTG) traces using Recurrence Plot” [8] was also instrumental in helping us with the data preprocessing and recurrence plot features.

## Appendix

All of our code and instructions on how to reproduce our results can be found on our GitHub repo.(https://github.com/sfu-cmpt340/fetal-health-classification)

Source code for part of our data preprocessing can be found on Doug William's ”ML/DL analysis of Cardiotocography (CTG) traces using Recurrence Plot”. [8] The altered code as used in our project is on our github.

## References

1. Andrew M. and Other Contributors, Fetal Health Classification Dataset, https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification. Last accessed Mar 31, 2024
2. Václav Chudácek, Jirí Spilka, Miroslav Burša, Petr Janku, Lukáš Hruban, Michal Huptych and Lenka Lhotská: CTU-CHB Intrapartum Cardiotocography Database, https://physionet.org/content/ctu-uhb-ctgdb/1.0.0/. (2014) Last accessed March 31, 2024
3. Yahui Xiao, Yaosheng Lu, Mujun Liu, Rongdan Zeng, Jieyun Bai: A deep feature fusion network for fetal state assessment. Frontiers in Physiology, Volume 13 https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2022.969052/full. (2022)
4. Zhidong Zhao, Yang Zhang, Zafer Comert, Yanjun Deng: Computer-Aided Diagnosis System of Fetal Hypoxia Incorporating Recurrence Plot With Convolutional Neural Network. Frontiers in Physiology, Volume 10 https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2019.00255/full. (2019)
5. Zhidong Zhao, Yang Zhang, Yanjun Deng: A Comprehensive Feature Analysis of the Fetal Heart Rate Signal for the Intelligent Assessment of Fetal State. Journal of Clinical Medicine, Volume 7 https://www.mdpi.com/2077-0383/7/8/223 (2018)
6. Jun Ogasawara, Satoru Ikenoue, Hiroko Yamamoto, Motoshige Sato, Yoshifumi Kasuga, Yasue Mitsukura, Yuji Ikegaya, Masato Yasui, Mamoru Tanaka, Daigo Ochiai: Deep neural network-based classification of cardiotocograms outperformed conventional algorithms. Scientific Reports, Volume 11 https://www.nature.com/articles/s41598-021-92805-9 (2021)
7. Cömert, Z., Kocamaz, A.F: Fetal Hypoxia Detection Based on Deep Convolutional Neural Network with Transfer Learning Approach. In: Silhavy, R. (eds)

Software Engineering and Algorithms in Intelligent Systems. CSOC2018 2018. Advances in Intelligent Systems and Computing, vol 763. Springer, Cham. (2019). https://doi.org/10.1007/978-3-319-91186-1_25

8. williamsdoug: ML/DL analysis of Cardiotocography (CTG) traces using Recurrence Plot, https://github.com/williamsdoug/CTG_RP/tree/master. Last accessed March 31.
9. Al-Yousif, Shahad and Najm, Ihab A and Talab, Hossam Subhi and Al Qahtani, Nourah Hasan and Alfiras, M and Al-Rawi, Osama YM and Al-Dayyeni, Wisam Subhi and Alrawi, Ali Amer Ahmed and Mnati, Mohannad Jabbar and Ghabban, Fahad and others: Intrapartum cardiotocography trace pattern pre-processing, features extraction and fetal health condition diagnoses based on RCOG guideline, Volume 8 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9454876/ (2022)
10. Lewis Potter: How to Read a CTG, https://geekymedics.com/how-to-read-a-ctg/ (2023)
11. J. Spilka a, V. Chudáček a, M. Koucký b, L. Lhotská a, M. Huptych a, P. Janků c, G. Georgoulas d, C. Stylios d:Using nonlinear features for fetal heart rate classification. Science Direct, Vol 7, (2012) https://doi.org/10.1016/j.bspc.2011.06.008
12. Lukáš Hruban, Jiří Spilka, Václav Chudáček, Petr Janků, Michal Huptych, Miroslav Burša, Adam Hudec, Marian Kacerovský, Michal Koucký, Martin Procházka, Vladimír Korečko, Jan Seget'a, Ondřej Šimetka, Alena Měchurová, Lenka Lhotská:Agreement on intrapartum cardiotocogram recordings between expert obstetricians. Journal of Evaluation in Clinical Practice, Vol 21 (2015) https://doi.org/10.1111/jep.12368
13. Shishir Dash; J. Gerald Quirk; Petar M. Djurić :Fetal Heart Rate Classification Using Generative Models. IEEE Xplore, Vol 61, (2015) https://doi.org/10.1109/TBME.2014.2330556
14. Zafer Cömert a, Adnan Fatih Kocamaz b, Velappan Subha c :Prognostic model based on image-based time-frequency features and genetic algorithm for fetal hypoxia assessment. Science Direct, Vol 99, (2018) https://doi.org/10.1016/j.compbiomed.2018.06.003