Bachelor thesis
Computing Science

Radboud University

Separating the Heartbeats of Twins during Pregnancy using Random Forest

Author:
Daan Derks
s1011515

First supervisor/First assessor:
dr.ir. R. (Rik) Vullings
rik.vullings@ru.nl

Second assessor:
prof. T.M. (Tom) Heskes
t.heskes@science.ru.nl

January 13, 2021

Abstract

Monitoring of twin pregnancies is very important, because they are vulnerable for severe diseases that eventually can lead to stillbirths or neonatal deaths. Monitoring is typically done by using a cardiotocogram (CTG). The main component of the CTG is the foetal heart rate (FHR). Unfortunately, current methods for obtaining the FHR perform poorly in twin pregnancies, mostly because of they are higly vulnerable to motion. If the foetuses were to move around in the maternal abdomen, the FHR measurements are likely to fail or to confuse foetus 1 and foetus 2, and vice versa. A possible better technique to obtain the FHR of two foetuses in a twin pregnancy exploits non-invasive foetal electrocardiography (NI-fECG). NI-fECG works by measuring the electrocardiogram (ECG) from multiple locations on the abdomen at the same time. Separation of the heartbeats of the siblings is very hard, and no computerized methods yet exist. In this thesis, a random forest classifier (RFC) is trained using features of the ECG signals with the aim of separating heartbeats from both twins. Unfortunately, it is impossible to assign unique labels to heartbeats across all pregnancies. In other words, we cannot assign some heartbeats to foetus 1 and other heartbeats to foetus 2, as in another measurement on another patient, these foetuses are no longer the same and the RFC would perform poorly. As an alternative strategy, an approach is opted were the RFC classifies the similarity between pairs of heartbeats. If the pair is similar, the heartbeats come from the same foetus; if they are dissimilar they come from different foetuses. After training, the RFC achieves an accuracy of 95.81% for classifying the similarity of the heartbeats on an unseen test set. Despite this relatively high accuracy, the remaining errors in the classification can lead to sustained errors in the resulting FHRs. If one particular heartbeat was assigned to the wrong foetus, the following heartbeats might be correctly classified as being (dis)similar to that particular heartbeat and as a result also assigned to the wrong foetus. In this research is demonstrated that this effect is larger for cases where the FHRs of both foetuses are close to another, as compared to measurements where the FHRs are more clearly separated.

Contents

1 Introduction

Twin pregnancies are vulnerable for mortal diseases. They account for approximately 3% of all live births, but account for 6.3% of stillbirths and 12.7% of neonatal deaths [5]. Monochorionic twin pregnancies have a high potential to develop the morbid conditions of twin-twin transfusion syndrome, twin anemia polycythemia sequence, or twin oligohydramnios-polyhydramnios sequence [6]. To reduce the risk of such complications, monitoring of the condition of both foetuses is of vital importance. Monitoring of the unborn child during pregnancy is done by means of the cardiotocogram (CTG), a simultaneous registration of foetal heart rate (FHR) and uterine activity. Antepartum cardiotocography is typically performed via Doppler ultrasound (DU). Here, a DU probe is placed on the maternal abdomen and a small region within the abdomen is insonified by the probe. The ultrasound that is reflected from the foetal heart is subsequently used to determine the FHR. Currently, monitoring of twin pregnancies is done by using multiple DU probes, that produce a CTG for each of the siblings. Unforunately, DU-based cardiotocography suffers from strong limitations. When a foetus would move relative to the DU probe, it might move outside of the narrow DU beam and consequently no FHR information is available. For twin pregnancies, two DU probes are required, not only raising the chance that at least one of the foetuses might move outside its DU beam, but posing additional challenges. For example, when moving outside its DU beam, a foetus might move into the DU beam that is meant for its sibling, leading to unreliable FHR monitoring for that fetus. As a result, monitoring of twin pregnancies in clinical practice is challenging and technological solutions to improve this are critically needed.

Non-invasive foetal electrocardiography (NI-fECG) is a candidate to solve the main issues with twin monitoring. NI-fECG uses multiple electrodes, possibly combined in a single patch, on the maternal abdomen to monitor both foetal and maternal heart rate as well as uterine contractions [7]. Every electrode records a mixture of signals, of which the foetal electrocardiogram (ECG) is one. More specifically, the mixture comprises of signals originating from the mother (e.g. maternal ECG, abdominal electromyogram, electrohysterogram), from extracorporal sources (e.g. mains powerline), and the ECG from both foetuses. The shape of the foetal ECG depends on the position of the foetus relative to the position of the electrode. Because both foetuses cannot lie in the same position, the foetal ECG of both siblings in each electrode must be different. Because of those differences, the hypothesis is that the heart rates of the two siblings can be separated with the help of machine learning. In this research, the random forests machine learning method will be used because it is one of the most accurate learning algorithms available. Consequently, this research determines how accurate classification methods such as random forest can separate heartbeats from both siblings and determine the FHR of both foetuses in twin pregnancies.

This thesis is structured as follows. First, the challenges and limitations of the current methods of measuring a twin pregnancy and some explanation about the random forest classifier are given. Subsequently, there will be a description about possible features to enable learning by our classification algorithm. In the section Materials and Methods is discussed how a model is created using the random forest classifier and which dataset is used for training and testing my created model. In the section Results the findings of my research of the previous section are presented. These results will be compared to those of related work to gauge the performance of the random forest classifier. Furthermore, an evaluation about this thesis is given and some recommendations for further research are stated. At the end, a conclusion about the findings of this thesis is written down.

2 Background

It is meaningful to measure the FHR of a sibling because it reveals information about its well-being. Usually a foetus of 18 to 21 weeks has an average FHR between 110 and 160 beats per minute [4]. If the FHR deviates significantly, it may indicate the foetus is not getting enough oxygen or other problems are occurring to the foetus. In this section, the current measurement methods that are used for measuring the FHR of twin pregnancies and the challenges and limitations they bring are discussed. Also, a brief explanation why the random forest classifier is selected is given and how it is able to create a predicting model.

2.1 Cardiotocography

There are two methods to record the CTG: external cardiotocography and internal cardiotocography. Both measure the FHR and the uterine contractions during pregnancy.
External cardiotocography (figure 1) uses a device that is placed on the abdomen of the mother. This device consists of a tocodynamometer which measures the uterine contractions and a DU transducer which measures the FHR. DU uses high-frequency sound waves that can measure the blood flow through the arteries and veins. The pulsatile character of variations in this blood flow can be used to determine the FHR and displays this as a CTG. However, this method is vulnerable for movements of the foetus, because the DU tranducer can lose the signal of the foetal heart.

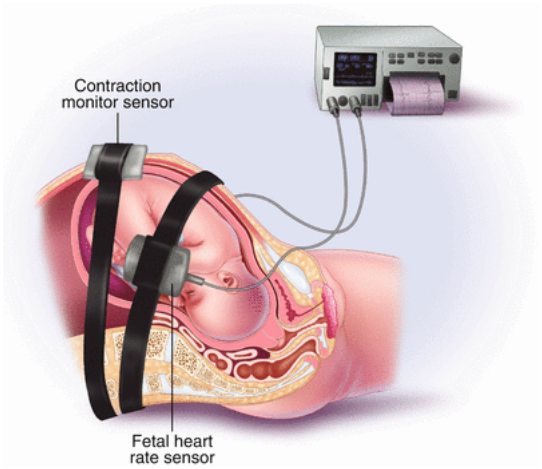

Figure 1: External cardiotocography

As alternative to external cardiotocography, internal cardiotocography can be used. This method uses an electronic transducer that is connected directly to the foetus. An electrode is attached to a body part of the foetus, as been illustrated in figure 2. To find the foetus, the electrode must go through the cervical opening. This method can measure the FHR more accurately than external cardiotocography, because it is directly connected to the foetus. Therefore, it is not vulnerable for movements of the foetus. However, a disadvantage of this method is that it can lead to infections and it can only be applied during labour, after rupture of the foetal membranes and sufficient cervical dilatation. For monitoring the foetus in earlier stages of pregnancy than labour, internal cardiotocography is not an option.

External cardiotocography is commonly used for singleton pregnancies, but it can also be used for twin pregnancies. Both foetuses need their own DU transducer which monitors their hearts. But several things can go wrong when using multiple transducers. As mentioned earlier, they are vulnerable for movements of the foetus. During twin pregnancy, the foetuses also move around in the uterus. This makes it hard to find the hearts of the foetuses. Secondly, the transducers may measure the same heart. If the doctor finds good signals on the transducers, it is possible that both the transducers receive the same signals of the heart that belongs to one foetus. Therefore, the other foetus stays unmonitored. It is also possible that the two hearts are exactly inline with the transducer. Due to this, the transducer receives signals from both hearts. The signal it receives is disturbed by a lot of noise from the other heart. It is very difficult to know when the transducers are in the correct place because it is also possible that the transducers are measuring the correct hearts but the FHR is the same.

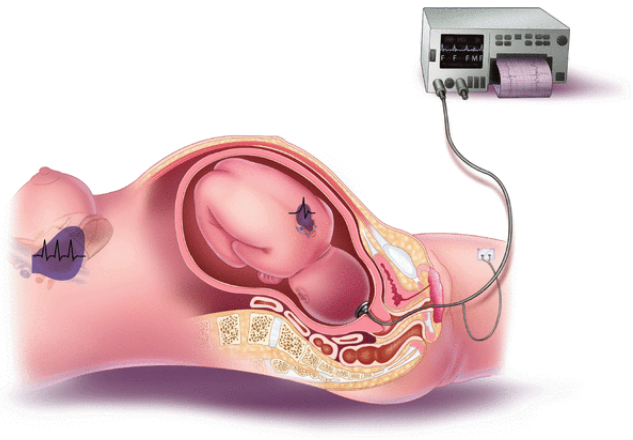

Figure 2: Internal cardiotocography

2.2 Non-invasive foetal electrocardiography (NI-fECG)

A more accurate method to measure the heart rates of twins during pregnancies could be non-invasive foetal electocardiography (NI-fECG). NI-fECG uses multiple electrodes on the abdomen of the mother. In this case, 8 electrodes are used: six channels, one ground (GND) and one common reference (REF). The ground electrode is used to level possible potential differences between mother and measurement device and prevent saturation of the input range of the measurement device, which would lead to clipping of the recorded signals. Each of these six channels measures differences in voltage between the corresponding electrode and reference electrode. These voltage differences are caused by the propagation of electrical currents, generated by the heart while it is beating, all the way to the cutaneous surface. When plotted as function of time, these voltage differences represent the ECG. Because this propagation of electrical currents is virtually instantaneous, the ECG from one heart is measured by all channels at the same time. The ECGs of the other hearts arrive at different times, because the hearts are each beating at their own pace. Moreover, because the electrodes are placed on different positions on the maternal abdomen, the shape of the ECG of one heart is different per channel. In figure 3, channel 4 and channel 6 are displayed with their ECG. In channel 4, foetus 1 ($F1$) is stronger than foetus 2 ($F2$). In channel 6, it is exactly the other way around, foetus 2 is stronger than foetus 1. Naturally, the ECGs of different hearts also have a different shape.

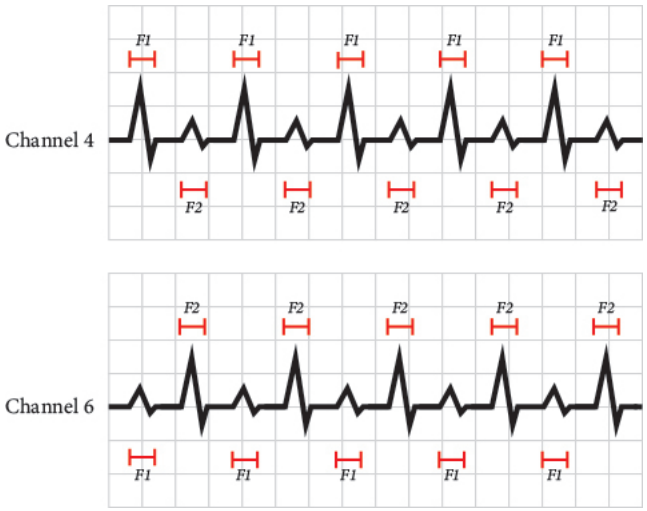

Figure 3: An illustration of a foetal ECG highlighting channel 4 and 6. These channels are recorded simultaneously with multiple electrodes on the maternal abdomen. In one of the channels (Channel 4) the peaks from foetus 1 ($F1$) are stronger and in the other channel (Channel 6) the peaks from foetus 2 ($F2$) are stronger.

It has been shown in singleton pregnancies that NI-fECG is much more accurate than external cardiotocography, because it measures on different places on the abdomen synchronous [9]. In theory, the method could also be used to measure the FHR for twin pregnancies. By exploiting the fact that the recordings are performed with multiple electrodes on multiple positions on the maternal abdomen, channels could be selected where the heartbeat of foetus 1 is observed more explicitly than foetus 2 (and vice versa). For instance, because the foetus is much more closely positioned to that specific electrode. Subsequently, the FHR per twin could be detected from the ECG of these particular channels.

Unfortunately, no computerized methods exist so far to assign each of the recorded channels to a specific foetus. Moreover, due to possible foetal movement, this assignment of channels to foetuses needs to be done continuously; assigning them at the start of a recording will not suffice.

2.3 Random Forest Classifier

Computers have surpassed the human brain on different fronts in the past decades. They are much faster in solving hard computations than our brains. Thus, comparing a lot of data is much more efficient on a computer. As mentioned before, in this research a classification algorithm is used to separate the FHR of both twins in NI-fECG recordings. A classification algorithm is an al-

gorithm that can classify data into two or more classes. Here, each of these classes represent a different mechanism or source from which the data was generated. For instance in the case of twin pregnancies, the classifier could classify all detected heartbeats as originated from the one foetus (class 0) or the other foetus (class 1). Unfortunately, it is difficult to create an algorithm that will always give the correct result, so making such an algorithm is one of the biggest challenges of this research.
When developing a classification algorithm, a model is created and trained to enable it to predict classes. Different types of methods for making such a model can be used; a Random Forest Classifier is used for this research.
A Random Forest Classifier (RFC) creates a model that consists of multiple decision trees. Those trees each independently result a decision in which class the input data belongs to. To generalize the decisions, the classifier takes the majority vote of all the decisions that were made by the trees in the forest. Thus, the better the individual decision trees predict the correct class, the more significant the accuracy of the RFC will be.
To create the best possible classifier, the model needs to learn from a large and representative dataset. The dataset that is used to train the model is called the train set. In other words, the model is trained on the train set to optimize the decision trees. To facilitate the optimization of these trees, relevant characteristics of the dataset are determined and used as input to the RFC. These characteristics are called features. Those features are very important to get a good classification. Section 3 digs deeper into the definition of features and which features are chosen for the problem of this research.
A decision tree comprises of decisions about the various feature. Every decision (e.g. whether the feature is smaller or larger than a certain value) is one of the nodes of the tree. The amount of branches a node has depends on the amount of choices the decision has. So for example the decision could be: What color has a specific dog and the possible choices could be grey, brown, black or white. So this node would have 4 different branches. Each of those branches could direct to another node or to a leaf. If it directs to a node, another decision must be made. If it directs to a leaf, all the decisions are done and the tree would give a solution, for example it would classify the dog as a Jack Russell, see Figure 4.
A RFC has a lot of different decision trees. Every time you fit a new train item into the model it will change and optimize the decision trees such that it results into the corresponding class.

A RFC is chosen as the machine learning algorithm, because it has shown good performance for a variety of problems. Advantages of RFC are that it is versatile, it is parallellizable, it handles high dimensional data, it removes automatically outliers and it handles unbalanced data. Drawbacks are that it is not interpretable (sort of black box), for large datasets it can cost a lot of memory and it can tend to overfit. This tendency to overfit can be avoided by tuning the hyperparameters of the classifier.

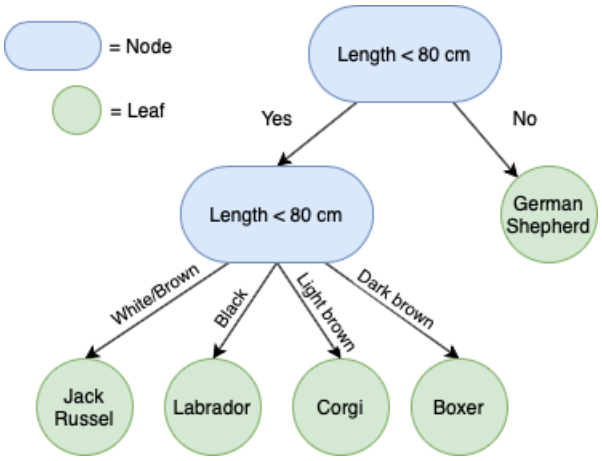

Figure 4: Example of a decision tree

3 Features in classification

In order to learn and recognize patterns, machine learning methods typically need features. A feature is an individual measurable property or characteristic of a phenomenon being observed [1]. The features are extracted from the data and used to train and test the model. In the example about classifying dogs breeds, the dataset consists of pictures of dogs. A good feature could be the length of the nose, the color of the coat or the size of the ears. In the case of the research problem the most useful features needs to be extracted for every peak, so that the model can correctly classify which peak belongs to which label; foetus 1 or foetus 2. The data where features need to be extracted from, consists of peaks of both foetuses. Those peaks are already detected, but from such peak the useful features needs to be extracted from the data. In this case the data is an ECG of a twin pregnancy. More information about the dataset can be found in section 4.1
As a first step, the data was observed to get an idea which characteristics seem the most useful for a good classification. The different features are subcategorized in two classes: those that can be rather easily observed from the data and those that relate more to mathematical but less intuitive features, referred to from hereon as visual features and mathematical features, respectively.

3.1 Visual features

Every foetal heartbeat leads to a so-called QRS peak in the electrical activity of the heart. This peak is displayed in each of the 6 recorded channels, but the amplitude of the peak differs in every channel. In some channels the peak might be clearly distinguishable, while in others, it is not. For the other foetus, this

amplitude distribution over the channels will likely be different. Unfortunately, it is not possible to know beforehand which channels show the amplitudes that allow for reliable classification of the peaks belong to either foetus 1 or foetus 2 . Therefore, all features are computed for each of the channels and the Random Forest Classifier is used to assess which channels might be the most relevant. Upon examining the recorded data, three features show the highest potential for reliable classification: the amplitude of the peak, the width of the peak and the sign of the peak. A peak that is pointed down is also a peak, but with negative sign. From hereon, the location of each peak is referred to as i, where i is the sample index of the peak in the recording. The recording itself is a multidimensional array consisting of six channels and is referred to from hereon as $channels[x][y]$ where x is a channel and y is a sample location.

The amplitude of a peak is in our case the distance between 0 (i.e. the baseline) and the height of the incoming signal at the time of the heartbeat. So, if the peak is pointed upwards and its summit is on 20, the amplitude is 20. If the peak is pointed downwards and its summit is on -30, the amplitude is 30.
The amplitude in each channel A is determined by taking the absolute value of the signal at location i : abs($channels[A][i]$). In some situations, the process of simulating twin measurements, yields NaN values in some of the channels. The RFC cannot do computations with NaN values, so if $channels[A][i]$ is equal to NaN, the amplitude of the peak at location i is set to 0.

The sign of a peak reflects whether the amplitude of the peak is positive or negative and is described here as a boolean. If the amplitude is positive or zero the sign is 1 and if the amplitude is negative the sign is 0.
Computing the sign of a peak at location i is similar to the calculation of the amplitude. As mentioned before, i and the multidimensional array $channels$ are known. To compute the sign of the peak at location i for channel A, in the multidimensional array is checked if $channels[A][i]$ is greater or equal to 0, if so the sign is 1 otherwise the sign is 0. The problem with NaN values is already solved for computing the sign, because the amplitude is already set to 0. So the sign of a peak with amplitude NaN is 1.

The width of a peak is the distance between the start and end of the QRS complex of the ECG. The QRS complex reflects the electrical activity of the contraction of the cardiac ventricles and comprises the main electrical activation of the heart. The QRS complex can be considered to consist of three individual waves: Q-wave, R-wave, and S-wave. The width of the QRS complex is defined as the distance between the peaks in the Q-wave Q and the peak in the S-wave S.
In case the R-wave has a positive deflection, the peak of the Q-wave is defined as the first local minimum before the R-wave. The peak of the S-wave is defined as the first local minimum after the R-wave. If the sign of the R-wave is negative, the peaks of Q and S are the first local maxima on either side of the R-wave.
Compared to the amplitude and sign of the peak, it is more complex to compute

the width of the peak, because the Q and S values of the peak that were detected at location i need to be computed for that peak. Another problem is, that the location i typically does not exactly point to the position of the R-peak; often the location of the R-peak is a few samples shifted with respect to i. For the detection of the width of the QRS-complex a method is followed that consists of two steps.
In the first step, the location i needs to be modified such that i is at the R-peak, as to ultimately enable the calculation of the values Q_{start} and S_{end}. To do this, two functions are created: `computeMaximum(data, peak)` and `computeMinimum (data, peak)`. For R-peaks with a positive sign, `computeMaximum` is used and it returns the modified i that points to the R-peak. The same for peaks with a negative sign, but then vice versa, `computeMinimum` is used and it returns the modified i that points to the R-peak (who is pointed downwards). Now the pointer to the peak is at the right location, the second step of my method can be proceeded.
In this second step, the location of the peaks in the Q-wave (Q) and S-wave (S) are determined. To compute Q, the computation begins at the location of the R-peak and goes one sample backwards, step-by-step. In case the R-wave has a positive sign, it evaluates per step whether the amplitude of the signal at location $i-1$ is larger than the amplitude of the signal at location i. If so, the location i is used as Q. If not, it proceeds with the next step until the condition is met. In case the R-wave has a negative sign, the computation checks for the condition where the amplitude of the signal at location $i-1$ is smaller than the amplitude of the signal at location i to find a local maximum. For finding S, a similar strategy is followed, but moving forwards from the location of the R-wave and evaluating the amplitude of the signal at location $i+1$ compared to the amplitude of the signal at location i. Once Q and S are known, the width of the peak can be calculated.

Not only the width of each peak seems a promising feature for classification, but also the distance between peaks could be a good feature for the RFC. The distance between peaks relates to the heart rate of the foetuses which is naturally different between siblings. Typically, the peak before and after the peak at location i are not from the same foetus than the peak at i. The heart rates of both siblings, albeit different, are both normally in the range between 110 and 170 beats-per-minutes. This reflects a distance between 273 samples ($\approx 0.55s$) and 176 samples ($\approx 0.35s$) for peaks from the same foetus. Likely, the peaks from the other foetus fall somewhere in-between. Logically, the distance between the peak at location i and the second peak before i is more relevant as this might reflect in the inter-peak interval (and thus also the heart rate) for a specific foetus. Similarly, the distance between the peak at location i and the second peak after i is also a promising feature.
To compute the distance, two functions are created: `distToPrev(`*peaks*, *a*`)` and `distToNext(`*peaks*, *a*`)` where *peaks* is the array with all the sample locations of the peaks, and a is a natural number. The function `distToPrev(`*peaks*, *a*`)` returns the distance between a peak p and a peak that is a peaks before p. The

function `distToNext(`*peaks*, *a*`)` works similar. It returns the distance between a peak p and a peak that is a peaks after p. If there are no a peaks before p, for `distToPrev(`*peaks*, *a*`)`, or no a peaks after p, for `distToNext(`*peaks*, *a*`)`, the distance is 0.

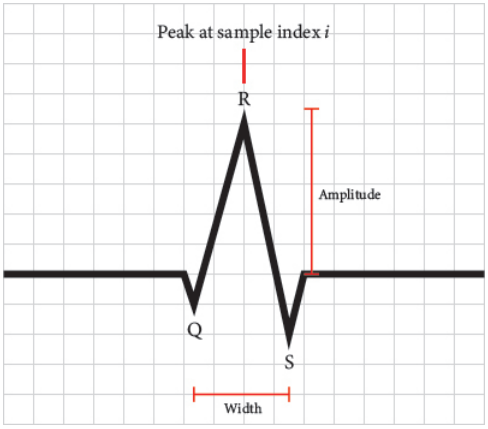

Figure 5: ECG Complex of a peak at sample index i

3.2 Mathematical features

As mentioned before, next to visual features, also mathematical features can add value to the classification of fetal heartbeats. Mathematical features used in this study are the standard deviation, skewness and kurtosis for every peak in every channel.

The standard deviation (SD) measures the variation in a set of values. If the SD is low, it means that all the values in the set are close to the mean of set. If the SD is high, they are spread out over a wider range. Computing the SD (σ) of set Z is done by:

$$\sigma_Z = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N+1}}$$

where N is number of elements in Z, $x_i \in S$ and μ is the mean of Z.

The skewness measures the asymmetry over a set of values. If the skewness is positive it means that it is right-skewed, negative skewness means it is left-skewed and a skewness of 0 means it is even skewed on both sides. The skewness is measured over a set Z by:

$$skewness_Z = \frac{\sum_{i=1}^{N}(x_i - \mu)^3}{(N-1) \times \sigma^3}.$$

The kurtosis is a measurement of the "tailedness" of a set of values. A high kurtosis means that the chance of extreme outliers is low. The formula of kurtosis over a set Z is:

$$kurtosis_Z = N \times \frac{\sum_{i=1}^{N}(x_i - \mu)^4}{(\sum_{i=1}^{N}(x_i - \mu)^2)^2}.$$

For all peaks in every channel are the SD, skewness and kurtosis computed. The set Z of datapoints is used to calculate these features is defined the following way. Consider a peak at location i in channel A. Z is now defined as the amplitudes of the signal within a window of length 41 that is centered around i. In other words:

$$\begin{aligned} Z &= \{channels[A][i-20], channels[A][i-19], \ldots, \\ &\quad channels[A][i+19], channels[A][i+20]\} \end{aligned}$$

The features now say something about the shape of the peak. If there are no 20 samples before i in channel A, all possible samples before i are taken and still include only 20 samples after the peak location, effectively shortening the length of the set Z. This has a similar approach for a peak with a sample location i where there are less than 20 samples after i.

4 Materials and Methods

A Random Forest Classifier needs a lot of data to learn a good model. This section describes which data is used to create the model and solve the separation problem. Furthermore, the methodology for implementing and training the classifier and how to evaluate the results are described in this section.

4.1 Dataset

My dataset consists of 1399 simulated recordings, generated in Matlab (The Mathworks, Natick MA, USA). This dataset is separated in 997 train files and 392 test files. Every file represents a simulated twin recording by combining two recordings that were done in singleton pregnancies with NI-fECG. This approach enables us to have ground truth labels about the class of the heartbeats that will be detected in the simulated twin data. These ground truth labels are crucial to train our model and optimize its classification performance. The twin data is generated by first removing the maternal ECG and some noise from each of the two singleton recordings and subsequently adding them to each other. For example, on the first sample, singleton pregnancy 1 receives a signal of 5 μV and singleton pregnancy 2 receives a signal of 25 μV, the simulated signal on the first sample becomes in the combined file 30 μV.

Each Matlab file consists of three arrays:

- a $[6 \times N]$ multidimentional array that contains the simulated data, where each row represents one of the six channels and N represents the number of samples of the recording.
- an array with the locations (which sample) of all peaks that belong to foetus 1
- an array with the locations (which sample) of all peaks that belong to foetus 2

With this dataset, all the six channels of the NI-fECG are available, the locations of all the heartbeats in those channels are known and all those locations have a label that specifies to which foetus the heartbeat belongs. The peak detection is not in the scope of this research; more information on the method for peak detection can be found in [10].

Overfitting the model is one of the key phenomenon when training a machine learning model [3]. When a model is overfitted, its performance on new data is typically poor, even if the performance on the training data would be very good. Overfitting can occur when the training data is not heterogeneous enough. This can lead to overestimation of the relevance of specific features. For example, let us consider a model that needs to distinguish dogs from other animals. However, the train data is not heterogeneous and only contains dogs of the same breed that have a very long nose. Then it is plausible that application of the trained model on a dog of another breed with a small nose would results in a misclassification by the model, because the model has learned from its training data that all dogs have long noses.
The dataset is on forehand splitted into a train and a test set. Two singleton pregnancy recordings are picked from a big collection of singleton pregnancy recordings. A single recording can be used in multiple simulated twin pregnancies (that each are a combination of two singleton recordings), but if a recording is used in the train set, it can never be used in the test set, and vice versely. This is to ensure that overfitting of the model to specific features from a single recording/patient cannot go by unnoticed.

There is also one file consisting a real twin pregnancy measurement using NI-fECG. This data file has, in contrast to the combined singleton pregnancies measurements, not a ground truth. In the Introduction it is stated that it is really challenging to monitor the two foetuses separately in the abdomen of the mother. For this data file, this has been attempted by using source separation techniques that make linear combinations of the recorded channels in such way as to maximize the amplitude of the foetal ECG of one foetus in one linear combination and maximize the foetal ECG of the other foetus in another linear combination [7]. The peak detection is subsequently performed twice, once for each foetus. This way, at least their is a plausible truth that can be used to assess the performance of my methods on separating the CTGs of twins.

4.2 Classification similarity learning

The first attempt to train the model was in a rather straightforward manner: by giving every peak the label of the corresponding foetus. For example: peak 1 is associated to foetus 1 and therefore gets the label 0, peak 2 is associated to foetus 2 and get the label 1, peak 3 has a label of 0, etc. But it was not possible to train the model in this way, because the model is trained using multiple files. For every file, it is known from the dataset which peaks have the label of foetus 1 and which the label of foetus 2. However, concatenating multiple files leads to a problem: foetus 1 of the first file is not the same as foetus 1 in the other files, because the different files all come from different mothers. The same also applies to foetus 2. This would complicate the classification, because the RFC is trained under the erroneous assumption that all peaks with the same label come from the same foetus. Classification similarity learning is a solution for this problem.

Classification similarity learning (CSL) can classify if a pair of objects is similar or dissimilar to each other. In the case of classifying peaks in pregnancy recordings, the CSL classifies whether a pair of peaks come from the same foetus or from different foetuses. To enable the training of the classifier using CSL, the features from two peaks are concatenated to yield the features of a pair. The label of the pair indicates the similarity of the pair. If the peaks are from the same foetus, the pair is similar and has label 1. If the peaks are from different foetuses, the pair is dissimilar and the label is 0.
In this case, concatenating the features or creating the similarity label is rather easy, but how to pick two peaks to create a pair is a bigger challenge. Creating pairs between different files will lead to the same problem that came across earlier, so the pairs have to be picked in the same file, so that it is certain that the similarity between the foetuses is correct. Also, picking two random peaks in a file is not the best way to create pairs, because of possible foetal movement. During monitoring, it is normal that the foetuses are moving in the uterus of the mother. Some features, in particular the amplitude and sign, will therefore change as a function of time. Taking a pair of peaks that are far away from each other in time, increases the chance that foetal movement has happened in between and therefore that the features are no longer distinctive enough to classify (dis)similarity.

Although CSL resolves the previously mentioned problem of having to classify peaks as belonging either to foetus 1 or foetus 2, ultimately this research wants to find out per peak whether it originated from foetus 1 or 2. Therefore, the pairs of peaks are classified by the RFC are selected such that it enables the assigning of individual peaks to one of the two foetuses. Initially, a number of pairs is chosen, called `trainPairs`, such that there are pairs created for a certain peak j equal to `trainPairs`. In other words, for the peak with index j, a few other peaks are selected that together comprise the `trainPairs`. For example, if the dimension of `trainPairs` is equal to 3, then 3 pairs are created

for peak j: $(j,\ j+1)$, $(j,\ j+2)$ and $(j,\ j+3)$. For testing and employing the RFC, it works similarly, but instead of variable `trainPairs` the variable `testPairs` is used.

As mentioned above, in this specific application, the similarity classification should ultimately allow to assign peaks to one of the two foetuses. For every peak that is classified by the RFC, a 3 dimensional tuple is created that contains information on the location i of the peak, as also explained in section 3.1, on which foetus the peak belongs to, and the probability of the classification. This probability is described as $p(A)$, meaning the probability of condition A to be true. This condition could for instance be whether the pair of peaks was similar.

If a new peak j needs to be assigned to a foetus, pairs are created with peaks that were already assigned to a specific foetus together with j and determine for these pairs whether they are similar or dissimilar. The amount of pairs is equal to the dimension of `testPairs`. For each of these pairs, a probability whether they are similar or dissimilar is determined by the RFC. Although in some situations these classifications might disagree, probability theory can still be used to determine the most likely foetus that the peaks belong to. Consider for instance the scenario where a new peaks is used in two pairs, one pair with a peak that was previously ascribed to foetus 1 and one with a peak that was previously ascribed to foetus 2. In this example, the RFC could give as a result that for both pairs the highest probability is that they are similar. However, one probability might be more conclusive than the other, i.e. 90% vs 51%. Moreover, the peaks that were previously ascribed to a certain foetus, might also have been described with less or more conclusive statistics. In the example above, if the pair that led to similarity probability of 51% was constructed using a peak that was ascribed to foetus 2 with only 51% certainty itself, chances that the new peaks belong to foetus 1 are much higher than that it belong to foetus 2. In our employment scheme, these probabilities are multiplied (and subsequently normalized) to assign a peak to a certain foetus based on the highest marginalized probability.
In this process, the probability of a classification is defined as the percentage of decision trees in the RFC that classified the pair according to one of the two labels. Logically, the probabilities of similar and dissimilar pairs sum to 1 for each peak. In other words, if the RFC determines the probability that a pair is similar. The probability of the pair being dissimilar can be computed as well and all marginalized probabilities for the peak belonging to either foetus 1 or foetus 2 can be calculated.
To illustrate the process of ascribing a peak to a specific foetus, let us consider the description below where peak i has already been ascribed to a certain foetus with probability $p_i(f_1)$. Here $p_i(f_1)$ is the probability that peak i belongs to foetus 1 and thus $p_i(f_1)+p_i(f_2)=1$, where $p_i(f_2)$ is the probability that peak i belongs to foetus 2. Now, consider a new peak k that needs to be ascribed to one of the two foetuses and the RFC returned a probability $p(s_{i,k})$ that the peaks i and k are from a similar class. The probability that the new peak k belongs

to foetus 1 is then $p_k(f_1) = p_i(f_1)p(s_{i,k})$. This same calculation is also done for other peaks j that were used in pairs with the new peak k. Consequently, multiple probabilities are obtained that can be combined via marginalization. To ensure that those probabilities still following the rules of probability theory, the probability $p_k(f_1)$ is normalized such that $p_k(f_1) + p_k(f_2)$ at all times remains equal to 1. If $p_k(f_1) > p_k(f_2)$, then peak k is ascribed to foetus 1 and a new 3-dimensional tuple is created with the sample location of peak k, the foetus to which the peak is ascribed and the probability of the classification.

In the initialization phase of this process, an assumption is made for the first N peaks in the ECG, where N is equal to the dimension of $testPairs$, because otherwise it is not possible to create pairs with peaks that were already ascribed to a specific foetus. For example, if peak k is the 20th peak in the sequence of all peaks and $testPairs$ has dimension equal to 25, it is not possible to compare with peak k with 25 peaks before k. Consequently, k needs to be one of the peaks that needs to be ascribed to a foetus a priori and for which the process described above cannot be used.

4.3 Monitoring the heart using a cardiotocogram

In the background section, it was mentioned that monitoring of the foetuses is very important. One of the best ways to monitor a foetus makes use of its FHR. The FHR is the number of contractions (beats) of the heart per minute (BPM: beats per minute). The heart rate is displayed as function of time in the CTG. The goal of this research is therefore not only to classify peaks to a specific foetus, but from this classification to determine the CTG for each of the foetuses.

In the subsection 4.2, an array is created that consists of all the detected peaks. Every peak in this array is described by a 3-dimensional tuple with sample location in the ECG, foetus to which the peak belongs to, and probability of the classification. By taking all peaks that are ascribed to foetus 1 and all peaks that are ascribed to foetus 2 separately, two new arrays can be determined that each belong to one of the two foetuses. From the information in these arrays, the heart rates of both foetuses can be determined.
The distance between consecutive peaks in an array reflects the interval T between two consecutive heartbeats. Considering a sampling rate of 500 Hz for our recordings, the FHR can be determined as:

$$FHR_k = \frac{60[s/min] \times 500[samples/s]}{T_k},$$

where T_k is the distance between peak k and its predecessor peak. For example, $T_k = 200$ means there are 200 samples between peak k and its predecessor $k-1$. Then the FHR at the time of peak k is $60 \times 500/200 = 150$ BPM.
Sometimes peaks are not correctly detected or are not detected at all, so there are outliers that are irrelevant for monitoring. Moreover, the heart rate cannot change abruptly on short time scales. Based on this, outliers can be detected as

physiologically unplausible variations in heart rate or as heart rates that fall too far from the normal range. After outlier detection (and correction by omitting these outliers), CTGs can be plotted as the sequence of FHR_k for both foetuses.

5 Results

In this section the research results will be provided. In this light, it might be good to repeat the main research question: How accurate can classification methods such as random forest separate heartbeats from both siblings and determine the FHR of both foetuses in twin pregnancies?
To provide an answer to this question, as a first step, evaluation what (hyper)parameters for the RFC and which features provide the best classification performance in the similarity learning is needed.

5.1 Parameter settings for RFC

For the implementation of the random forest classifier, the Scikit-Learn API [2] is used. This is a Python library consisting of machine learning algorithms. The RFC has a lot of parameters that can be optimized for my problem: The most important parameters are `n_estimators`, `max_depth` and `max_features`. The parameter `n_estimators` is the number of trees in the forest. Too many trees leads to overfitting the model, but to few trees can lead to a sub-optimal (i.e. underfitting) classification of the model. So, the optimal number of trees in the forest needs to be found. A forest of 100 trees is chosen; whether this number represents the optimal balance between under- and overfitting remains to be determined and should be part of future research. The `max_depth` parameter is the maximal depth of a tree in the forest. This parameter is set to 20 because setting it much higher did not change the results of the classifier, but it leads to much higher computing times. The last parameter that differs from the default is `max_features`. This parameter is equal to the maximal number of features that may be used for a node split in the tree. In this research, there is no restriction of the maximum number of features and allowed the RFC to use all the features that are created to train the model. In the Python listing below, the exact implementation of the RFC is detailed.

Listing 1: Defining the RFC

```
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(
    n_estimators=100,
    max_depth=20,
    max_features=None,
    n_jobs=-1,
    verbose=3,
)
```

Now the RFC is able to create a model that is defined above. In section 4.2 is described that CLS will be used instead of the straightforward approach.

5.2 Performance of features

Some features are better than others, and some features are the reason why some training data will overfit the model, because they can contain characteristics that are specific for the training data and that does not necessarily generalize well to the test data. To investigate the importance of the features used in this study, the RFC is trained and evaluated with different subsets of the features that were listed in section 3. First, the RFC is trained using only the visual features (amplitude, sign, width and distance between peaks), they gave already a good performance, see Table 1. However, improving the RFC is always a good thing, so mathematical features (SD, skewness and kurtosis) are added to the feature list of a peak. Adding features will not always give better accuracy scores, because it is possible that they overfit the model, as mentioned above. The results of this analysis are shown in Table 1 as well. The accuracy is used to determine how good the model performs. It is computed by dividing all correct classified pairs by the total number of pairs times 100%. The accuracy can be considered as a good performance indicator for the models because in this study the classes (similar or dissimilar) are in balance in the testset, see Figure 6. Based on the results in Table 1, it can be argued that only the visual features are already very good in classifying the similarity between peaks. Training a model using the mathematical features together with the visual features does not increase the accuracy. Moreover, it decreases the accuracy a little, this could mean that adding extra features the model tends to overfit.

Some features are more used in the decision trees in a RFC than others. To find out which features are the most important for the model, the function `feature_importances_` from the Scikit-Learn API is used. This returns the portion of all the features in the model, such that the summation of all portions is 1. The visual features are separated in two parts to find out the importances: all the features about the distance between peaks and all other visual features (amplitude, sign, width). The importances of the mathematical features are

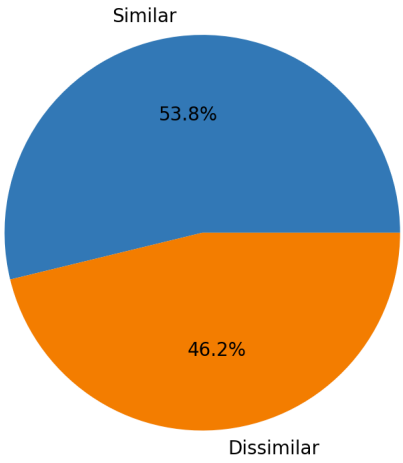

Figure 6: The distribution of the two classes in my testset.

Table 1: Accuracy of the testset compared with different models of subsets of my features.

Which features included in model	**Accuracy score**
Visual features	95.81%
Visual features + mathematical features	95.57%
Only distance features	94.68%

merged. In Figure 7, the importances of those features are illustrated in a barplot. From this Figure it can be argued that the distance features between the peaks are very important for the model compared to the other features. As can be seen, the other visual features are adding a bit to the model and the mathematical features are barely adding anything anymore. So a model is trained using only the distance features and compared to the accuracy of the other models with all visual features and all visual together with the mathematical features. In Table 1 is shown that the distance features alone lead to an accuracy of 94.68%, which is almost as accurate as the other models. From hereon the mathematical features are left out of the models. But all visual features are remained in the models, because they still add a bit to them.

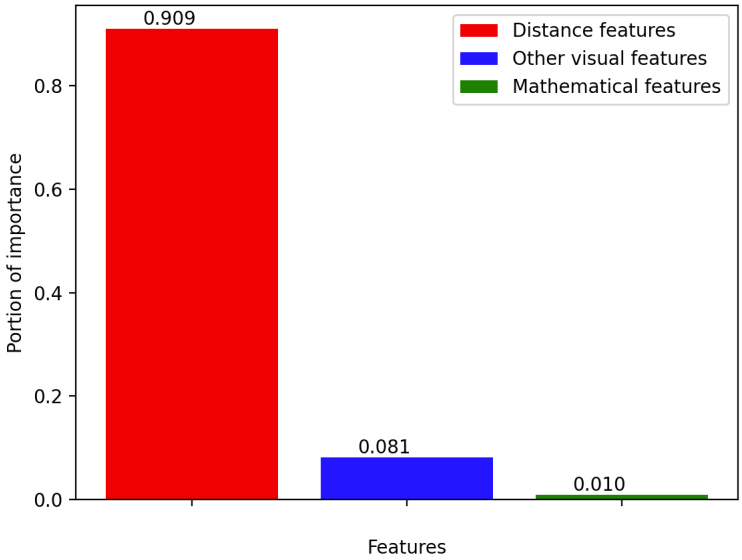

Figure 7: The portion of importance for the features. Grouped into 3 classes: features about distance between peaks, all other visual features from every channel and all mathematical feature from every channel.

5.3 Performance in classifying (dis)similarity

As mentioned before in section 4.2, the RFC is trained to predict the similarity or dissimilarity between pairs of peaks, rather than classify to which foetus a certain peak belongs.
In Table 2, the performance of the RFC for different values of `trainPairs` and `testPairs`, different amount of train files and with the parameter settings as detailed in section 5.1 are reported. From these results, combined with Figures 8 and 9, it can be seen that, even though the RFC has good performance on the test set, the performance is not perfect (i.e. accuracy of 100%). In Figure 8 a cardiotocogram (CTG) is displayed with the expected results (i.e. ground truth). Two graphs that correspond to the individual foetuses are distinguishable such that the FHR of both foetuses can be monitored. Figure 9 displays the CTG of the same recording after assigning all peaks to a label using the RFC. Ideally, the CTGs in Figures 8 and 9 are identical.
Unfortunately, only one wrong classification can have large impact on the ultimate goal of this research: the determination of the heartrate tracings of both twins. If one peak is assigned to the wrong foetus, the subsequent peaks can be classified as similar (or dissimilar) to that peak and assigned to the same (or other) foetus as well, effectively accumulating errors.

Table 2: Accuracy of models using different parameters.

#	**amount of train files**	`trainPairs`	`testPairs`	**Accuracy of model**
1	100	3	3	91.65%
2	100	5	5	95.81%
3	100	5	10	89.03%
4	100	10	10	96.83%

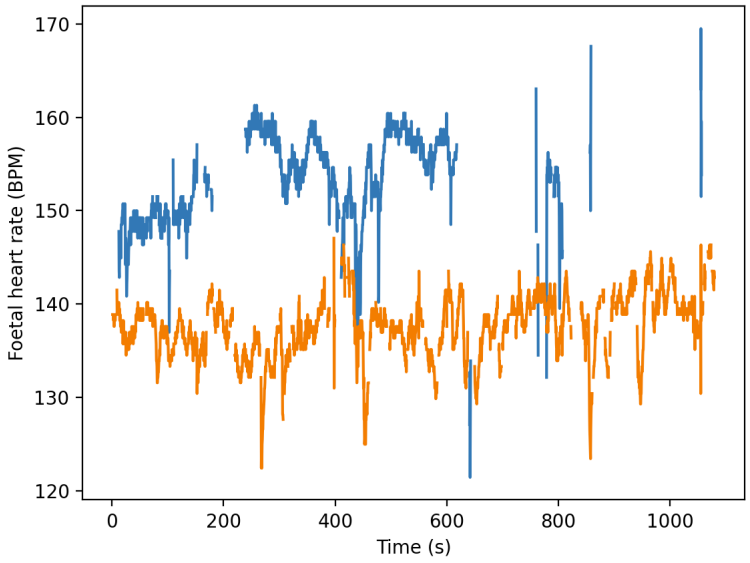

Figure 8: Expected CTG of foetus 1 (blue) and foetus 2 (orange).

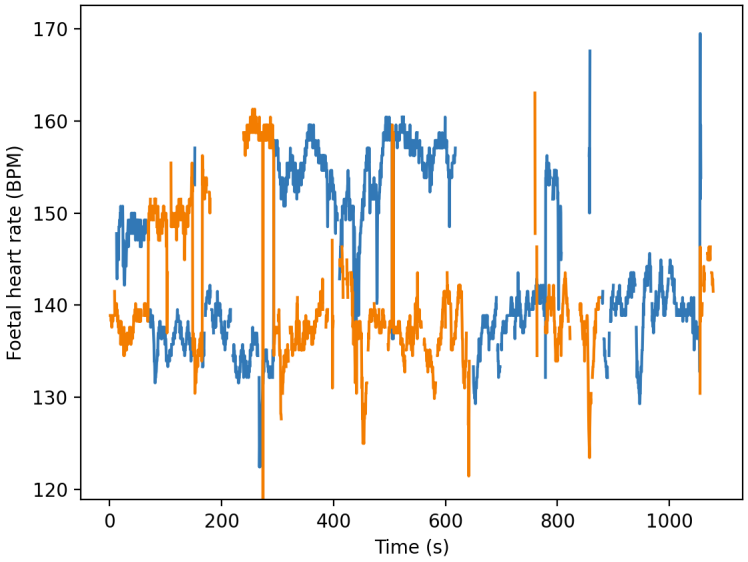

Figure 9: A CTG of both foetuses after assigning all peaks using the RFC using 100 train files, 3 `trainPairs` and 3 `testPairs`.

For the foetal heart rate tracing (i.e. CTG), this will lead to so-called 'swings'. Swings are moments where the classified heart rates switch between foetus 1 and foetus 2 and vice versa. As said, these swings occur because a certain peak in the ECG signal is assigned to the wrong foetus, affecting subsequent peaks as well. In figure 10, an example is shown of an ECG where with red and purple crosses the ground truth labels for the foetuses are shown. In this figure, an assumption is made: a red cross corresponds with foetus 1 and a purple cross corresponds with foetus 2. The orange and green dots are the assigned labels to which foetus the each peak belongs according to the RFC. Before the first swing, the orange predictions agree with the red crosses (so the peaks are assigned to foetus 1) and the green predictions are associated with foetus 2. After the first swing, orange suddenly corresponds to foetus 2 (purple crosses) and green to foetus 1 (red crosses). And it swings back after the second swing. These swings can also be shown in a CTG. In Figure 11 is a zoomed in version of the ECG shown in Figure 9. In this Figure, there are two graphs displayed, a graph around 150 BPM (G1) and a graph around 140 BPM (G2). Those graphs should display the FHR of the two foetuses. G1 starts orange, but after the first swing it becomes blue. G2 starts blue, but after the first swing it turns orange. This means that there is an error in the assigning of the peaks that can be caused by only incident where the peaks are misclassified.

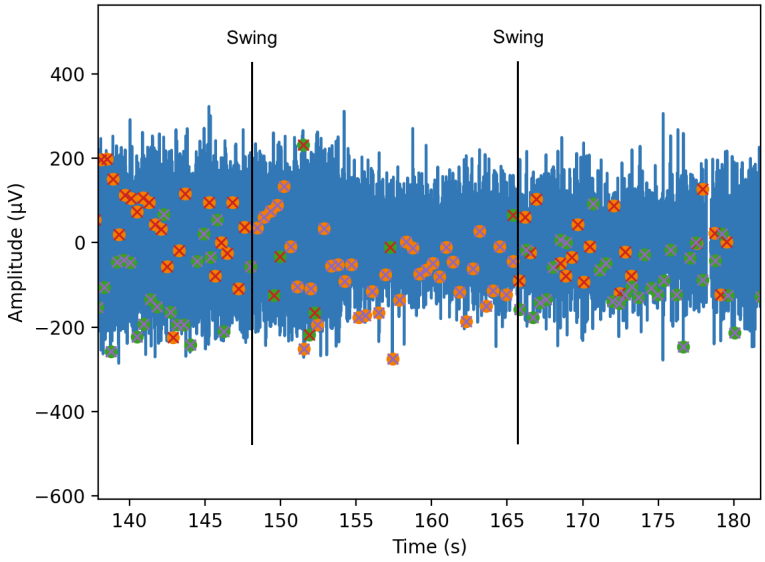

Figure 10: An ECG with the ground truth peaks, red crosses (foetus 1) and purple crosses (foetus 2), and the predicted peaks using the RFC, orange and green dots.

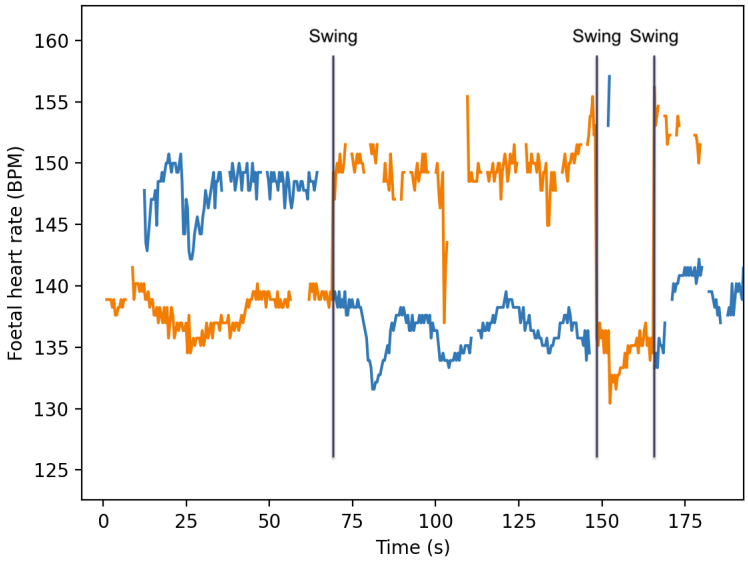

Figure 11: A CTG of two foetuses. A graph around 150 BPM (foetus 1) and a graph around 140 BPM (foetus 2).

Because the CTG is used for clinical decision making, it is of the utmost importance to minimize the incidence of swings. This is possible in two ways: get a accuracy of the RFC of nearly 100%, such that there are almost no mistakes in the similarity pairs. If there are no mistakes, there are no contradictions between different pairs and every peak can be assigned using the label of the previous peak and the pair that was made between a new peak and this previous peak. Another method to reduce the amount of swings is to increase the `testPairs` such that there are more pairs to compare with. If one new peak is compared to multiple peaks that were previously assigned to one of the foetuses, a few classification of (dis)similarity might be wrong, but the majority of classifications would assign the peak to the correct foetus. In other words, by using more pairs of peaks assess to which foetus a specific peak belongs, the majority of pairs should yield a consistent result and the few that are wrong will be overruled by this majority. Figure 12 is a really clear example of the CTG from Figure 9 using more `testPairs`. A disadvantage of increasing the number of pairs is however that these peaks will then be further apart in time. Fetal movements that might have happened in between can have decreased the similarity between the peaks, leading to reduced performance in the classification of (dis)similarity (Table 2, model 3). To some extent, this issues can be resolved by training the RFC with pairs of peaks that are further apart as well. This is shown in Table 2 by comparing model #3 with model #4. Increasing the `trainPairs`, if using more `testPairs` increases the accuracy a lot. The performance of the model is the best if `trainPairs` equal to the `testpairs` is used.

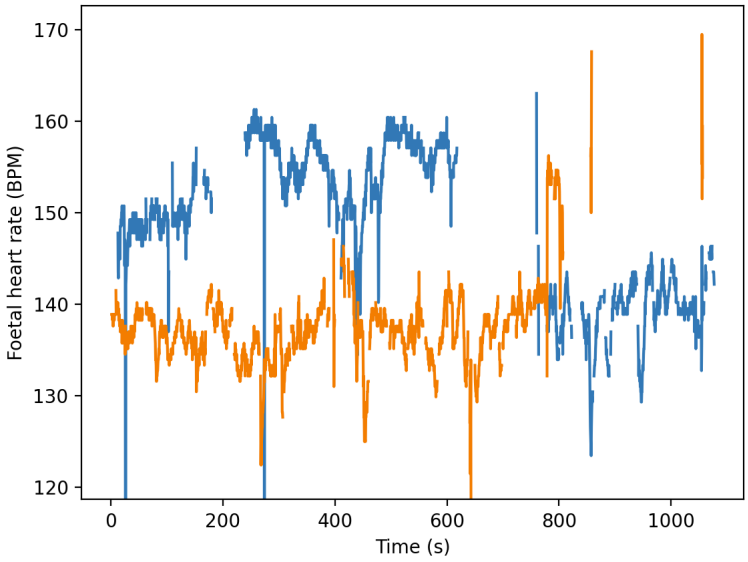

Figure 12: A CTG of both foetuses after assigning all peaks using the RFC using 100 train files, 5 `trainPairs` and 5 `testPairs`.

In Figure 12 a CTG is shown where the FHR of the two foetuses is different. Figure 13, in contrast, shows a CTG where the FHR of the two foetuses are much closer to each other and at some points are even virtually the same. In this Figure the ground truth FHR of both foetuses is shown. Figure 14 shows these FHRs after the detected peaks from the simulated twin recordings have been processed by the RFC and all peaks are assigned to the most probable foetus. As can be seen, there are still a few swings in the data. So, the `testPairs` is increased even higher to possibly get a CTG with less swings, but Figure 15 shows that it does not really make a difference. So forming bigger groups of similarity pairs for testing will decrease the amount of swings, but on a certain size of those groups it does not decrease any further. However, increasing group size leads to increased computation time.

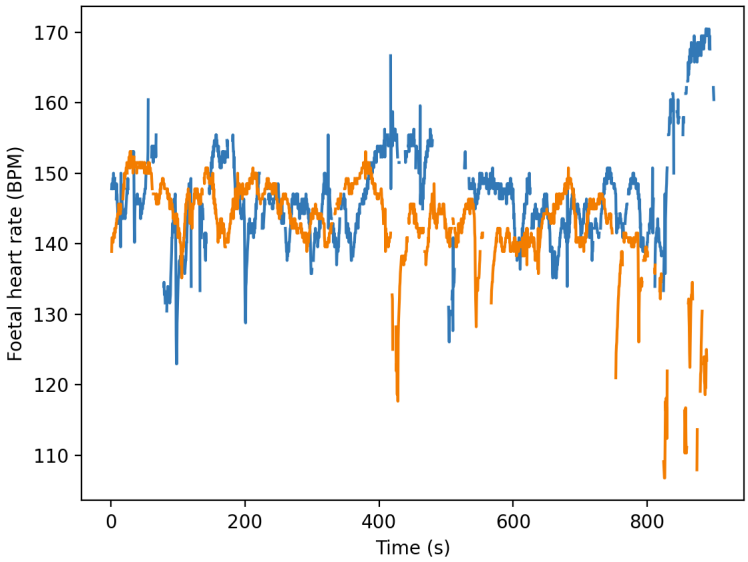

Figure 13: Expected CTG of foetus 1 (blue) and foetus 2 (orange), where the FHRs of both foetuses are close to another.

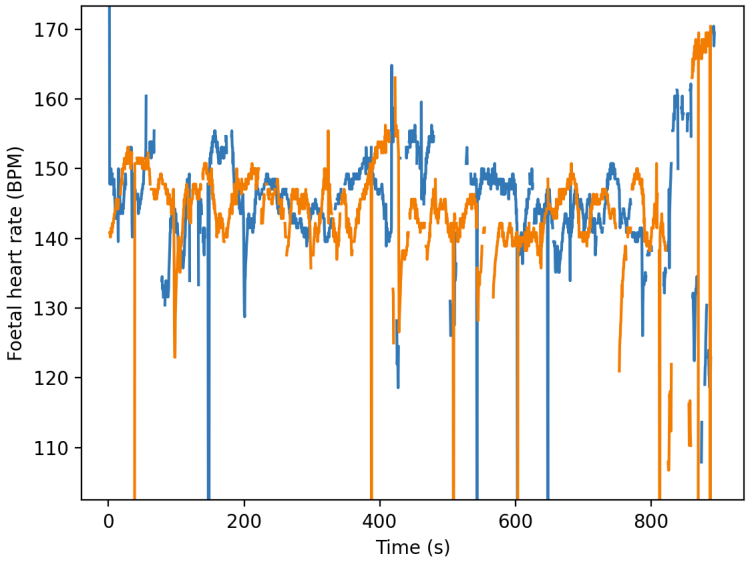

Figure 14: A CTG of both foetuses after assigning all peaks using the RFC using 100 train files, 5 `trainPairs` and 5 `testPairs`, where the FHRs of both foetuses are close to another.

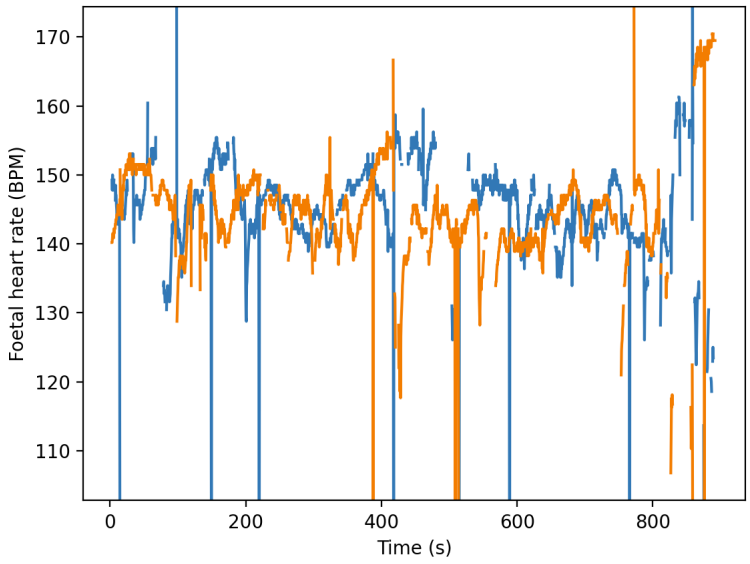

Figure 15: A CTG of both foetuses after assigning all peaks using the RFC using 100 train files, 10 `trainPairs` and 10 `testPairs`, where the FHRs of both foetuses are close to another.

5.4 Real twin pregnancy data file

In the results of the previous subsections, simulated twin measurements were used, that were obtained by combining singleton pregnancy measurements. In section 4.1, an assumption is made that these combined singleton data are representative for real twin pregnancy measurements. In this subsection, the results of testing the model on a real twin pregnancy data file are shown. Based on these results, it can be evaluated whether this assumption is valid.

As mentioned in section 4.1, the real twin measurement has no ground truth, but only a plausible truth. For the sake of simplicity, the plausible truth is assumed to be the actual ground truth. Testing is done by using 5 `testPairs`, because this gave me the best results on the simulated data in the previous subsections. As a consequence, the model that was trained with `trainPairs` equal to 5 is used. This results in a accuracy score of 95.94%. This score is more or less the same as the scores for the combined singleton pregnancies, see Table 2. In Figure 16 CTG as plausible truth is shown. Comparing this with Figure 17 (CTG after assigning all peaks using the RFC), it is clear that there are still swings. But the behavior is very similar to the results for the combined singleton pregnancies data files. So, the trained model is also applicable for a real twin pregnancy measurement.

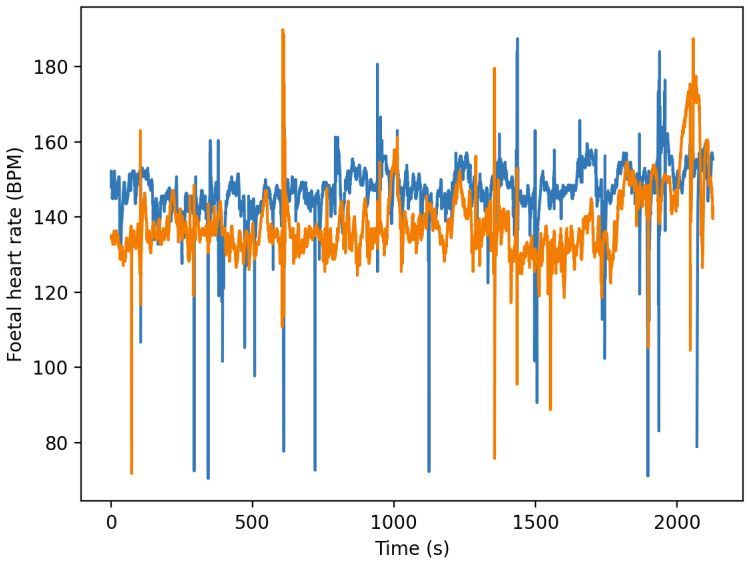

Figure 16: Expected CTG of a real twin pregnancy: foetus 1 (blue) and foetus 2 (orange).

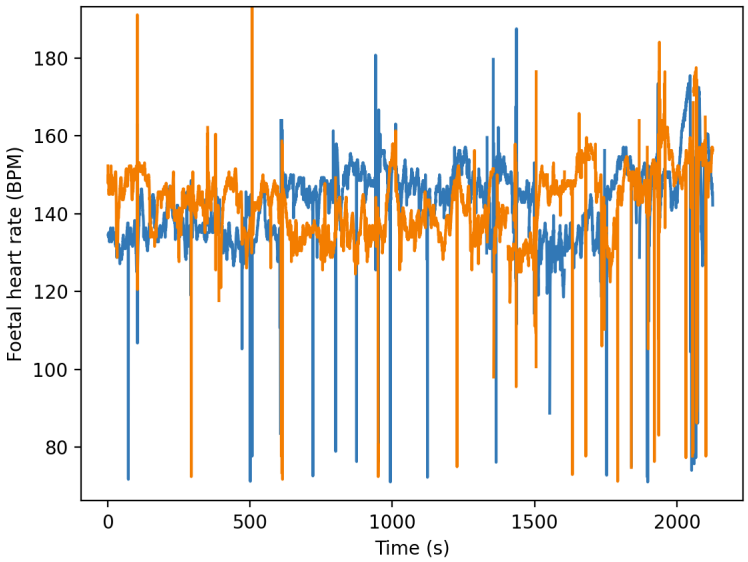

Figure 17: A CTG of a real twin pregnancy of both foetuses after assigning all peaks using the RFC using 100 train files, 5 `trainPairs` and 5 `testPairs`,

6 Related work

In this research, a model is created that tries to separate the heartbeats of two foetuses during twin pregnancy. The model extracts features from the signals that are obtained by NI-fECG measurements. These features are centered around peaks in the signal that are detected with a peak detection algorithm and these peaks are associated with heartbeats from one of the two foetuses. I want to compare my work with work from others, who are trying to solve the same problem. There is little data available about twin pregnancies where the peaks are already separated, making it very hard to check if the separation of the developed model is correct. For this reason, in this research singleton pregnancies were combined and used to simulate a twin pregnancy. In this section a technique is discussed that is used for solving the same problem.

In this paper of Sud, the Fractional Fourier Transform (FrFT) is used to separate the heartbeats of two foetuses in one ECG [8]. FrFT is able to decompose a composed function. An ECG with the heartbeats of two foetuses is an example of such a composed function. The decomposed functions are the heartbeats of the two foetuses separately. FrFT makes use of features of the peaks. The amplitude and the offset in time (distance between a peak and the next peak of the same foetus) are slightly different for each foetus. This method is able to separate the signals when there are slight differences in the amplitude and the time offset. If this is not the case, it becomes more difficult to separate the heartbeats.

There are a lot of similarities between Sud's research and my research. We both use features as input for the separation of the signals. In my research the distance between peaks was very important where Sud used the offset between peaks of the same foetus. And the conclusion of Sud's work was the same as mine, because foetuses with the same FHR are hard to separate. So from this it can be concluded that the amplitude and distance between peaks are good features, but not strong enough to separate foetuses' heartbeats in every twin pregnancy.

7 Discussion

The results who are described in section 5 are a good indicator that it is possible to separate the heartbeats of the foetuses. Hence, the features that are extracted from the peaks in the ECG data seem to contain a sufficient amount of information to distinguish peaks from one foetus from peaks from the other. However, there are still swings in some twin pregnancy measurements, so further improvements are desired. The CTG that can be created from a twin pregnancy measurement is very important for monitoring the foetuses during pregnancy. The CTG is an important instrument for the (early) detection of a deteriorating foetal condition. So, it is of the utmost importance that the heartbeats of the foetuses are separated correctly and there are no swings in a CTG. My created model is able to separate the heartbeats very accurately but not accurate enough to use it for clinical purposes yet. Improving the accuracy of the model will lead to less swings. A possibility to increase this accuracy is optimizing the (hyper)parameters of the RFC. Using more train files can be useful too. The models are trained using only 100 files, because increasing the amount of files increases the computing time of training a RFC a lot. Another possibility to get a higher accuracy is adding features that are able to further optimize the decision trees so that they can distinguish the (dis)similarity of peaks even better. Increasing the accuracy is not the only option to correct the swings. Another method of combining the results from different, partly overlapping pairs, might also improve the assigning of certain peaks to a certain foetus.

8 Conclusion

In this thesis is researched if it is possible to separate the heartbeats of twins during pregnancy using a Random forest classifier. To measure how good the classifier performs the accuracy score is used. The relatively straightforward features, like distance between peaks, amplitude of the peak, the sign of the peak and the width of the peak, are doing really well in training the model. Especially the distance between peaks, which has a relation with the FHR, seems to be highly relevant in the separation of heartbeats. If two peaks have a small distance and if they are from the same foetus, then the FHR of that foetus

is high, and vice versa if a two peaks have a lange distance the FHR would be low. Therefore, the further the FHRs from both foetuses are apart, the easier it gets to correctly classify the different beats. Rapid fluctuations in the FHR are not likely, so the distance between consecutive beats from the same foetus should be more or less similar to the distance between previous beats of that foetus. This is evidenced by the results were the separation of foetuses which have a different FHR performed better than for foetuses that had almost the same FHR.
The model that is trained using only the visual features and 5 `trainPairs` gives an accuracy on the test set (also using 5 `testPairs`) of 95.81%. This is is a very good performance of a model, but the 4.29% peaks that are classified incorrectly still lead to swings in the CTG. This is more likely to happen for foetuses that have a similar FHR, which unfortunately happens relatively frequently. A CTG (the visual representation of the FHR) is used for clinical decision making, so such swings should not occur. Increasing the `testPairs` even further, using information from more peaks to decide if a new peaks belongs to the one foetus or the other, unfortunately does not resolve this issue. So, the trained model is very accurate in classifying if two peaks are from a (dis)similar foetus. But one mistake in a similarity pair can lead to a wrong assignment of a peak, which can affect subsequent peaks as well.
In this thesis is shown that it is possible to separate the FHR in twin pregnancies using a machine learning method like RFC. This method works well for cases where the heartrates of both siblings are relatively far apart. In cases where these heartrates are more similar, the performance decreases and swings in the resulting CTG occur.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[2] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[3] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM Computing Surveys*, 27(3):326–327, September 1995.

[4] J. Hopkin. Fetal heart monitoring, https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/fetal-heart-monitoring.

[5] Emma Long and Emma Ferriman. Twin pregnancy. *Obstetrics, Gynaecology & Reproductive Medicine*, 26(2):38–45, February 2016.

[6] E. Lopriore, J.M. Middeldorp, D. Oepkes, H.H. Kanhai, F.J. Walther, and F.P.H.A. Vandenbussche. Twin anemia–polycythemia sequence in two monochorionic twin pairs without oligo-polyhydramnios sequence. *Placenta*, 28(1):47–51, January 2007.

[7] Lore Noben, Michelle E. M. H. Westerhuis, Judith O. E. H. van Laar, René D. Kok, S. Guid Oei, Chris H. L. Peters, and Rik Vullings. Feasibility of non-invasive foetal electrocardiography in a twin pregnancy. *BMC Pregnancy and Childbirth*, 20(1), April 2020.

[8] S. Sud. Blind separation of twin fetal heartbeats in an electrocardiogram using the fractional fourier transform. *International Journal of Engineering Research and Applications*, 6(4):14–18(5), January 2016.

[9] Rik Vullings and Judith O. E. H. van Laar. Non-invasive fetal electrocardiography for intrapartum cardiotocography. *Frontiers in Pediatrics*, 8, December 2020.

[10] Guy J. J. Warmerdam, Rik Vullings, Lars Schmitt, Judith O. E. H. Van Laar, and Jan W. M. Bergmans. Hierarchical probabilistic framework for fetal r-peak detection, using ECG waveform and heart rate information. *IEEE Transactions on Signal Processing*, 66(16):4388–4397, August 2018.