Classification of Fetal Heart Rate using Feed Forward Neural Networks and Sensitivity Analysis

Malin Hjärtström & Abel de Wit

***Abstract*—Cardiotocography provides a measurement of the fetal heart rate and uterine contractions during childbirth, in order to discover fetal oxygen deprivation. A Feed Forward Neural Network can be used to classify the measured data as normal, suspicious or pathological. Using data from the UCI Machine Learning Repository around 100 models were tested and evaluated, resulting in a final model containing three hidden layers. The final model was evaluated further, using 10 fold cross-validation, which results showed great variability in recall value (normal: 0.9785, suspicious: 0.5695, pathological: 0.6716). The reason for this is believed to be due to the limited amount and skewness of the data. The sensitivity analysis method SHAP was used to receive Shapley values of the predictions. The most important features for the model's prediction were acceleration of the fetal heart rate, percentage of time with abnormal long term variability of the FHR, uterine contractions and percentage of time with abnormal short term variability of the FHR. In the future, it would be encouraged to develop a model using a bigger amount of and less skewed data. An interesting and highly useful approach would be to develop a real time application of the model.**

I. Introduction

Cardiotocography (CTG) is the visual representation of *fetal heart rate* (FHR) and uterine contractions during childbirth. The heart rate of the fetus is an important indicator of fetal status and is a standard procedure during childbirth [1]. A declining heart rate indicates insufficient oxygen supply. If the fetus is not getting enough oxygen, an emergency ceasarian delivery might be necessary [3].

During a more complicated childbirth, the fetal heart rate is measured continuously until the child is born. The heart rate is presented in the shape of a graph, which is manually interpreted [3]. The possibility to use machine learning to interpret the result of the measurement could strengthen the midwives and the obstetricians in their choice of action during complications. It could also relieve the workload for the clinicians and hence giving them more time to care for the patient and the baby.

The use of machine learning in healthcare and medicine is thought to be deeply beneficial due to the sheer amount of data being generated in this area. As for deep learning per se, this kinds of system can accept multiple data types as input, which is of particular relevance for heterogeneous healthcare data [2]. The aim of this project is therefore to implement a *Feed Forward Neural Network* (FNN) to automatically interpret the cardiotocographic measurements and classify the fetal heart rates as normal, suspicious or pathological.

In order for the medical personnel to trust the predictions of the models, the machine must have high medical credibility. One way to reassure this is to measure the importance of each input feature to the output prediction. Therefore the aim is also to determine which features are the most important for the model to make its predictions. This will be done using the sensitivity analysis method *SHAP*.

The background covers the medical aspects of the cardiotocography, as well as the FNN and the sensitivity analysis of choice. Then follows the methodology, where the pre and post processing of the data is presented. The choice of metrics and evaluation method is discussed and argued for. In the result and discussion section, the process of developing and evaluating the models is described, as well as the final model, its results and the results from the sensitivity analysis being presented. After that follows a conclusion and possible approaches to future work are presented.

A. Background

During a pregnancy, the fetus is oxygenated via blood vessels in the placenta. At every contraction, the tense muscles in the abdomen constitutes a risk for inhibiting the blood supply to the placenta and thereby also decrease the fetus's access to oxygen. Normally this does not constitute a problem for the fetus, but in the case of other issues with the blood circulation, the fetus risks being exposed to oxygen deprivation [3]. In order to examine the status of the baby and the mother, bodily signals are measured with cardiotocography.

The word cardiotocography consists of three parts; *cardio* for heart, *toco* aiming to the contractions during childbirth and *graphy* for recording. The CTG measurement is done by placing an ultrasound transducer on the mother's abdomen. The ultrasound transducer measures the fetal heart rate and the changing muscle tensions in the abdominal wall due to contractions. This provides data of, among other things, variability, accelerations and decelerations of the FHR as well as information of the time and strength of contractions [1]. The importance of knowing when contractions occur is due to the inevitable change in FHR when tense muscles limit the blood flow to the baby [3].

The data used in this project is provided by the UCI Machine Learning Repository. The measurements are analysed

June 9, 2021

Malin Hjärtström: malin19@ru.is
Abel de Wit: abel19@ru.is
Yngvi Björnsson, School of Computer Science, Reykjavik University

by *SisPorto*, a measuring system providing a computerized analysis of the CTG [4]. Using deep learning to classify fetal heart rates, both binary and categorically, has been done multiple times, using this data, see for example [6] and [5]. To provide a more precise prediction, this project's implementation aims to categorize the output in three categories (normal, suspicious and pathological) instead of the binary version demonstrated in for example [6].

Feed forward neural networks got their name from the information flowing exclusively through the function without any feedback connections. The information is being evaluated from x, sent through the intermediate computations used to dene f, and forwarded to the output y [8].

A more simple model's predictions can be easy to interpret and therefore preferred, even though its lack of complexity might result in a less accurate model. To increase accuracy, a more complex model is necessary, which then inevitably is more difficult to interpret. Deep learning models are considered more complex and consequently make it harder to understand the reason for the model's prediction. A variety of methods have been proposed to increase the interpretability of model prediction. The sensitivity analysis method *SHAP*, or *SHapley Additive exPlanations* is a method that combines all of these earlier methods into a single equation, see equation (1). This results in assigning each feature an importance value for a particular prediction [9].

$$\phi_i(v) = \frac{1}{|N|!}\sum_R [v(P_i^R \cup \{i\}) - v(P_i^R)] \tag{1}$$

II. Method

The preprocessing of the data consisted soley of normalizing the data. If the data was not normalized, the loss got substantially large. Moreover, previous research has shown that outliers can affect the ability of the machine [7]. Although, it has also been shown that not all outliers are noise. Since it was unclear which outliers were desirable and which were not, all outliers in the data set were kept.

It is a commonly used method to perform *data augmentation* in order to increase the amount of data accessible for training the model. When it comes to medical data, this is unfortunately not an option. Therefore, no augmentation of the data was used and the training and testing was performed on exclusively the data provided from the UCI Machine Learning Repository.

Early runs of the models resulted in very different results, where the predictions for the different classes could vary extensively from run to run. The problem was derived to the dividing of the data into a training and a test set. When starting to develop and evaluate the model, the data was not shuffled. This made the data divided into training and test sets that could be consisting of almost only one class. After realising this, the data was shuffled which improved the learning of the models.

Developing a model for diagnosing illness demands a high level of recall as to not categorize a pathological or suspicious heart rate as normal. That is, identifying a very high level of sick patients out of the total amount of sick patients, see equation (2).

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

TP = True positive
FN = False negative

To evaluate the final model, *10 fold cross-validation* [10] was used. Cross-validation is a statistical method used to estimate the skill of machine learning models. The method was chosen since it generally results in a less biased or less optimistic estimate of the model skill than other methods such as an ordinary train/test split.

III. Result and discussion

When having preprocessed the data and decided upon validation method and choice of metrics, the building of the model began. In the following section, the development and evaluation of the model is described and the final model is further evaluated using 10-fold cross validation. The performance of the final model is discussed and Shapley values are calculated for the predictions. The model and its results are further discussed.

A. Developing the model

Starting with a seemingly basic model, the first tests were run. Although, these runs showed that the model was highly overfitting. In accordance with standard model development procedure, the next step was to start with a very simple base model and slowly increment on it in order to improve the learning while avoiding overfitting [12].

The simple base model consisted of two layers. The input layer was a dense layer with just 10 neurons, and the output layer was a dense layer with 3 neurons matching the three classes that it had to predict. Softmax was used as the activation function in the output layer. This model still seemed to overfit, and therefore a development loop was entered which consisted of:

1) Regularizing the model using kernel regularizers and dropout.
2) When the model stopped overfitting: deepen and/or widen the network in a non-disruptive manner to the results.
3) If the model started to overfit again, the loop would be repeated.

This development process was repeated around a hundred times, varying network depth, width, as well as hyperparameters and callbacks such as early stopping. With each iteration it could be tested whether the model did improve or not on that certain variable and if it did, the variable would be kept. If the model did not improve it would be reversed to the last iteration with the best performance and another variable change would be approached.

The criteria for halting the development for the final model was that no modification seemed to better the metrics that were the target of improvement. The metrics of the five best performing models were saved and can be seen in table I.

Since recall is regarded a very important metric in medical research, one could argue that from the values of table I the

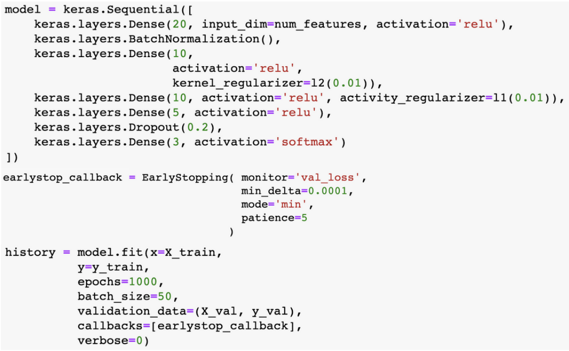
TABLE I
RESULTS OF TEH FIVE BEST PERFORMING MODELS.

	Accuracy	Precision	Recall	F1
1	0.901408	0.8257	0.7637	0.7903
2	**0.929577**	**0.8977**	0.8527	**0.8733**
3	0.924883	0.8643	0.8323	0.8440
4	0.920188	0.8467	**0.8727**	0.8590
5	0.920188	0.8780	0.8427	0.8590

best model would be the 4th (recall 0.8727). All other metrics however excelled in model 2. Moreover, the difference in performance on the recall between model two and four was not substantial enough to choose model 4 over model 2 (recall 0.8527), which had a higher over all score.

B. *The final model*

The final model, model 2, was a Feed Forward Neural Network with three hidden layers (see figure 1). Between the input layer and the first hidden layer batch normalization was used. The model had a dropout of 20% between the third hidden layer and the output layer. The activation function for the input layer and the hidden layers was Rectified Linear Units, while the activation function for the output layer was softmax given that the task is to perform classification of multiple classes.



```
model = keras.Sequential([
    keras.layers.Dense(20, input_dim=num_features, activation='relu'),
    keras.layers.BatchNormalization(),
    keras.layers.Dense(10,
                       activation='relu',
                       kernel_regularizer=l2(0.01)),
    keras.layers.Dense(10, activation='relu', activity_regularizer=l1(0.01)),
    keras.layers.Dense(5, activation='relu'),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(3, activation='softmax')
])
earlystop_callback = EarlyStopping( monitor='val_loss',
                                    min_delta=0.0001,
                                    mode='min',
                                    patience=5
                                  )
history = model.fit(x=X_train,
          y=y_train,
          epochs=1000,
          batch_size=50,
          validation_data=(X_val, y_val),
          callbacks=[earlystop_callback],
          verbose=0)
```

Fig. 1. The structure of the final model. The model had three hidden layers and was regularized with for example batch normalization and dropout. Early stopping was used to avoid overfitting.

For the first hidden layer, a kernel regularizer with parameter 0.01 was used, trying to reduce the weights. The second hidden layer contained an activity regularizer with parameter 0.01, aiming to reduce weights and adjust bias. The activity regulatizer makes the output closer to zero [11].

The input layer had 20 nodes, the first and the second hidden layers had 10 nodes each and the third had 5 nodes. Since predicting the classes normal, suspicious and pathological the output layer had three nodes. In order to avoid overfitting, early stopping was used. The validation error was monitored and when the change from previous loss was less then 0.0001 for six times in a row, the model stopped its training. Number of epochs was set to 1 000, but since using early stopping the model never went through that many iterations. The batch size was set to 50.

C. *Evaluation of the model*

To evaluate the model, 10 fold cross-validation was used. The results can be seen in table II. This process showed great variation between different combinations of the train and test sets. The big discrepancy between the scoring on different classes can be seen in figure 2. Each line represents the recall score of a different fold. As can be seen, the recall is both lower and varies more for the classes suspicious and pathological, than for normal, which are all tightly packed together at a rather high score.

TABLE II
AVERAGE OF THE 10 FOLD CROSS-VALIDATION

	Precision	Recall	F1
Normal	0.9405	0.9785	0.9591
Suspicious	0.701	0.5695	0.6272
Pathological	0.7673	0.6716	0.7172
Average	0.8029	0.7399	0.7678

As can be seen comparing the recall values of table I and II, the recall is lower using the average value of the 10 fold cross-validation compared to simply using a training and a test set. This coincides well with theory, where K-fold cross-validation is perceived as the more realistic metric. Notable is also the before mentioned great discrepancy between the different classes of the cross-validation.

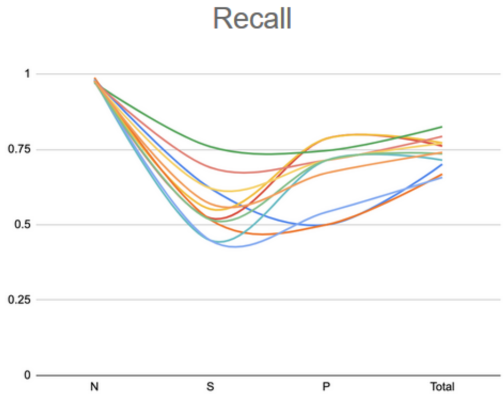


Fig. 2. The different recall graphs represent the 10 different folds in the cross-validation. The recall is both higher and with considerably less variation for the class normal, than for the other two classes.

This variation can have been caused by the skewed data, where the majority of the data belonged to the normal class. That provides the model plenty of data to train for predicting inputs as belonging to the class normal, while considerably less data to train for prediction inputs as belonging to the class suspicious or pathological. It would therefore be of great use to have a both bigger and more evenly distributed data set.

As commented on in II Method, the solution for having too few data points is often data augmentation. But, as previously stated, this is not an option dealing with biomedical data. The greatest possibility of a better model would therefore probably be to extract more data, that is performing measurements during more childbirths. Since CTG is both a standard measurement during childbirth and a non-invasive procedure and therefore not putting the baby or the mother at risk, it would possibly not be too hard to receive large amounts of data this way.

D. Shapley value

A considerable part of developing a model for medical use is how to explain the results to a professional working in the medical field. Therefore, it is an important aspect of the research results to show which elements the network regards as most important in making a decision.

To understand the decisions the model makes and thereby explain the predictions to professionals outside of the computer science field, SHAP was used to perform sensitivity analysis on the final model. SHAP runs the full data set trough the model and observes each feature's impact on the model's decision. The features with the most impact get the highest value, see figure 3. For further explanation of the abbreviations used in fig 3, see table III.

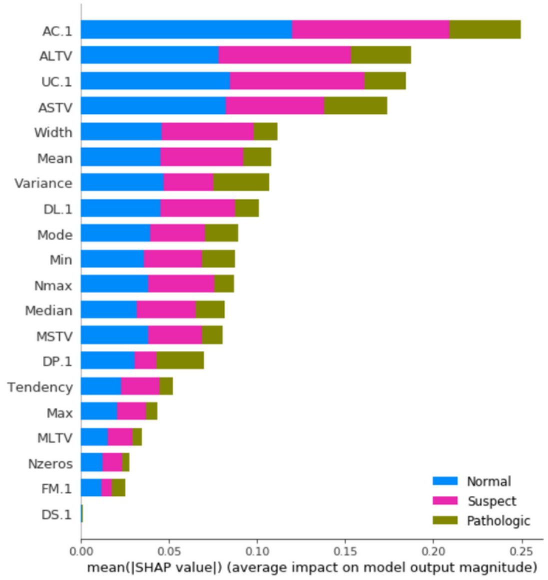


Fig. 3. The result from the sensitivity analysis. As can be seen in the chart bar, the most important input features for predicting the outputs has to do with acceleration and variability of the FHR as well as the uterine contractions of the mother.

The most important features for predicting the output was acceleration of the fetal heart rate, percentage of time with abnormal long term variability of the FHR, uterine contractions and percentage of time with abnormal short term variability of the FHR. When the obstetricians and midwives are provided this kind of information, their decision to trust the model or not can be based on their understanding of the model and not just it's black-box output.

All code and results referenced in this project can be found in the Github repository [13]

TABLE III
DESCRIPTION OF ABBREVIATIONS USED IN FIGURE 3

Abbreviation	Description
AC.1	Accelerations (SisPorto)
ALTV	Percentage of time with abnormal long term variability (SisPorto)
UC.1	Uterine contractions (SisPorto)
ASTV	Percentage of time with abnormal short term variability (SisPorto)

IV. Conclusion

Using machine learning in biomedical applications has great possibilities, given the sheer amount of data measured every day. One downside though is the incapability of augmenting data in cases where the given data set is not enough. The model developed in this project was thoroughly tested on the given data and provided the best results compared to about 99 other models. Despite this, the 10 fold cross-validation showed a lower recall than during the development of the model, as well as a big discrepancy in recall value between the three different classes. The cause of the latter was derived to the skewed data.

The decisions doctors make can have life changing consequences. Therefore, it is uttermost important that the medical personnel can trust the neural networks predictions. This project used the sensitivity method SHAP to derive a Shapley value, which provides a grading for how important each feature is for the model's predictions. For the final model, the predictions were made mostly conditioning on the acceleration and the variability of the fetal heart rate, and also the contractions of the mother's uterus.

In conclusion, the limited amount of data as well as the skewness of it, made the model less accurate than desired. Despite this, the model still had an over all recall of 0.7399. The Shapley value provided extra information about the model's predictions, making the model more useful for doctor's in a real life situation.

A. Future work

In order to make the model more robust, more data is needed. Future work would then contain both the measurement and extraction of more data, as well as the development of a model using this larger amount of data. In addition, the current model is constrained to predicting outcomes of past childbirths. It would be highly interesting and important for the development of using machine learning in medical application, to perform research into real time applications of such model.

References

[1] Science Direct (2019). *Cardiotocography* https://www.sciencedirect.com/topics/nursing-and-health-professions/cardiotocography [2019-11-17]

[2] Nature Medicine (2019-01-07) *A guide to deep learning in healthcare*. url: https://www.nature.com/articles/s41591-018-0316- z [2019-11-15]

[3] Ingemarsson, I & Ingemarsson, E *Fosterövervakning med CTG*. 2012

[4] Omniview-SisPorto (2019). *The Fetal Monitoring Central System with SisPorto Autamted Analysis* [2019-11-20] http://www.omniview.eu/

[5] Miao, J H & Miao, K H (2018). Cardiotocographic Diagnosis of Fetal Health based oon Mutliclass Morphologic Pattern Predictions using Deep Learning Classification. *(IJACSA) International Journal of Advanced Computer Science and Applications*. 9:5

[6] Cömert, Z & Kocamaz, A F (2017). Comparison of Machine Learning Techniques for Fetal Heart Rate Classification. *Acta Physica Polonica A* 132:3, doi: 10.12693/APhysPolA.132.451

[7] Jacob, S G & Ramani G (2012). Evolving Efficient Classification Rules from Cardiotocography Data through Data Mining Methods and Techniques*European Journal of Scientific Research*. 78:3, ISSN 1450-216X

[8] Gupta, T (2917-01-05). Deep Learning: Feedforward Neural Network. *Toward Data Science*. https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7 [2019-11-05]

[9] Lundberg, S M & Lee, S-I (2017-11-25). A Unified Approach to Interpreting Model Predictions. *arXiv:1705.07874v2 [cs.AI]* https://arxiv.org/pdf/1705.07874.pdf

[10] Brownlee, J (2018-05-23). A Gentle Introduction to k-fold Cross-Validation. *Machine Learning Mastery: Making Developers Awesome at Machine Learning* https://machinelearningmastery.com/k-fold-cross-validation/ [2019-11-20]

[11] Stack Exchange. *Cross Validated: Difference between kernel, bias, and activity regulizers in Keras* https://stats.stackexchange.com/questions/383310/difference-between-kernel-bias-and-activity-regulizers-in-keras [2019-11-18]

[12] Goodfellow, I, Bengio, Y & Courville A (2016). *Deep Learning*. http://www.deeplearningbook.org MIT Press: 2016

[13] Github Repository consisting of the notebooks used https://github.com/Abeldewit/CTGDeep