# Modelling Segmented Cardiotocography Time-Series Signals Using One-Dimensional Convolutional Neural Networks for the Early Detection of Abnormal Birth Outcomes

Paul Fergus, Carl Chalmers, Casimiro Curbelo Montanez, Denis Reilly, Paulo Lisboa, and Beth Pineles

***Abstract*—Gynaecologists and obstetricians visually interpret cardiotocography (CTG) traces using the International Federation of Gynaecology and Obstetrics (FIGO) guidelines to assess the wellbeing of the foetus during antenatal care. This approach has raised concerns among professionals with regards to inter- and intra-variability where clinical diagnosis only has a 30% positive predictive value when classifying pathological outcomes. Machine learning models, trained with FIGO and other user derived features extracted from CTG traces, have been shown to increase positive predictive capacity and minimise variability. This is only possible however when class distributions are equal which is rarely the case in clinical trials where case-control observations are heavily skewed in favour of normal outcomes. Classes can be balanced using either synthetic data derived from resampled case training data or by decreasing the number of control instances. However, this either introduces bias or removes valuable information. Concerns have also been raised regarding machine learning studies and their reliance on manually handcrafted features. While this has led to some interesting results, deriving an optimal set of features is considered to be an art as well as a science and is often an empirical and time consuming process. In this article, we address both of these issues and propose a novel CTG analysis methodology that a) splits CTG time-series signals into n-size windows with equal class distributions, and b) automatically extracts features from time-series windows using a one dimensional convolutional neural network (1DCNN) and multilayer perceptron (MLP) ensemble. Collectively, the proposed approach normally distributes classes and removes the need to handcrafted features from CTG traces. The 1DCNN-MLP models trained with several windowing strategies are evaluated to determine how well they can distinguish between normal and pathological birth outcomes. Our proposed method achieved good results using a window size of 200 with 80% (95% CI: 75%, 85%) for Sensitivity, 79% (95% CI: 73%, 84%) for Specificity and 86% (95% CI: 81%, 91%) for the Area Under the Curve. The 1DCNN approach is also compared with several traditional machine learning models, which all failed to improve on the windowing 1DCNN strategy proposed.**

***Index Terms*—1D convolutional neural networks, classification, deep learning, machine learning, pathological births.**

## I. Introduction

According to the United Nations Children's Fund (UNICEF) 130 million babies are born each year. Approximately 3.5 million will die due to perinatal complications and one million will result in stillbirth. [1]. According to a National Health Service (NHS) Resolution report published in 2017, the number of reported live birth deliveries in England in 2015 was 664,777 of which 1137 resulted in death [2]. The report also states that in the same year there were 2,612 stillbirths. In 2016/2017, maternity errors linked to adverse outcomes cost the NHS £1.7bn with the most expensive claims being for avoidable cerebral palsy [2].

According to MBRRACE-UK there has been a steady fall in the rate of stillbirths, however, neonatal deaths have remained largely static [3]. Cardiotocography (CTG) transducers placed on the mother's abdomen record fetal heart rate and uterine contractions and is the gold standard for assessing the wellbeing of the fetus during antenatal care. The foetal heart rate describes the modulation influence provided by the foetuses central nervous system. When the oxygen supply is compromised, the cardiac function of the fetus is impaired [4].

Clinicians use features defined by the International Federation of Gynaecology and Obstetrics (FIGO) to interpret CTG traces. FIGO features include the real fetal heart rate baseline (RBL), Accelerations, and Decelerations. The RBL is the mean of the signal [5] with peaks and troughs (±10 beats per minute (bpm) from a virtual base line (VBL)) removed from the signal. VBL is the mean of the complete signal.

Accelerations and Decelerations are described as the number of transient increases and decreases (±10 bpm) from the RBL, that last for 10s or more [6]. Accelerations are a sudden increase in the baseline fetal heart rate. They are a good indicator of adequate blood delivery and a reassuring sign for medical practitioners. Decelerations occur due to physiological provocation, such as compromised oxygenation, which often happens when uterine contractions are present. If decelerations fail to recover (i.e. no visible accelerations are present), this is a strong indication that the fetus is compromised due to

Manuscript received March 8, 2020; accepted August 18, 2020. *(Corresponding author: Paul Fergus.)*

Paul Fergus, Carl Chalmers, Casimiro Curbelo Montanez, Denis Reilly, and Paulo Lisboa are with the Faculty of Engineering and Technology at Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, U.K. (e-mail: p.fergus@ljmu.ac.uk; c.chalmers@ljmu.ac.uk; C.A.CurbeloMontanez @ljmu.ac.uk; d.reilly@ljmu.ac.uk; p.j.lisboa@ljmu.ac.uk).

Beth Pineles is with the University of Maryland Medical Center, Baltimore, MD 21201 USA (e-mail: bpineles@gmail.com).

Digital Object Identifier 10.1109/TETCI.2020.3020061

some underlying pathological incidence, such as umbilical cord compression, and is a worrying sign for clinicians [7]. One of the fundamental problems with human CTG analysis is poor interpretation and high inter-intra-observer variability. In many cases, it is not easy to interpret CTG traces and often requires expert knowledge in signal processing. This has therefore made the prediction of neonatal outcomes challenging among healthcare professionals [8]. Computer scientists have investigated this problem using machine learning algorithms to automatically interpret CTG trace patterns. Warrick *et al.* [9] for instance, use FHR and Uterine Contraction (UC) signal pairs to model and estimate the dynamic relationships that often exist between the two [50]. Using their trained models a system was developed to detect pathological outcomes one hour and forty minutes before delivery with a 7.5% false positive rate. Kessler *et al.* [10], on the other hand modelled ST waveforms to provide timely intervention for caesarean and vaginal deliveries. While Menai *et al.* [11] developed a system to classify foetal state using a Naive Bayes (NB) classifier model and four feature selection techniques: Mutual Information, Correlation-based, ReliefF, and Information Gain. The NB model with ReliefF features produced 93.97%, 91.58%, and 95.79% for Accuracy, Sensitivity and Specificity, respectively. Spilka *et al.* [12], used a Random Forest (RF) classifier and latent class analysis (LCA) [13] and produced Sensitivity and Specificity values of 72% and 78% respectively [14]. Generating slightly better results in [15], Spilka *et al.* detected perinatal outcomes using a Support Vector Machine (SVM) with 10-fold cross validation, this time achieving 73.4% for Sensitivity and 76.3% for Specificity.

The fundamental problem with most machine learning studies in CTG trace analysis are twofold. First, machine learning algorithms are sensitive to skewed class distributions which is often the case with data derived from clinical trials where observations are typically normal outcomes [16]. For example, the dataset used in this study, contains 552 singleton pregnancy CTG recordings of which 46 are cases (abnormal birth deliveries) and 506 are controls (normal deliveries). The Synthetic Minority Oversampling Technique (SMOTE) is commonly used to solve this problem [17]. Case observations (minority class) are oversampled using each case record from the training set. This means that new synthetic records are generated along the line segments that join the k minority class nearest neighbours. For a detailed account of our own work in CTG and SMOTE analysis the reader is referred to [18].

Second, expert knowledge is required to extract features from CTG traces and these are application specific. This means handcrafted features are time-consuming and expensive to generate. The rapid progression of signal processing technologies therefore needs a general signal analysis framework that can quickly be deployed to automate this process and accommodate new application requirements.

In this paper, we solve both of these issues using CTG trace segmentation (windowing) to balance class distributions and a one-dimensional convolutional neural network (1DCNN) to automatically learn features from the segmented CTG traces [19]. All windows derived from cases are retained while windows are randomly sampled in controls such that both class distributions are equal. Features are then automatically learned from all case-control window segments. The learnt feature space in the 1DCNN (based on random uniform kernel initialisation) are feed into several fully connected MLPs as input during training. The trained 1DCNN-MLP classifiers are evaluated in several experiments and the results are compared with those obtained from an MLP trained with random weight initialisation, a Support Vector Machine (SVM), a Random Forest (RF), and a Fishers Linear Discriminant Analysis (FLDA) classifier.

The main contributions in this paper are therefore twofold: First, the morphological and nonlinear patterns in CTG traces are modelled using a 1DCNN. The benefits provided by this approach are: 1) it offers a paradigm to learn low and high-level features and interactions that are more flexible than those crafted manually (typically a laborious, subjective, and error prone process), and 2) since all existing state-of-the-art computerised CTG systems use manually extracted features, they generally do not scale well with new data. Therefore, the proposed CTG framework can be quickly deployed to perform CTG modelling on new CTG modalities and applications with little to no human intervention. Second, skewed datasets are balanced using a windowing strategy. The benefits of windowing are: 1) synthetic data to balance classes is not required (algorithms are modelled using real data only), and 2) datasets are not biased due to the addition of data points that are similar to those used by resampling techniques. The performance of the proposed approach is assessed with 552 singleton pregnancy CTG recordings to demonstrate that the proposed framework achieves better performance than existing state-of-the-art methods modelled with synthetic data and handcrafted features.

The remainder of this paper is organised as follows. Section 2 describes the Materials and Methods used in the study. The results are presented in Section 3 and discussed in Section 4 before the paper is concluded and future work presented in Section 5.

## II. Materials and Methods

This section describes the dataset adopted in this study and the steps taken to a) pre-process the data and balance class distributions and b) automatically learn features from n-sized windows with a 1DCNN. The section is concluded with a discussion on the performance metrics implemented to evaluate the machine learning models presented in the results section.

### A. Data Collection and Preprocessing

Cudacek *et al.* carried out a study between April 2010 and August 2012 alongside obstetricians to captured intrapartum CTG Traces from the University Hospital in Brno (UHB) in the Czech Republic with support from the Czech Technical University (CTU) in Prague. The CTU-UHB database contains 552 CTG recordings for singleton pregnancies with a gestational age less than 36 weeks. The STAN S21/31 and Avalon FM 40/50 foetal monitors were connected to the mothers abdomen to acquire the CTG records. The dataset contains ordinary clean obstetrics cases and the duration of stage two labour is less than or equal to 30 minutes. The foetal heart rate signal quality

TABLE I
TRAINING SET CASE/CONTROL SEGMENTS

| Window | Case | Control | Total |
|---|---|---|---|
| 100 | 3898 | 3898 | 7796 |
| 200 | 1947 | 1947 | 3894 |
| 300 | 1299 | 1299 | 2598 |
| 400 | 974 | 974 | 1948 |
| 500 | 779 | 779 | 1558 |

TABLE II
TEST SET CASE/CONTROL SEGMENTS

| Window | Case | Control | Total |
|---|---|---|---|
| 100 | 1241 | 1241 | 2482 |
| 200 | 620 | 620 | 1240 |
| 300 | 413 | 413 | 826 |
| 400 | 310 | 310 | 620 |
| 500 | 248 | 248 | 496 |

is greater than 50% in each 30 minute window and the pH umbilical arterial blood sample for each record is available. The dataset contains 46 caesarean section deliveries and 506 ordinary vaginal deliveries. The 46 cases in this study are classified as caesarean delivery due to pH $\leq$ 7.20 - acidosis, $n = 18$, pH $\geq$ 7.20 and pH $\leq$ 7.25 - foetal deterioration, $n = 4$; and caesarean section without evidence of pathological outcome measures, $n = 24$. Note that the dataset curators do not give a reason why caesarean deliveries were necessary for the 24 subjects were no pathological outcome measures were recorded. Therefore, in this study an assumption is made that the decision to deliver by caesarean was supported by underlying pathological concerns (however, there is no way to validate this). The CTU-UHB database is publicly available from Physionet.

The recordings begin 90 min or less before delivery and contain both the FHR (measured in beats per minute) and uterine contraction (UC) time-series signals. Each signal is sampled at 4 Hz. The FHR is recorded via an ultrasound transducer attached to the abdominal wall and is the only signal used in this study as it provides direct information about the foetal state. Noise and missing values are removed from all recordings using cubic Hermite spline interpolation.

### *B. Cardiotocography Time-Series Windowing*

Each of the 552 signals are split using several windowing strategies with n-size data point coefficients equal to 100, 200, 300, 400 and 500 respectively. First the data set is split into training and test datasets. 405 observations from control records are retained for training and 101 for testing. While 36 case records are retained for training and 10 for testing. In each observation, windowing begins at the first data point in the record with no segments overlapping. For example, in record 2001, using a 300 data point windowing strategy, the first segment starts at 0 and ends and 300, while segment 2 begins at 301 and ends at 600, and so on. All segments are retained from all case observations in the training dataset respectively with an equal number of segments randomly selected from all control records in the training. Note there will be significantly more segments in controls as the dataset is skewed in favour of CTG records for those mother who had normal deliveries. Therefore, we do not need them all only enough such that the number of control segments are equal to the number of case segments - this allows the dataset to be balanced. Table I describes the number of segments in the training data set using different windowing strategies. The resulting datasets are used to train the machine learning models in this study.

The same process is repeated for the test dataset as illustrated in Table II. Again these are used to test all trained models produced.

This class balancing strategy allows the number of case observations to be increased using real data only. Most studies reported in the literature, including our own, have addressed the class skew problem using either over or under sampling [20]. We will discuss our own over sampling strategy later in the paper and compare the results with those obtained using the 1DCNN models produced in this study.

### *C. Feature Learning with One Dimensional Convolutional Neural Network*

In contrast to manually extracted features based on input from domain knowledge experts, features in this study are automatically learnt from the data using a 1DCNN [21]. Windowed CTG traces are input directly to a convolutional layer in the 1DCNN. The convolutional layer detects local features along the time-series signal and maps them to feature maps using learnable kernel filters. Local connectivity and weight sharing is adopted to minimise network parameters and avoid overfitting [22]. Pooling layers are implemented to reduce computational complexity and generate hierarchical data representations [22]. A single convolutional and pooling layer pair along with a fully connected MLP comprising two dense layers and softmax classification output is used to complete the 1DCNN network. The proposed 1DCNN architecture implements one dimensional vectors for kernel filters and feature maps as illustrated in Fig. 2.

The network model is trained by minimizing the cost function using feedforward and backpropagation passes. The feedforward pass constructs a feature map from the previous layer to the next through the current layer until an output is obtained. The input and kernel filters of the previous layer are computed as follows:

$$z_j^l = \sum_{l=1}^{M^{l-1}} 1dconv(x_i^{l-1}, k_{ij}^{l-1}) + b_j^l, \tag{1}$$

where $x_i^{l-1}$ and $z_j^l$ are the input and output of the convolutional layer, respectively, and $k_{ij}^{l-1}$ the weight kernel filter from the $ith$ neuron in layer $l-1$ to the $jth$ neuron in layer $l$, $1dconv$ represents the convolutional operation, and $b_j^l$ describes the bias of the $jth$ neuron in layer $l$. $M^{l-1}$ defines the number of kernel filters in layer $l-1$. A ReLU activation function is used for transforming the summed weights (empirically this activation function produced the best results) and is defined as:

$$x_j^l = ReLU(z_j^l) \tag{2}$$

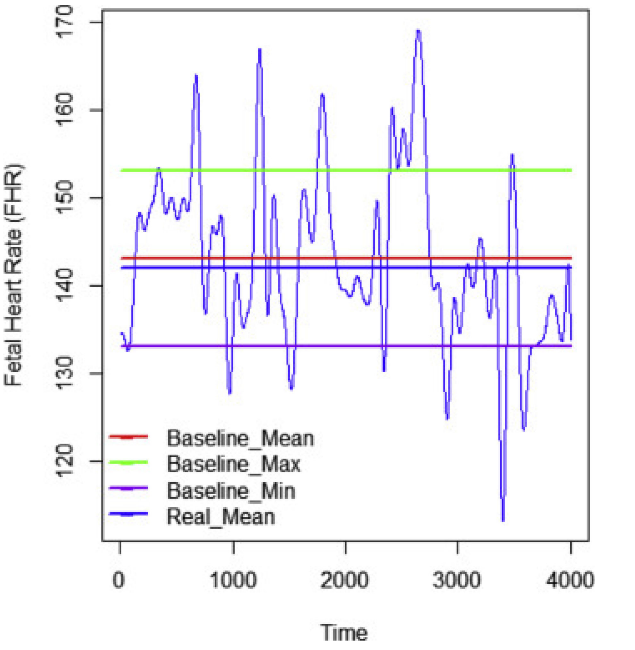


Fig. 1. Using the FHR signal (Beats per Minute) to calculate the real baseline.

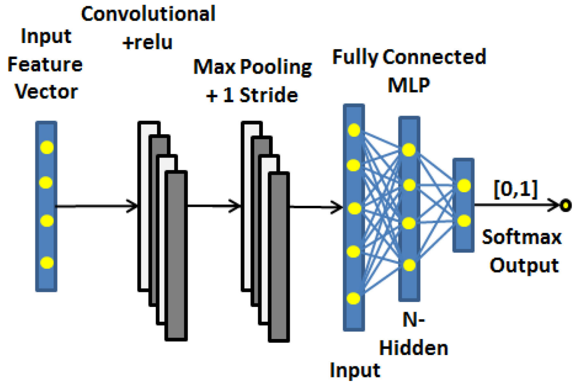


Fig. 2. One dimensional convolutional neural network architecture.

where $x^l_j$ is the intermediate output at current layer $l$ before down sampling occurs. The output from current layer $l$ is defined as:

$$y^l_j = downsampling(x^l_j)x^{l+1}_j) = y^l_j \tag{3}$$

where $downsampling()$ represents a max pooling function that reduces the number of parameters, and $y^l_j$ is the output from layer $l$, as well as the input to the next layer $l+1$. The output from the last pooling layer is flattened and used as the input to a fully connected MLP. Figure 3 shows the overall process.

The error coefficient $E$ is calculated using the predicted output $y$:

$$E = -\sum_n \sum_i (Y_{ni} log(y_{ni})) \tag{4}$$

where $Y_{ni}$ and $y_{ni}$ are the target labels and the predicted outputs, respectively, and $i$ the number of classes in the $nth$ training set. The learning process optimizes the network free parameters and minimises $E$. The derivatives of the free parameters are obtained and the weights and biases are updated using learning rate $(\eta)$. To prompt rapid convergence, we utilise Adam as an optimisation algorithm and apply $He$ weight initialisation. The learning rate $(\eta)$ is set to 0.005 for all experiments. The weights and bias in the convolutional layer and fully connected MLP layers are updated using:

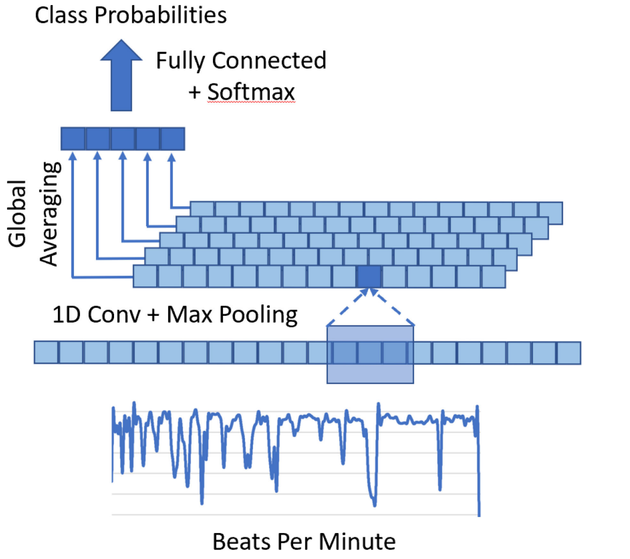


Fig. 3. Convolution and max pooling process.

$$k^l_{ij} = k^l_{ij} - \eta \frac{\partial E}{\partial k^l_{ij}} b^l_j = b^l_j - \eta \frac{\partial E}{\partial b^l_j}, \tag{5}$$

Small learning rates are used to reduce the number of oscillations and allow lower error rates to be generated. Rate annealing and rate decay are implemented to address the local minima problem and control the learning rate change across all layers.

Momentum start, ramp and stable are set to 0.5, $1 * 10^{-6}$ and 0 respectively. Momentum start and ramp control momentum when training starts and the amount of learning for which momentum increases. While momentum stable controls the final momentum value reached after momentum ramp training examples. Complexity is controlled with an optimised weight decay parameter, which ensures that a local optimum is found.

The number of neurons and hidden layers required to minimise $E$, including activation functions and optimisers, were determined empirically. Using 10 input neurons in two hidden layers, and 1 final output node for softmax classification produced the best results.

The network free parameters where obtained using the training and validation sets over 500 epochs and evaluated with a separate test set comprising unseen data.

### D. Performance Measures

Sensitivity and Specificity are implemented to describe the correctly classified normal and pathological birth outcomes. Sensitivity describes the number of true positives (normal deliveries) and Specificity the true negative rate (pathological deliveries).

The area under the curve (AUC) metric calculates the degree of separability between normal and pathological observations.

If $S_0$ is the sum of the ranks of values of inferences for test data in class $C_1$, and similarly for class $C_2$, then the AUC can be defined as:

$$\hat{A} = \frac{1}{n_1 n_2}\left(S_0 - \frac{1}{2}n_1(n_1+1)\right) \tag{6}$$

where $n_1$ and $n_2$ are the number of samples in each class.

Confidence intervals are used to quantify the uncertainty of an estimate based on asymptotic normal approximation. This is described as:

$$CI(p_i) = (\hat{p}_i - k\sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{\#W_i}}, \hat{p}_i + k\sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{\#W_i}}) \tag{7}$$

where $i$ is the $1-\alpha/2)$ - quantile of the standard Gaussian distribution, and the term $\sqrt{\hat{p}_i(1-\hat{p}_i)/\#W_i}$ is an estimate of the standard deviation of the estimated probability $\hat{p}_i$. A 95% confidence level shows the likelihood that the range $x$ to $y$ covers the true Sensitivity, Specificity and AUC values of a particular model.

Logloss is implemented in this study to manage overfitting and measure the accuracy of classifiers - penalties are imposed on classifications that are false. The Logloss value is calculated by assigning a probability to each class rather than stating what the most likely class would be as follows:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(p_i) + (1-y_i)log(1-p_i)] \tag{8}$$

where $N$ is the number of samples, and $y_i$ is a binary indicator for the outcome of instance $i$. If models classify all instances correctly the Logloss value will be zero. For miss-classifications, the value will be progressively larger.

## III. Experiments

In this section three experiments are performed to evaluate CTG classification. First, a trained multi-layer feedforward neural network classification model using raw CTG traces and several windowing strategies is demonstrated. Second, a trained 1DCNN is compared with the trained MLP approach under the same experimental conditions (raw windowed signals). Third, our proposed 1DCNN model is compared with an SVM, RF and a FLDA classifier, again under the same experimental conditions. The performance of each model is measured using Sensitivity, Specificity, AUC, and Logloss (during training). The data set is split randomly into training (80%), validation (10%) and testing (10%). Our method was implemented in Python with Tensorflow GPU 1.13 [23] and Keras 2.2.4 [24]. All experiments were conducted on a computer with an NVidia GTX1060 GPU, a Xeon Processor, and 16GB of RAM.

TABLE III
Baseline MLP Training and Validation Results

| | Training | | Validation | |
|---|---|---|---|---|
| Window | AUC | Logloss | AUC | Logloss |
| W=100 | 0.6395 | 0.6427 | 0.6578 | 0.6464 |
| W=200 | 0.6556 | 0.6349 | 0.7138 | 0.6298 |
| W=300 | 0.5815 | 0.6676 | 0.5638 | 0.6810 |
| W=400 | 0.6546 | 0.6296 | 0.6781 | 0.6202 |
| W=500 | 0.6412 | 0.6389 | 0.4578 | 0.7052 |

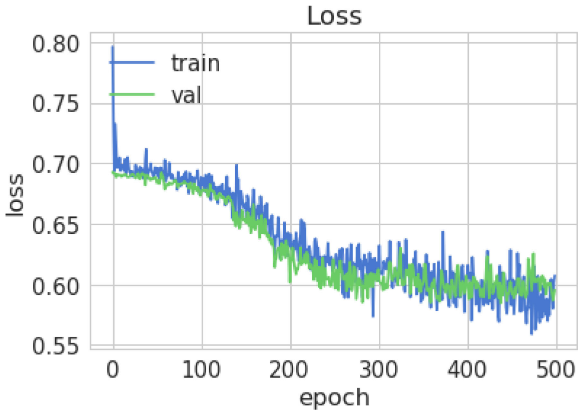


Fig. 4. Baseline MLP training and validation logloss plot for window size 200.

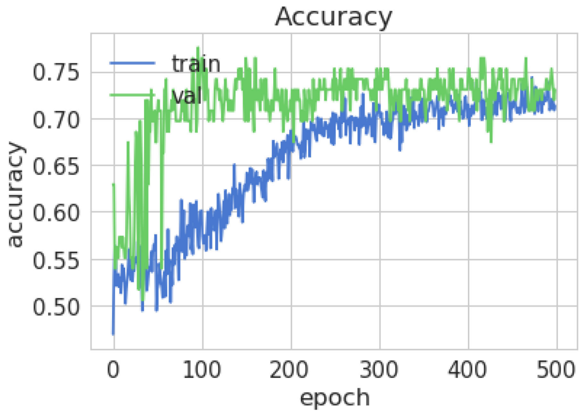


Fig. 5. Baseline MLP training and validation accuracy plot for window size 200.

### A. Multi-Layer Feedforward Neural Network

*1) Classifier Performance:* In the first experiment a single MLP is evaluated using five hidden layers with 10 nodes in each and a final softmax output to classify normal and abnormal birth outcomes. A Relu activation function is used with dropout equal to 0.5. Adam optimisation is implemented with the initial learning rate equal to 0.001. The batch size coefficient is set to 32 and training occurs over 500 epochs. Table III provides the performance metrics for the training and validation sets. Metric values for window sizes 100, 200, 300, 400, and 500 were obtained and averaged over 500 epochs, respectively.

Looking at the validation set the best model was achieved with W = 200. Fig. 3 and 5 show that overfitting is appropriately managed. The AUC plots provide information about early divergence between the training and validation curves. As evidenced in Fig. 3 and 5 learning tends to plateau around 400 epochs.

Table IV provides the performance metrics for the test set. Metric values for window size 100, 200, 300, 400, and 500 were again obtained and averaged over 500 epochs, respectively.

TABLE IV
BASELINE MLP TEST SET RESULTS

| Window | Sens (95% CI) | Spec (95% CI) | AUC (95% CI) |
|---|---|---|---|
| W=100 | 0.31(0.28,0.34) | 0.92(0.90,0.94) | 0.69(0.66,0.73) |
| W=200 | 0.89(0.85,0.93) | 0.51(0.45,0.58) | 0.74(0.68,0.80) |
| W=300 | 0.94(0.90,0.98) | 0.44(0.36,0.52) | 0.71(0.63,0.78) |
| W=400 | 0.42(0.34,0.49) | 0.79(0.73,0.85) | 0.62(0.55,0.69) |
| W=500 | 0.43(0.34,0.51) | 0.84(0.77,0.90) | 0.69(0.61,0.77) |

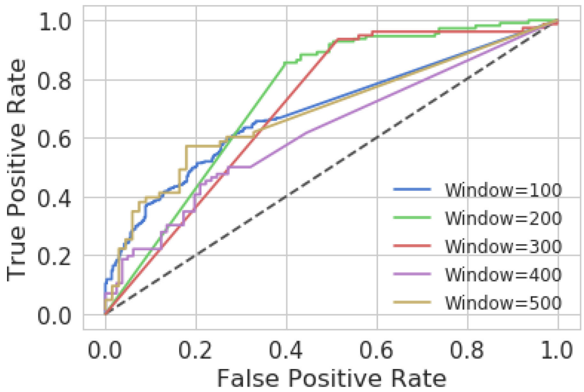


Fig. 6. Baseline MLP test ROC curves for all window sizes.

TABLE V
CONV1D TRAINING AND VALIDATION RESULTS

| | Training | | Validation | |
|---|---|---|---|---|
| Window | AUC | Logloss | AUC | Logloss |
| W=100 | 0.6848 | 0.5812 | 0.6584 | 0.6126 |
| W=200 | 0.7279 | 0.5345 | 0.7284 | 0.5019 |
| W=300 | 0.7737 | 0.4600 | 0.7185 | 0.5762 |
| W=400 | 0.7735 | 0.4750 | 0.6730 | 0.6683 |
| W=500 | 0.7982 | 0.4628 | 0.7333 | 0.6157 |

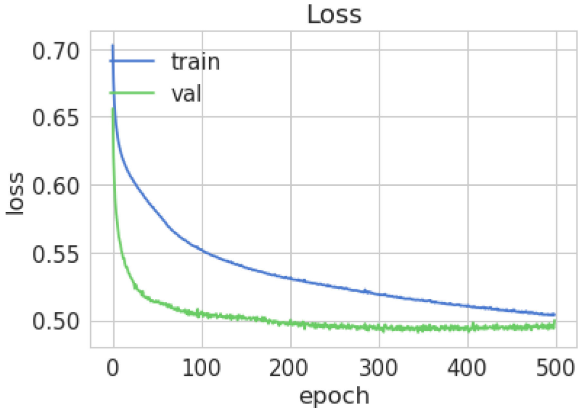


Fig. 7. 1D CNN training and validation logloss plot for window size 200.

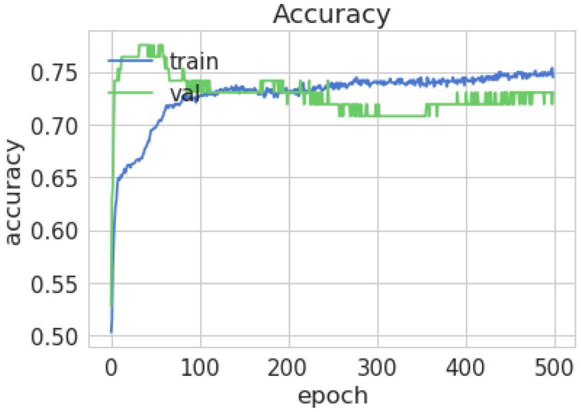


Fig. 8. 1D CNN training and validation accuracy plot for window size 200.

The results are better than those achieved by the validation set, however there is significant imbalance between Sensitivity and Specificity values across all window configurations.

*2) Model Selection:* The ROC curve in Fig. 6 shows that an MLP model with W=200 produced the best results with Sensitivity=89% (CI: 85%,93%), Specificity=51% (CI: 45%,58%) and AUC=74% (CI:68%,80%). As can be seen the Specificity values are low indicating that the model has difficulty classifying pathological birth outcomes.

### *B. One-Dimensional Convolutional Neural Network*

In the second experiment, the same raw CTG signals are used to model a 1DCNN with the network configuration described in Fig. 2 and the network parameter coefficient settings previously discussed.

*1) Classifier Performance:* This time several 1DCNN models are trained using all window size configurations. A single convolutional layer with 20 filters and a kernel size half that of the windowing strategy, i.e. 150 for 300 data points (empirically this produced the best results). A ReLU activation function is implemented in the convolution layer, which is followed by a single max pooling layer and two fully connected dense layers (the first layer contains 10 nodes and the second a single node to classify case and control instances). The nodes in the fully connected layers implement a sigmoid activation function.

All models are compiled with a binary cross entropy loss function and Adam optimizer with the learning rate set to 0.0001, $beta_1$ to 0.9, $beta_2$ to 0.999, epsilon to 0.0, decay to 0.0, and amsgrad to false. Accuracy and Logloss are used as the evaluation metrics with a batch size of 32 and a training strategy that utilises 500 epochs. Ten percent of the training data is retained for model validation.

Table V provides the performance metrics for the training and validation sets. Again, different window size configurations are used and averaged over 500 epochs. The results show that the validation set produced the best results with W=200 based on the highest AUC and lowest Logloss values.

As shown in Fig. 7 the Logloss value converges around 0.50 after 500 epochs with no significant evidence of overfitting. Fig. 8 supports this and shows that both the training and validation plots are closely aligned after 500 epochs.

Table VI this time illustrates that the best performance metrics for the test data was produced with W=200: Sensitivity=80% (CI: 75%,85%), Specificity=79%

TABLE VI
CONV1D TEST SET RESULTS

| Window | Sens<br>(95% CI) | Spec<br>(95% CI) | AUC<br>(95% CI) |
|---|---|---|---|
| W=100 | 0.67(0.63,0.70) | 0.68(0.65,0.72) | 0.76(0.73,0.79) |
| W=200 | 0.80(0.75,0.85) | 0.79(0.73,0.84) | 0.86(0.81,0.91) |
| W=300 | 0.77(0.70,0.84) | 0.73(0.66,0.80) | 0.82(0.76,0.88) |
| W=400 | 0.70(0.63,0.77) | 0.63(0.55,0.70) | 0.72(0.65,0.79) |
| W=500 | 0.68(0.60,0.76) | 0.76(0.70,0.84) | 0.74(0.67,0.82) |

TABLE VII
SVM TEST SET RESULTS

| Window | Sens<br>(95% CI) | Spec<br>(95% CI) | AUC<br>(95% CI) |
|---|---|---|---|
| W=100 | 0.52(0.49,0.56) | 0.52(0.48,0.56) | 0.52(0.48,0.56) |
| W=200 | 0.41(0.34,0.47) | 0.53(0.47,0.60) | 0.47(0.40,0.54) |
| W=300 | 0.51(0.42,0.59) | 0.59(0.51,0.67) | 0.55(0.47,0.63) |
| W=400 | 0.47(0.39,0.54) | 0.55(0.47,0.62) | 0.51(0.43,0.58) |
| W=500 | 0.68(0.60,0.76) | 0.56(0.48,0.65) | 0.62(0.54,0.70) |

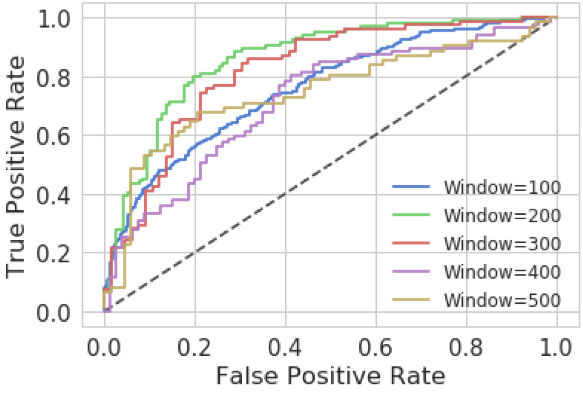


Fig. 9. Baseline CNN test ROC curves for window sizes.

(CI: 73%,84%) and AUC=86% (CI:81%,91%). The metric values are higher than those obtained by the validation set and significantly higher than those produced by the MLP models. The Sensitivity and Specificity values are balanced indicating that the model can distinguish reasonably well between case and control observations with equal accuracy.

*2) Model Selection:* This time Fig. 9 shows that models trained on W=200 and W=300 performed much better than all other window size configurations.

The likely improvement is due to the fact that 1DCNNs are able to extract complex non-linear features (particularly data points with strong relationships) in a way not possible using an MLP alone.

## *C. Comparison With SVM, RF and FLDA*

In the final experiment the 1DCNN results are compared with SVM, RF and FLDA models. The same window configurations are used to model normal and pathological birth CTG traces.

*1) SVM Classifier Performance:* In the first experiment, the same windowed CTG trace configurations are adopted to train the SVM models. Each model is trained by fitting a logistic distribution, using maximum likelihood, to the decision values.

TABLE VIII
RANDOM FOREST TEST SET RESULTS

| Window | Sens<br>(95% CI) | Spec<br>(95% CI) | AUC<br>(95% CI) |
|---|---|---|---|
| W=100 | 0.56(0.52,0.60) | 0.74(0.71,0.76) | 0.65(0.62,0.69) |
| W=200 | 0.65(0.59,0.71) | 0.69(0.63,0.75) | 0.67(0.61,0.73) |
| W=300 | 0.54(0.46,0.63) | 0.75(0.68,0.82) | 0.65(0.57,0.73) |
| W=400 | 0.59(0.51,0.66) | 0.78(0.71,0.84) | 0.68(0.61,0.75) |
| W=500 | 0.55(0.46,0.63) | 0.80(0.73,0.87) | 0.67(0.59,0.75) |

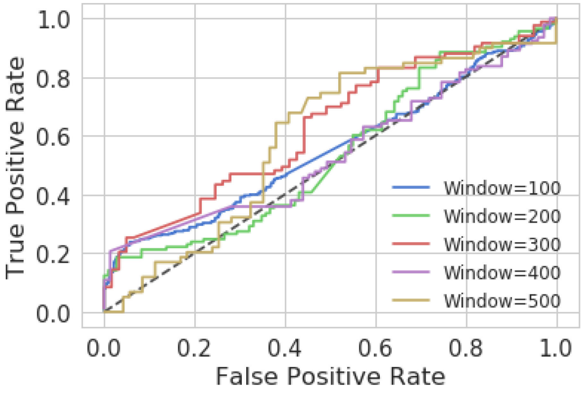


Fig. 10. SVM Test ROC curves for all window sizes.

The same data split strategy is used and a radial kernel function is implemented with gamma and cost parameters 0.3333 and 1 respectively.

This time the performance values for the test set are provided in Table VII. The results shown that all SVM models perform poorly. The best model using W=500 achieved Sensitivity=68% (CI: 60%,76%), Specificity=56% (CI: 48%,65%) and AUC=62% (CI:54%,70%).

*2) Model Selection:* Fig. 9 shows that the ROC curves for all SVM models are located close to the dashed line (random guessing). All models in this experiment fail to produce better classification results than the proposed 1DCNN model.

*3) Random Forest Classifier Performance:* In this second experiment, a Random Forest (RF) model is evaluated using Breiman's RF ensemble learning classifier. Models are trained by decorrelating 500 grown trees generated using bootstrapped training samples. The best model using W=200 achieved Sensitivity=65% (CI: 59%,71%), Specificity=69% (CI: 63%,75%) and AUC=67% (CI:61%,73%). The RF performed better than the SVM models, but failed to improve on the results obtained by the 1DCNN.

*4) Model Selection:* The ROC curves in Fig. 11 for all trained RF models interestingly shows that all models across the five windowing strategies produced similar results - window size appears to have had little or no effect on performance.

*5) FLDA Classifier Performance:* In the final experiment, a FLDA classifier is implemented to linearly combine features to determine the optimal separation between the normal and pathological birth observations. By finding the ratio of between-class to within-class variances, data can then be projected onto a line. This allows classification to be performed in a one-dimensional space. The projection maximizes the distance between the means

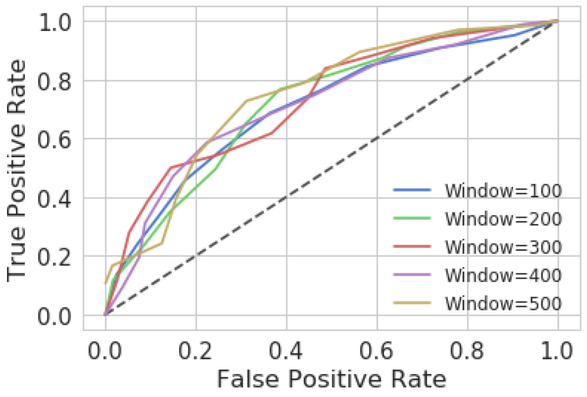


Fig. 11. RF test ROC curves for all window sizes.

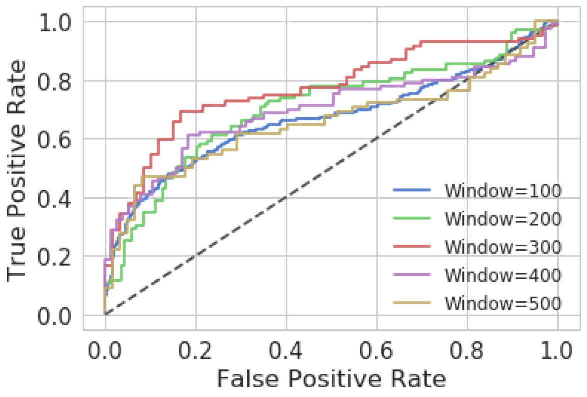


Fig. 12. FLDA test ROC curves for all window sizes.

TABLE IX
FLDA TEST SET RESULTS

| Window | Sens (95% CI) | Spec (95% CI) | AUC (95% CI) |
|---|---|---|---|
| W=100 | 0.63(0.60,0.67) | 0.66(0.62,0.69) | 0.65(0.61,0.68) |
| W=200 | 0.70(0.64,0.76) | 0.66(0.59,0.72) | 0.68(0.62,0.74) |
| W=300 | 0.71(0.64,0.79) | 0.77(0.70,0.84) | 0.74(0.70,0.81) |
| W=400 | 0.63(0.56,0.71) | 0.70(0.63,0.77) | 0.67(0.60,0.74) |
| W=500 | 0.62(0.54,0.70) | 0.61(0.53,0.70) | 0.62(0.53,0.70″) |

TABLE X
SMOTE OVERSAMPLING RESULTS

| Classifier | Sensitivity $(95\%CI)$ | Specificity $(95\%CI)$ | AUC $(95\%CI)$ |
|---|---|---|---|
| FLDA | 0.53(0.46,0.59) | 0.70(0.68,0.72) | 0.67(0.64,0.71) |
| RF | 0.59(0.54,0.65) | 0.57(0.55,0.59) | 0.62(0.60,0.64) |
| SVM | 0.66(0.58,0.74) | 0.41(0.35,0.46) | 0.55(0.52,0.57) |

of the two classes while minimizing the variance within each class.

Table IX provides the performance metrics for the test set. The best performing model was trained using W=300 with Sensitivity=71% (CI: 64%,79%), Specificity=77% (CI: 70%,84%) and AUC=74% (CI:70%,81%). The best performing model performs well given that the FLDA is one of the most simplest and less computationally expensive machine learning models to implement. However, despite these results the FLDA model does not outperform those produced by the 1DCNN.

*6) Model Selection:* Fig. 12 shows the ROC curves for all trained FLDA models. Again, like the RF models, all windowing strategies produced similar results.

## IV. DISCUSSION

Gynaecologists and obstetricians visually interpret CTG traces using FIGO guidelines to assess the wellbeing of the foetus during antenatal care. This approach has raised concerns among healthcare professionals with regards to inter-intra variability were clinicians only positively predict pathological outcomes 30% of the time. Machine learning models trained with features extracted from CTG traces have shown to improve predictive capacity and minimise variability. However, this is only possible when datasets are balanced which is rarely the case in data collected from clinical trials.

Concerns have also been raised on the efficacy of FIGO and handcrafted features and their ability to sufficiently describe normal and pathological CTG traces. Feature engineering requires expert knowledge to extract features and these are often directly related to modality and application. This means that handcrafted features are expensive to produce because manually intensive efforts are required to tune machine learning models for automated CTG analysis.

Both these issues were addressed in this paper by a) splitting CTG time-series signals into n-size windows with equal class distributions using real data only, and b) automatically extracting features from time-series windows using a 1DCNN. The former minimises the amount of bias introduced into the analysis phase and the later automatically extracts features thus removing the need for manual feature engineering. Collectively, we argue this simplifies the data analysis pipeline and provides a robust, rigorous and scalable platform for automated CTG trace modelling and classification.

The findings presented in this paper support the claims made in the study. Splitting CTG traces into n-size windows is a very simple way to balance case-control datasets using real data only. Deep learning extracts important hidden features contained within the data that best describe normal and abnormal CTG traces. More importantly, using a relatively simple 1DCNN it is possible to capture the subtle nonlinear dependencies between the features themselves which may not be easily detected using human visual inspection alone. Consequently, this has the effect of eliminating noise and increasing robustness within the feature extraction process.

Three experiments were presented in this study to evaluate and justify the methodological decisions made. In the first experiment, an MLP, using random weight initialisation, and several window size strategies were evaluated to provide baseline results. An MLP model with a window size=200 produced the best results using the test set (Sensitivity=89% (CI: 85%,93%), Specificity=51% (CI: 45%,58%) and AUC=74% (CI:68%,80%)). When either decreasing or increasing the window size, results dropped with the lowest obtained with window=400 (Sensitivity=42% (CI:

34%,49%), Specificity=79% (CI: 73%,85%) and AUC=62% (CI:55%,69%)). Therefore, changing the window size in this study using the CTG-UHB dataset had no positive impact on overall performance. More importantly, the MLP configuration was unable to equally model and predict between case and control instances as indicated by the Sensitivity and Specificity values.

The second experiment introduced the results for the proposed 1DCNN which automatically extracts features from several CTG window size configurations and uses them to train a fully connected MLP. The results obtained with the test set showed significant improvements in classification accuracies. The best result was achieved using W=200 (Sensitivity=80% (CI: 75%,85%), Specificity=79% (CI: 73%,84%) and AUC=86% (CI:81%,91%)). The worst result was obtained using W=400 (Sensitivity=70% (CI: 63%,77%), Specificity=63% (CI: 55%,70%) and AUC=72% (CI:65%,79%)). The results were much better than those produced using a standard MLP. The Sensitivity value was lower, however, Specificity and AUC increased.

The final experiment modelled an SVM, RF and FLDA classifier, to determine whether these less complex and computationally expensive models could outperform the proposed 1DCNN approach. Under the same evaluation criteria, raw CTG traces where used to train the models with window sizes 100, 200, 300, 400, and 500. The results obtained showed that the best performing classifier was the FLDA using W=300 with (Sensitivity=71% (CI: 64%,79%), Specificity=77% (CI: 70%,84%) and AUC=74% (CI:70%,81%)). The SVM classifier produced the worse results with the best model using W=500 obtaining (Sensitivity=68% (CI: 60%,76%), Specificity=56% (CI: 48%,65%) and AUC=62% (CI:54%,70%)). This was followed by the RF classifier with the best model using W=200 with (Sensitivity=65% (CI: 59%,71%), Specificity=69% (CI: 63%,75%) and AUC=67% (CI:61%,73%)). All of the traditional classifiers performed worse than the 1DCNN, however, the FLDA results were interestingly close to those produced by the 1DCNN with 8% less for Sensitivity and 2% less for Specificity. This result is particularly interesting given that the FLDA is a much simpler model to train compared with CNNs in terms of compute requirements.

In our previous work, SMOTE was utilised as an alternative class balancing strategy [18]. Using the same dataset 80% of observations were allocated for training and the remaining 20% were retained for testing. To balance the dataset the majority classes in the training data were undersampled by 100% and the minority classes oversampled by 600% (resulting in 192 caesarean section records and 224 normal delivery records). An FLDA, SVM and RF classifier were modelled and the average performance of each classifier was evaluated using 30 simulations.The results are shown in Table VIII. The best performing classifier overall was the RF model with (Sensitivity=59% (CI: 54%,65%), Specificity=57% (CI: 55%,59%) and AUC=62% (CI:60%,64%)). However, as can be seen the best windowing and 1DCNN classifier combination posited in this paper outperforms the standard SMOTE approach. For a more detailed discussion on our SMOTE approach the reader is referred to [18].

The results presented in this study are encouraging. While many other studies based on handcrafted features have reported better results, in many cases it is not clear how such results were obtained, i.e. particularly in cases where only accuracy metrics are shown without reference to Sensitivity and Specificity values. In other cases, the good results are likely due to the training and test set minority data points being oversampled rather than the training data points only. Where this is the cases it introduces bias and the trained models are unlikely to generalise well on unseen data. In this sense, we regard the work performed by Spilka *et al.* who use the same dataset, a more realistic fit for evaluation purposes and on these grounds our proposed approach outperforms the results in [12] and [15].

## V. Conclusion

A novel framework to deal with imbalanced clinical datasets, using real data and a windowing strategy is proposed in this paper. Features are automatically extracted using a 1DCNN removing the need for manually handcrafted feature extracted algorithms. Using a dataset containing 552 CTG trace observations (506 controls and 46 cases) a 1DCNN was trained with a W=200 windowing strategy to obtain ((Sensitivity=80% (CI: 75%,85%), Specificity=79% (CI: 73%,84%) and AUC=86% (CI:81%,91%)). Figures 7 and 8 show that there is no significant evidence of overfitting and Fig. 9 shows that our trained models have good predictive capacity.

Nonetheless, there is a great deal of work needed. The results presented in this study are interesting, but the CTG traces used to train the machine learning models did not contain annotations. This means that clinically relevant data and noise are combined in the feature extraction and modelling processes. Therefore, irrelevant data is being modelled alongside key data points representative of abnormal and normal CTG information. Performing signal processing alongside clinicians to only retain parts of the CTG trace directly representative of normal and pathological signals will likely improve the overall predictive capacity of our 1DCNN network.

In future work it may also be interesting to model CTG traces from mothers who have normal vaginal deliveries and implement anomaly detection to identify and triage pregnant mothers with reside outside of normal CTG trace parameters and compare the results with the 1DCNN approach. Making this accessible using web technologies would also be useful to the research community. Therefore, future work will convert Keras models to protobuf models for web hosting using Flask and online inferencing.

Overall, the results highlight the benefits of using CTG trace windowing to balance class distributions and 1DCNNs to automatically extract features from raw CTG traces. This contributes to the instrumentation, measurement and biomedical fields and provides new insights into the use of deep learning algorithms when analysing CTG traces. Work exists in automated CTG trace analysis, however, to the best of our knowledge the study in this paper is the first comprehensive study that windows CTG traces and implements a 1DCNN to automatically extract features for modelling and classification tasks.

## Acknowledgment

The CTU-UHB Intrapartum Cardiotocography Database, developed by Václav Chudáček *et al.* was used in this study. The authors would like to express there appreciation for making the data available to the research community and physionet.org for providing access.

## References

[1] worldometer, "Current World Population," 2018. [Online]. Available: http://www.worldometers.info/world-population/, Accessed: 28 Nov., 2018.

[2] N. Resolution, "Five years of cerebral palsy claims: A thematic review of NHS resolution data," 2017, [Online]. Available: https://resolution.nhs.uk/wp-content/uploads/2017/09/Five-years-of-cere bral-palsy-claims_A-thematic-review-of-NHS-Resolution-data.pdf, 2017, Accessed: Nov. 28, 2018.

[3] "Sands", "MBRRACE-UK: Mothers and babies: Reducing risk through audits and confidential enquiries across the UK," 2017. [Online]. Available: https://www.sands.org.uk/sites/default/files/MBRRACE-UK%20report%20resp onse_08.09.17.pdf, Accessed: Nov. 28, 2018.

[4] P. Olofsson, H. Norén, and A. Carlsson, "New Figo and Swedish intrapartum cardiotocography classification systems incorporated in the fetal ECG st analysis (STAN) interpretation algorithm: Agreements and discrepancies in cardiotocography classification and evaluation of significant ST events," *Acta Obstetricia et Gynecologica Scand.*, vol. 97, no. 2, pp. 219–228, 2018.

[5] A. L. Goldberger *et al.*, "Physiobank, Physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circ.*, vol. 101, no. 23, pp. e215–e220, 2000.

[6] R. Mantel, H. Van Geijn, F. Caron, J. Swartjes, E. Van Woerden, and H. Jongswa, "Computer analysis of antepartum fetal heart rate: 2. detection of accelerations and decelerations," *Int. J. Bio-Med. Comput.*, vol. 25, no. 4, pp. 273–286, 1990.

[7] H. Murray, "Antenatal foetal heart monitoring," *Best Pract. Res. Clin. Obstetrics & Gynaecol.*, vol. 38, pp. 2–11, 2017.

[8] S. Rhöse, A. M. Heinis, F. Vandenbussche, J. van Drongelen, and J. van Dillen, "Inter- and intra-observer agreement of non-reassuring cardiotocography analysis and subsequent clinical management," *Acta Obstetricia et Gynecologica Scand.*, vol. 93, no. 6, pp. 596–602, 2014.

[9] P. A. Warrick, E. F. Hamilton, D. Precup, and R. E. Kearney, "Classification of normal and hypoxic fetuses from systems modeling of intrapartum cardiotocography," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 4, pp. 771–779, 2010.

[10] J. Kessler, D. Moster, and S. Albrechtsen, "Delay in intervention increases neonatal morbidity in births monitored with cardiotocography and st-waveform analysis," *Acta Obstetricia et Gynecologica Scandinavica*, vol. 93, no. 2, pp. 175–181, 2014.

[11] M. E. B. Menai, F. J. Mohder, and F. Al-mutairi, "Influence of feature selection on naïve Bayes classifier for recognizing patterns in cardiotocograms," *J. Med. Bioeng.*, vol. 2, no. 1, pp. 66–70, 2013.

[12] J. Spilka, G. Georgoulas, P. Karvelis, V. Chudacek, C. D. Stylios, and L. Lhotska, "Discriminating normal from "abnormal" pregnancy cases using an automated FHR evaluation method," in *Proc. Hellenic Conf. Artif. Intell.*, 2014, pp. 521–531.

[13] D. Rindskopf and W. Rindskopf, "The value of latent class analysis in medical diagnosis," *Stat. Med.*, vol. 5, no. 1, pp. 21–27, 1986.

[14] V. Chudáček *et al.*, "Open access intrapartum CTG database. BMC Pregnancy Childbirth," Jan. 13, 2014, doi: 10.1186/1471-2393-14-16.

[15] J. Spilka *et al.*, "Using nonlinear features for fetal heart rate classification," *Biomed. Signal Process. Control*, vol. 7, no. 4, pp. 350–357, 2012.

[16] J. Mathew, C. K. Pang, M. Luo, and W. H. Leong, "Classification of imbalanced data by oversampling in kernel space of support vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4065–4076, 2018.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[18] P. Fergus, M. Selvaraj, and C. Chalmers, "Machine learning ensemble modelling to classify caesarean section and vaginal delivery types using cardiotocography traces," *Comput. Biol. Med.*, vol. 93, pp. 7–16, 2018.

[19] J. Kang, Y.-J. Park, J. Lee, S.-H. Wang, and D.-S. Eom, "Novel leakage detection by ensemble CNN-SVM and graph-based localization in water distribution systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4279–4289, 2017.

[20] P. Fergus, A. Hussain, D. Al-Jumeily, D.-S. Huang, and N. Bouguila, "Classification of caesarean section and normal vaginal deliveries using foetal heart rate signals and advanced machine learning algorithms," *BioMed. Eng. OnLine*, vol. 16, no. 1, p. 89, 2017.

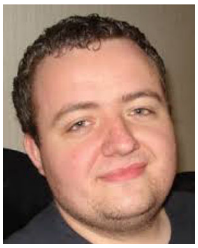
[21] J. Oh, J. Wang, and J. Wiens, "Learning to exploit invariances in clinical time-series data using sequence transformer networks," in *Proc. Machine Learning Healthcare Conf.*, vol. 85, pp. 332–347, 2018.

[22] T. Brosch, Y. Yoo, L. Tang, and R. Tam, "Chapter 3 - Deep learning of brain images and its application to multiple sclerosis," in *Proc. Mach. Learn. Med. Imag.*, G. Wu, D. Shen, and M. R. Sabuncu, Eds. Academic Press, 2016, pp. 69–96.

[23] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016, *arXiv:1603.04467*.

[24] F. Chollet *et al.*, "Keras: The Python deep learning library," *Ascl*, ascl-1806, 2018.

**Paul Fergus** is currently a Reader (Associate Professor) in Machine Learning. He is the Head of the Data Science Research Centre. His main research interests include machine learning for detecting and predicting preterm births. He is also interested in the detection of fetal hypoxia, electroencephalogram seizure classification and bioinformatics (polygenetic obesity, Type II diabetes and multiple sclerosis). He is also currently conducting research with Mersey Care NHS Foundation Trust looking at the use of smart meters to detect activities of daily living in people living alone with Dementia by monitoring the use of home appliances to model habitual behaviors for early intervention practices and safe independent living at home. He has competitively won external grants to support his research from HEFCE, Royal Academy of Engineering, Innovate U.K., Knowledge Transfer Partnership, North West Regional Innovation Fund and Bupa. He has authored or coauthored more than 200 peer-reviewed papers in these areas.



**Carl Chalmers** is currently a Senior Lecturer with the Department of Computer Science, Liverpool John Moores University, Liverpool, U.K. His main research interests include the advanced metering infrastructure, smart technologies, ambient assistive living, machine learning, high performance computing, cloud computing and data visualization. His current research area focuses on remote patient monitoring and ICT-based healthcare. He is currently leading a three-year project on smart energy data and dementia in collaboration with Mersey Care NHS Trust. As part of the project a six month patient trial is underway within the NHS with future trials planned. The current trail involves monitoring and modeling the behavior of dementia patients to facilitate safe independent living. In addition, he is also working in the area of high performance computing and cloud computing to support and improve existing machine learning approaches, while facilitating application integration.

**Casimiro Curbelo Montanez** received the B.Eng. degree in telecommunications from Alfonso X el Sabio University, Madrid, Spain, in 2011, and the M.Sc. degree in wireless and mobile computing and the Ph.D. degree in bioinformatics from Liverpool John Moores University (LJMU), Liverpool, U.K., in 2014 and 2019, respectively. He is currently a Research Assistant with LJMU, under the supervision of Dr. P. Fergus. His research interests include various aspects of data science, machine learning and their use in bioinformatics and biomedical applications.

**Denis Reilly** is currently a Principal Lecturer with the Department of Computer Science, Liverpool John Moores University, Liverpool, U.K. His research career began with the Advanced Robotics Research Centre, University of Salford (1993), where he was involved in research and development into robot control systems, robot programming languages and intelligent user interfaces. He later moved to the Department of Computer Science, The University of Manchester (1996) where he was involved in the development of language processing systems (Interpreters and Translators) for the syntax checking, generation and transformation of CAD is used for electronic PCB layouts. He joined the Department of Computer Science, Liverpool John Moores University in 2000 and undertook research into distributed systems and middleware, with particular emphasis on instrumentation for middleware monitoring. His recent research interests include cloud forensics, data analytics for missing person investigations, intelligent intrusion detection systems and IoT security. During his career, he has worked on a number of EPSRC and EU-funded projects. He has served on the committees of a number of international conferences and acts as a Technical Reviewer to the Association of Computing Machinery (ACM).

**Paulo Lisboa** received the Ph.D. degree in theoretical particle physics (mathematical physics) from Liverpool University, Liverpool, U.K., in 1983. He is currently a Professor and the Head of Department of Applied Mathematics, Liverpool John Moores University, Liverpool, U.K. His research focus is advanced data analysis for decision support. He has applied data science to personalized medicine, public health, sports analytics and digital marketing. In particular, he has an interest in rigorous methods for interpreting complex models with data structures that can be validated by domain experts. He is vice-chair of the Horizon2020 Advisory Group for Societal Challenge 1: Health, Demographic Change and Wellbeing, providing scientific advice to one of the world's largest coordinated research programmes in health. A member of Council for the Institute of Mathematics and its Applications, he is past chair of the Medical Data Analysis Task Force in the Data Mining Technical Committee of the IEEE, chair of the JA Lodge Prize Committee and chair of the Healthcare Technologies Professional Network in the Institution of Engineering and Technology. He is on the Advisory Group of Performance.Lab at Prozone and has editorial and peer-review roles in a number of journals and research funding bodies including EPSRC. He was appointed to the chair of Industrial Mathematics at Liverpool John Moores University in 1996 and Head of Graduate School in 2002.

**Beth Pineles** is currently working toward the M.D./Ph.D. degree with an emphasis on epidemiology to gain more experience in research and public health than offered by the M.D. degree, as an undergraduate and for two years after college, she worked on a variety of subjects, including neonatal pain response and placental microRNAs. She is currently a Fellow in maternal-fetal medicine with the University of Texas Health Science Center, Houston, TX, USA.