Identifying the health of fetal using Cardiotocography (CTG) data

Cynthia Bachor, *Student ID, 2111599,*,

Abstract—By 2030, it is anticipated that 52 million children under the age of five would have died from preventable causes. While limited access to quality primary health care, unequal access to basic vaccines, pneumonia, and malnutrition are all factors in nearly half of all under-five deaths, a recent UNICEF report on ENDING PREVENTABLE CHILD DEATHS found that the risk of dying is greatest in the first month of life, with 47 percent of deaths among children under five occurring in the first 28 days. Given the aforementioned information, it's critical to keep track of pregnancies at various stages to look out for diseases or medical issues that could put the fetal health at risk. In monitoring babies' heartbeats with a cardiotocograph (CTG), a continuous electronic record of the fetal heart rate is obtained via an ultrasound transducer placed on the mother's abdomen to identify babies who are running low on oxygen and may benefit from an early delivery by caesarean section or instrumental vaginal birth. While some pregnancies may be complicated by a medical condition in the mother (e.g., diabetes or high blood pressure) or a risk of fetal compromise that may result in morbidity or mortality in the fetus or newborn, detecting such complications with CTG can result in early interventions to improve the outcomes of such complications. Cardiotocogram (CTG) results are typically interpreted as belonging to one of three states: physiological, suspicious, or pathological. The goal of this paper is to to develop a multiclass model to automatically classify these states using cardiotocographic data.

Index Terms—Classification of Cardiotocogram (CTG) data, Models: SVC, Logistic regression, Random Forest, Decision Tree

1 Introduction

In 2015, 2.7 million of the 5.9 million deaths among children under the age of five occurred during the neonatal period. Preterm birth complications were the leading cause of death among children under the age of five (1.05 million [95 percent uncertainty range (UR) 0935–1179]) [1]. Some of these complications, if detected early enough, can be managed appropriately.

The foetal heart rate (FHR) is one of the most essential pieces of information about the foetus during the pregnancy cycle. Cardiotocography (CTG) is being used by obstetricians to obtain information such as FHR and other foetal information in order to detect a problematic state early. This information can assist the obstetrician in anticipating future issues, allowing them to take preventative actions before the foetus suffers a lifelong handicap.

A recent review by Grivell et al. reported a significant reduction in perinatal mortality with computerised CTG (relative risk: 0.20, 95 percent confidence interval [CI]: 0.04–0.88) as compared to traditional CTG. [2].In practise, there may be some variation in CTG monitoring success, especially in low-risk pregnancies. If foetal pain is incorrectly assessed, it may lead to ineffective treatments, or if foetal welfare is improperly investigated, it may lead to the exclusion of critical treatments [3]

2 Literature review/Backgroud

Similar efforts have been done in line with this topic, one of which is from Huang [4] who used three different machine learning techniques to predict foetal distress and obtained accuracies of 82.1 percent, 86.36 percent, and 97.78 percent, using discriminant analysis (DA), decision tree (DT), and artificial neural network (ANN) respectively.

Sundar et al. also [5] developed a model that uses an artificial neural network (ANN) to classify CTG data. The recall and F-score were used to evaluate performance. Furthermore, they proposed k-means clustering for CTG classification.

Ocak and Ertunc [6] also proposed a scheme based on neuro-fuzzy inference systems that are adaptive (ANFIS). An ANFIS was trained to predict the normal and pathological states using features extracted from the FHR and UC signals. The method was tested using clinical data from 1,831 CTG recordings. A consensus of three expert obstetricians classified 1,655 of the 1,831 recordings as normal, while the remaining 176 were classified as pathological. The ANFIS-based method was shown to be capable of classifying normal and pathologic states with 97.2 and 96.6 percent accuracy, respectively.

Furthermore, Ocak implemented a classification method based on SVM and Genetic Algorithm (GA). [7] The features extracted from normal and pathological FHR and UC signals were used to construct an SVM based classifier. The GA was then used to find the optimal feature subset that maximizes the classification performance of the SVM based normal and pathological CTG classifier. An extensive clinical CTG data, classified by three expert obstetricians, was used to test the performance of the new scheme. It was demonstrated that the new scheme was able to predict the fetal state as normal or pathological with 99.3% and 100% accuracy, respectively. The results reveal that, the GA can be used to determine the critical features to be used in evaluating fetal well-being and consequently increase

- *Cynthia Bachor is an MSC student with the Department of Computer Science and Electronics Engineering, University of Essex. E-mail: cb21330@essex.ac.uk*
- *Manuscript was prepared between January and April 12, 2022*

the classification performance. When compared to widely used ANN and ANFIS based methods, the proposed scheme performed considerably better.

3 METHODOLOGY

For this project we are given a dataset that contains 2126 records extracted from Cardiotocogram exams, and the aim is to create a multiclass model to classify the CTG features into three fetal health state classes:

- Normal
- Suspect
- Pathological

I have in this project split split the data into 60%, 20%, and 20% for training, validation and test sets respectively, and used four out of the diverse machine learning models in order to select the most appropriate machine learning model for the task, and I'll go over them quickly below:

3.1 Random Forest

Random Forest is the general term for tree-type classifiers.In this model, each tree casts an output vote for each input variable. The classifier's output is the majority of the votes cast by these trees. Pruning can be used to improve classification accuracy, but I did not do any pruning for this experiment. The number of trees used grows in proportion to the number of variables/features. This algorithm is capable of dealing with large amounts of data. [8]

Both algorithms were created using the Python programming language. This project made use of existing software from this package, including RandomForestClassifier.

3.2 Support vector machines (SVMs)

Support vector machines (SVMs) are supervised machine learning algorithms that are both powerful and flexible. They are used for classification and regression. However, they are most commonly used in classification problems. SVMs were first introduced in the 1960s, but they were refined in 1990. When compared to other machine learning algorithms, SVMs have a distinct implementation method. They have recently become extremely popular due to their ability to handle multiple continuous and categorical variables.

An SVM model is essentially a representation of various classes in a multidimensional hyperplane. The hyperplane will be generated iteratively by SVM in order to minimize error. SVM's goal is to divide datasets into classes in order to find the maximum marginal hyperplane (MMH). The primary goal of SVM is to divide datasets into classes in order to find a maximum marginal hyperplane (MMH) by generating hyperplanes iteratively that segregates the classes in best way, and then choosing the hyperplane that separates the classes correctly. [9]

3.3 KNN Algorithm - Finding Nearest Neighbors)

The KNN algorithm is a sort of supervised machine learning method that may be used to solve both classification and regression predicting problems. However, in industry, it is

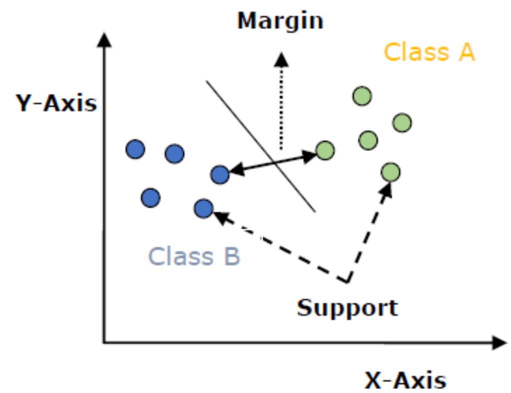

Fig. 1. SVM.

mostly utilised to solve classification and prediction problems. The following two characteristics would be a good way to describe KNN:

- KNN is a lazy learning algorithm because it doesn't have a dedicated training phase and instead uses all of the data for training and classification.
- KNN is also a non-parametric learning algorithm because it makes no assumptions about the underlying data. [9]

3.4 Decision tree(DT)

[9]Decision tree analysis is a predictive modelling approach that can be used in a variety of situations. An algorithmic strategy that can split the dataset in numerous ways based on different conditions can be used to create decision trees. Decision trees can be used for classification as well as regression, and decision nodes where the data is split, and the leaves where we get the result, are the two primary entities of a Decision tree. We have the following two types

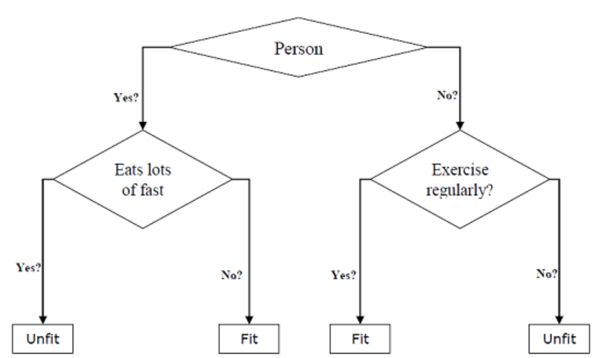

Fig. 2. Decision Tree example.

of decision trees:

- **Classification decision trees:** In this kind of decision trees, the decision variable is categorical. The above decision tree is an example of classification decision tree.

- **Regression decision trees:** In this kind of decision trees, the decision variable is continuous.

Assumptions

Some of the assumptions we make while developing a decision tree are as follows:

- The training set serves as the root node when creating decision trees.
- The categorical values of features are preferred by the decision tree classifier. If you wish to use continuous values in your model, you'll need to discretize them first.
- The records are recursively distributed based on the attribute values.
- To insert attributes at any node position, such as root or internal, a statistical approach will be employed.

3.5 Model Performance Analysis

After using the various classification models explained above to classify the CTG features, the classification results are presented by using precision, recall and F-measure.

Precision is defined as the percentage of instances that belong to a class (TP: True Positive) out of all instances categorised as belonging to that class by the classifier (including TP and FP).

$$Precision = \frac{TP}{TP + FP}, \tag{1}$$

Recall is defined as the percentage of instances classified in one class out of all instances classified in that class. A class's total number of instances includes TP and FN (False Negative).

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

F-measure is the combination of precision and recall

$$F - measure = \frac{2XPrecisionXRecall}{Precision + Recall}, \tag{3}$$

4 Results

For the 4 models used for this classification. we have the classification result for the training, validation and test data sets and also the confusion matrix for both validation and testing set as seen below

4.1 The KNN model

The images below shows classification results derived using the KNN model for each data set

Classification Result for KNN on the Training dataset

	precision	recall	f1-score	support
1.0	0.79	0.98	0.87	987
2.0	0.45	0.09	0.16	180
3.0	0.80	0.04	0.07	108
accuracy			0.78	1275
macro avg	0.68	0.37	0.37	1275
weighted avg	0.74	0.78	0.70	1275

Fig. 3. Classification Result for KNN on the Training dataset.

The images below shows the confusion matrix gotten using the KNN Model

Classification Result for KNN on the Validate dataset

	precision	recall	f1-score	support
1.0	0.80	0.95	0.87	340
2.0	0.00	0.00	0.00	52
3.0	0.00	0.00	0.00	33
accuracy			0.76	425
macro avg	0.27	0.32	0.29	425
weighted avg	0.64	0.76	0.69	425

Fig. 4. Classification Result for KNN on the Validate dataset

Classification Result for KNN on the test dataset

	precision	recall	f1-score	support
1.0	0.77	0.93	0.84	328
2.0	0.13	0.05	0.07	63
3.0	0.00	0.00	0.00	35
accuracy			0.73	426
macro avg	0.30	0.33	0.30	426
weighted avg	0.61	0.73	0.66	426

Fig. 5. Classification Result for KNN on the Test dataset

```
Confusion Matrix on Validate data set using KNN:
[[324   8   8]
 [ 52   0   0]
 [ 31   2   0]]
```

Fig. 6. Confusion Matrix on Validate data set using KNN

```
Confusion Matrix on test data using KNN:
[[306  19   3]
 [ 60   3   0]
 [ 34   1   0]]
```

Fig. 7. Confusion Matrix on Test data set using KNN

4.2 The Random Forest model

The images below shows classification results derived using the Random Forest model for each data set

Classification Report for Random Forest on the Train dataset

	precision	recall	f1-score	support
1.0	1.00	1.00	1.00	987
2.0	1.00	1.00	1.00	180
3.0	1.00	1.00	1.00	108
accuracy			1.00	1275
macro avg	1.00	1.00	1.00	1275
weighted avg	1.00	1.00	1.00	1275

Fig. 8. Classification Result for Random Forest on the Training dataset.

Classification Report for Random Forest on the Validate dataset

	precision	recall	f1-score	support
1.0	0.80	0.97	0.88	340
2.0	0.12	0.02	0.03	52
3.0	0.25	0.03	0.05	33
accuracy			0.78	425
macro avg	0.39	0.34	0.32	425
weighted avg	0.68	0.78	0.71	425

Fig. 9. Classification Result for Random Forest on the Validate dataset

The images below shows the confusion matrix gotten using the Random Forest Model

Classification Report for Random Forest on the Test dataset

	precision	recall	f1-score	support
1.0	0.77	0.98	0.86	328
2.0	0.17	0.02	0.03	63
3.0	0.00	0.00	0.00	35
accuracy			0.75	426
macro avg	0.31	0.33	0.30	426
weighted avg	0.62	0.75	0.67	426

Fig. 10. Classification Result for Random Forest on the Test dataset

```
Confusion Matrix on Validate data set using Random forest:
[[331   6   3]
 [ 51   1   0]
 [ 31   1   1]]
```

Fig. 11. Confusion Matrix on Validate data set using Random Forest

```
[[320   5   3]
 [ 62   1   0]
 [ 35   0   0]]
```

Fig. 12. Confusion Matrix on Test data set using Random Forest

4.3 The Support Vector model

The images below shows classification results derived using the Support Vector model for each data set

	precision	recall	f1-score	support
1.0	0.77	1.00	0.87	987
2.0	0.00	0.00	0.00	180
3.0	0.00	0.00	0.00	108
accuracy			0.77	1275
macro avg	0.26	0.33	0.29	1275
weighted avg	0.60	0.77	0.68	1275

Fig. 13. Classification Result for Support Vector on the Training dataset.

	precision	recall	f1-score	support
1.0	0.80	1.00	0.89	340
2.0	0.00	0.00	0.00	52
3.0	0.00	0.00	0.00	33
accuracy			0.80	425
macro avg	0.27	0.33	0.30	425
weighted avg	0.64	0.80	0.71	425

Fig. 14. Classification Result for Support Vector on the Validate dataset

	precision	recall	f1-score	support
1.0	0.77	1.00	0.87	328
2.0	0.00	0.00	0.00	63
3.0	0.00	0.00	0.00	35
accuracy			0.77	426
macro avg	0.26	0.33	0.29	426
weighted avg	0.59	0.77	0.67	426

Fig. 15. Classification Result for Support Vector on the Test dataset

The images below shows the confusion matrix gotten using the Support Vector Model

```
[[340   0   0]
 [ 52   0   0]
 [ 33   0   0]]
```

Fig. 16. Confusion Matrix on Validate data set using Support Vector

```
[[328   0   0]
 [ 63   0   0]
 [ 35   0   0]]
```

Fig. 17. Confusion Matrix on Test data set using Support Vector

4.4 The Decision Tree model

The images below shows classification results derived using the Decision Tree model for each data set

	precision	recall	f1-score	support
1.0	1.00	1.00	1.00	987
2.0	1.00	1.00	1.00	180
3.0	1.00	1.00	1.00	108
accuracy			1.00	1275
macro avg	1.00	1.00	1.00	1275
weighted avg	1.00	1.00	1.00	1275

Fig. 18. Classification Result for Decision Tree on the Training dataset.

	precision	recall	f1-score	support
1.0	0.80	0.71	0.75	340
2.0	0.11	0.15	0.13	52
3.0	0.10	0.15	0.12	33
accuracy			0.60	425
macro avg	0.34	0.34	0.33	425
weighted avg	0.66	0.60	0.63	425

Fig. 19. Classification Result for Decision Tree on the Validate dataset

	precision	recall	f1-score	support
1.0	0.75	0.70	0.73	328
2.0	0.11	0.13	0.12	63
3.0	0.05	0.06	0.05	35
accuracy			0.57	426
macro avg	0.30	0.30	0.30	426
weighted avg	0.60	0.57	0.58	426

Fig. 20. Classification Result for Decision Tree on the Test dataset

The images below shows the confusion matrix gotten using the Decision Tree Model

```
[[241  60  39]
 [ 39   8   5]
 [ 22   6   5]]
```

Fig. 21. Confusion Matrix on Validate data set using Decision Tree

```
[[231  59  38]
 [ 51   8   4]
 [ 26   7   2]]
```

Fig. 22. Confusion Matrix on Test data set using Decision Tree

5 Discussion and Conclusions

Upon reviewing the results derived using the Random forest, KNN, SVM and Decision Tree classifiers, we can deduce that;

For the KNN model, Confusion matrix for the validation data set is given in figure 6. Most of the Normal class is identified as Normal class whereas 8 cases of suspect (S) class is confused with normal (N) class and pathological (P) class respectively. The average classification accuracy is 70% for the validation data set using this model as seen in figure

4. However, For the test data set, most of the Normal class is identified as Normal class whereas 19 cases of suspect (S) class are confused with normal (N) class and 3 cases of pathological (P) class is also confused with normal class. The average classification accuracy is 67% for the test data set using this model as seen in figure 5

For the Random forest model, the Confusion matrix for the validation data set in this model, Most of the Normal class is identified as Normal class whereas 6 cases of suspect (S) class and 3 cases of pathological (P) class is confused with normal (N) class. The average classification accuracy is 72% for the validation data set using this model. However, For the test data set, most of the Normal class is identified as Normal class whereas 15cases of suspect (S) class are confused with normal (N) class and5 cases of pathological (P) class is also confused with normal class. The average classification accuracy is 68% for the test data set using this model

From the Support vector model, the Confusion matrix for the validation data set in this model, All of the Normal class is identified as Normal class with no misclassifications, and its average classification accuracy is 72% for the validation data set using this model. Same is applicable with the test data set, but the average classification accuracy is 68% for the test data set using this model

While the Decision Tree model shows the Confusion matrix for the validation data set in this model, Most of the Normal class is identified as Normal class whereas 60 cases of suspect (S) class and 39 cases of pathological (P) class is confused with normal (N) class. The average classification accuracy is 61% for the validation data set using this model. However, For the test data set, most of the Normal class is identified as Normal class whereas 59 cases of suspect (S) class are confused with normal (N) class and 38 cases of pathological (P) class is also confused with normal class. The average classification accuracy is 58% for the test data set using this model

Based on the performance evaluation of all four classifiers using three different performance measures, namely Precision, Recall, and F-measure, the pathological and suspicious states of the foetus were distinguished from the normal state. The Support Vector and Random Forest classifiers achieve the best overall classification accuracy of 72 percent on the Validation data set, and 68 percent on the test data set, as seen in this experiment.

REFERENCES

[1] G. M. G. Rosalie M Grivell, corresponding author Zarko Alfirevic and D. Devane, "Antenatal cardiotocography for fetal assessment," https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6510058/.

[2] L. Liu, S. Oza, D. Hogan, Y. Chu, J. Perin, J. Zhu, J. E. Lawn, S. Cousens, C. Mathers, and R. E. Black, "Global, regional, and national causes of under-5 mortality in 2000–15: an updated systematic analysis with implications for the sustainable development goals," *The Lancet*, vol. 388, no. 10063, pp. 3027–3035, 2016.

[3] H. P. Van Geijn, H. W. Jongsma, J. de Haan, and T. K. Eskes, "Analysis of heart rate and beat-to-beat variability: Interval difference index," *American journal of obstetrics and gynecology*, vol. 138, no. 3, pp. 246–252, 1980.

[4] M.-L. Huang and Y.-Y. Hsu, "Fetal distress prediction using discriminant analysis, decision tree, and artificial neural network," 2012.

[5] A. Subasi, B. Kadasa, and E. Kremic, "Classification of the cardiotocogram data for anticipation of fetal risks using bagging ensemble classifier," *Procedia Computer Science*, vol. 168, pp. 34–39, 2020.

[6] H. Ocak and H. M. Ertunc, "Prediction of fetal state from the cardiotocogram recordings using adaptive neuro-fuzzy inference systems," *Neural Computing and Applications*, vol. 23, no. 6, pp. 1583–1589, 2013.

[7] H. Ocak, "A medical decision support system based on support vector machines and the genetic algorithm for the evaluation of fetal well-being," *Journal of medical systems*, vol. 37, no. 2, pp. 1–9, 2013.

[8] M. Arif, "Classification of cardiotocograms using random forest classifier and selection of important features from cardiotocogram signal," *Biomaterials and Biomechanics in Bioengineering*, vol. 2, no. 3, pp. 173–183, 2015.

[9] T. Point, "Machine learning with python)," https://www.tutorialspoint.com/machine$_{l}earning_{w}ith_{p}ython/$.