Research Article

Vidya Sujit Kurtadikar* and Himangi Milind Pande

1D-CNN: Classification of normal delivery and cesarean section types using cardiotocography time-series signals

Abstract: Cardiotocography (CTG) is considered the gold standard for monitoring fetal heart rate (FHR) during pregnancy and labor to estimate the danger of oxygen deprivation. Visual interpretation of CTG traces is complex and frequently results in high rates of false positives and false negatives, leading to unfavorable and unwanted outcomes such as fetal mortality or needless cesarean surgery. If the data are well-balanced, which is uncommon in medical datasets, machine learning techniques can be helpful in interpretation. This study is designed to determine classification performance under various data balance approaches. We propose a robust methodology for the automated extraction of features that use a deep learning model based on the one-dimensional convolutional neural network (1D-CNN). We used a public database containing 552 intrapartum CTG recordings. Due to the imbalance in the dataset, the experiments were conducted under a variety of conditions such as (i) an unbalanced dataset, (ii) undersampling, (iii) a weighted binary cross-entropy approach, and (iv) oversampling utilizing the synthetic minority oversampling technique (SMOTE). We found an excellent sensitivity (99.80% for the unbalanced dataset, 96.25% for the weighted binary cross-entropy approach, and 99.81% with SMOTE) except for the undersampling situation, in which the sensitivity was 85.71%. Moreover, the 1D-CNN model incorporating SMOTE yielded promising results in 88% specificity, 93.72% quality index (QI), and 95.10% area under the curve. The model exhibited excellent performance in terms of sensitivity in every scenario except for undersampling. The oversampling of training data with SMOTE yielded a decent level of specificity, demonstrating the model's strong predictive capacity. In addition, the SMOTE scenario resulted in fewer training epochs, which is another accomplishment.

Keywords: fetal monitoring, cardiotocography, deep learning, one-dimensional convolutional neural network, classification

MSC 2020: 68T07, 92C55, 92C50, 68T05, 92B20

1 Introduction

According to the World Health Organization's statistical data, 2.4 million infant deaths occurred in the first month of life in 2020. Preterm delivery, issues during childbirth (birth asphyxia or lack of breathing), infections, and congenital disabilities accounted for most neonatal deaths in 2019 [1]. Fetal health is assessed during antepartum and intrapartum to diagnose hypoxia, the early stage of asphyxia. Experts and gynecologists use the Cardiotocography (CTG), which employs Doppler ultrasound and pressure sensors to monitor fetal heart
rate (FHR) and uterine contraction (UC) noninvasively to evaluate fetal well-being. Although various standards for CTG interpretations exist, [2–4], the structure of the guidelines varies. These changes significantly impact observer agreement and dependability [5]. In addition, the intricate structure of CTG traces makes visual interpretation difficult, increasing the false-positive rate and resulting in unnecessary surgical births and cesarean sections [6,7]. Computer scientists have researched many semi-automatic and automatic algorithms to address these limitations. These algorithms extract trustworthy and vital information by leveraging all the untapped latent in the FHR and UC signals to enhance diagnostic efficacy [8–12]. Modern CTG interpretation studies emphasize objective approaches leveraging machine and deep learning. The present study utilized a deep learning method based on the one-dimensional convolutional neural network (1D-CNN) technique, which is well known for automatic feature extraction [13]. In addition, the imbalance in the dataset is addressed using various approaches. Therefore, the contributions of this work are multifold, as given below:

- The current study is likely the first to experiment with and evaluate alternative scenarios based on the imbalance of the dataset.
- Feature extraction is performed automatically using the 1D-CNN deep learning model.
- The total training time is minimized in the SMOTE situation because very few epochs are required.

The remainder of the article is organized as follows: an overview of noteworthy machine and deep learning approaches for CTG interpretation is provided in Section 2; the materials and methods used in this study are described in Section 3; Section 4 explains the performance evaluation criteria for assessing results; Section 5 describes experiments conducted on parameter tuning and data balancing conditions; the results obtained using these experiments are presented in Section 6 and compared to current best practices, followed by a detailed discussion; and the article is concluded in Section 7.

2 Literature review

The notable research based on machine and deep learning methods for analyzing CTG signals is mentioned here. Using the least-squares support vector machine (LS-SVM)-based model and particle swarm optimization [14] classified CTG traces into normal, suspect, and pathological classes with 91.62% accuracy. However, additional helpful evaluation criteria, such as sensitivity and a specificity, must be added. The categorization of FHR signals utilizing generative models (GMs) and Bayesian theory on private datasets yielded a sensitivity of 81.7% and a specificity of 60.0% [15]. The sparse SVM model led to classification results of 73% sensitivity and 75% specificity [16]. An ensemble classifier consisting of SVM, random forest (RF), and Fisher’s linear discriminant analysis (FLDA) classifiers resulted in a sensitivity of 87%, a specificity of 90%, area under the curve (AUC) of 96%, and a mean squared error of 9% [17]. Artificial neural networks performed admirably, with a sensitivity of 99.73% and a specificity of 97.94% as compared to other machine learning techniques [18]. The convolutional neural network (CNN) model with an AUC of 82% surpassed long-term memory in [19]. For a private dataset, the CNN classifier achieved an accuracy of 93.24% [20]. The features were created using multivariate intrinsic mode functions to extract nonlinear and nonstationary relationships from FHR-UC signals [21]. The AdaBoost classifier achieved a sensitivity of 91.8%, a specificity of 95.5%, and an AUC of 98% using those features. A sensitivity of 77.40% and a specificity of 93.86% were obtained using 12 pertinent features and the SVM model after testing with various feature selection methods and machine learning models [22].

Using the multimodal convolutional neural network (MCNN) model and the stacked MCNN model to classify an extensive private database, Petrozziello et al. [23] found that the MCNN model performed better. The recurrence plot transformed an FHR signal from one dimension to two dimensions. The classes were balanced by choosing a small number of matching records from each class, and the image dataset was enriched by modifying several recurrence plot parameters. CNN classifier attained 98.69% accuracy, 99.29% sensitivity, 98.10% specificity, and 98.70% AUC [24]. Several machine learning models used a mixture of traditional and common spatial pattern features as input, and the SVM classifier produced the best results with 74.29% sensitivity, 99.53% specificity, and 94.75% accuracy [25]. The bagging ensemble achieved 99.02%
classification accuracy, a 99% F1-Score, and a 99.99% AUC [26]. The segmentation approach was employed for class balance, and CNN classification yielded an 80% sensitivity, 79% specificity, and 86% AUC [27]. The generative adversarial network (GAN) is used to augment FHR signals to handle the data imbalance problem, yielding 71.08% accuracy, 67.64% sensitivity, and 71.97% specificity [28]. The machine learning model has associated shortcomings in selecting optimal features, which is often complex, time-consuming, and requires skill. Recently, deep learning techniques have proven effective in a variety of real-world applications [29–31] for acquiring relevant and essential information without bias. Based on a 1D-CNN, we suggest a deep learning model, considering the significant benefits of autonomous feature learning and cutting-edge performance for one-dimensional data [13]. The experiment utilized an open-access dataset collected by the Czech Technical University (CTU) in Prague and the University Hospital in Brno (UHB) [32,33]. The dataset is unbalanced regarding cases and controls, including 46 cesarean section births (cases) and 506 vaginal births (controls). We conducted experiments to address this imbalance, considering various scenarios, including:

1. Considering the dataset without any balancing method (unbalanced dataset).
2. Undersampling the records from the control group to match the case group.
3. Using the weighted binary cross-entropy method, which assigns weights to classes such that one misclassification from the case category contributes to as many as 11 misclassifications in the control category (considering 46 cases vs 506 controls in the dataset).
4. Oversampling the minority class using the SMOTE [34], which is based on a combination of oversampling the minority group records and undersampling the majority group records. The oversampling uses k-nearest neighbors to generate synthetic examples in the feature space.

## 3 Materials and methods

This section explains the dataset used for various testing settings, the input normalization method used to normalize the FHR signal, and the network architecture employed in this study.

### 3.1 Dataset

The current study utilized the CTU-UHB intrapartum CTG dataset, freely available at Physionet [32,33]. The dataset comprised 552 intrapartum CTG samples carefully obtained from singleton pregnancies without known congenital disabilities or intrauterine growth restrictions. The significant attributes of the sample and attribute distributions are depicted in Table 1. The dataset consists of 506 recordings delivered vaginally.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Mean (Median)</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mother’s age (years)</td>
<td>29.8</td>
<td>18</td>
<td>46</td>
</tr>
<tr>
<td>Parity</td>
<td>0.43 (0)</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Gravidity</td>
<td>1.43 (1)</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Gestational age (weeks)</td>
<td>40</td>
<td>37</td>
<td>43</td>
</tr>
<tr>
<td>pH</td>
<td>7.23</td>
<td>6.85</td>
<td>7.47</td>
</tr>
<tr>
<td>Base excess</td>
<td>–6.36</td>
<td>–26.8</td>
<td>–0.2</td>
</tr>
<tr>
<td>Base deficit in extracellular fluid (BDecf, mmol/l)</td>
<td>4.60</td>
<td>–3.40</td>
<td></td>
</tr>
<tr>
<td>Apgar score (1 min)</td>
<td>8.26 (8)</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Apgar score (5 min)</td>
<td>9.06 (10)</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Infant weight (kg)</td>
<td>3.408</td>
<td>1.970</td>
<td>4.750</td>
</tr>
</tbody>
</table>

Infant sex (Female/Male): 259 / 293
Type of delivery: Normal (vaginal) = 506, Cesarean section = 46
considered normal delivery cases (control group) in the current study, whereas 46 recordings were delivered by cesarean section (case group). Each record also contains FHR in bpm and UC in mmHg time-series data with a 4 Hz sampling frequency. The recordings are 90 min at maximum and are started at most 90 min prior to delivery. The second phase of labor took up to 30 min. The FHR signal served as the input layer for the 1D-CNN model in the current work under all data balance conditions discussed in Section 5. The present study aims to classify FHR signals in normal delivery and cesarean section records. Each record’s delivery-type parameter is considered the gold standard for the classification. The FHR pre-processing used in the current study is explained in Section 3.2.

3.2 Dataset pre-processing

The movement of pregnant women and fetus, as well as incorrect sensor placement, can introduce noise in the FHR signal. The classification efficiency is highly influenced by input quality; therefore, the pre-processing phase is of utmost importance. The pre-processing used in this article is based on [24], where missing values were linearly interpolated, and missing values that lasted longer than 15 s were removed directly. If the absolute difference of two adjacent values was greater than 25 bpm, then interpolation was carried out between the initial sampling point and the first point of the next stable part. In addition, FHR values larger than 200 bpm or less than 50 bpm were considered extreme points and eliminated. The linear interpolation was used to fill up the corresponding segment.

3.3 Input normalization

The normalization method changes the features to a similar scale, which is essential to improve the classification model’s performance and training stability. This work employed the Z-score normalization method after an empirical analysis of other normalization strategies. The Z-score changes the input such that a new population with a mean of zero and a standard deviation of 1 is obtained. This method subtracted the FHR mean from each FHR value before dividing the difference by the standard deviation. The Z-score normalization approach is depicted by equation (1):

$$FHR'(i) = \frac{FHR(i) - \mu}{\sigma}.$$  \hspace{1cm} (1)

In equation (1), $\mu$ is the mean of the population, $\sigma$ is the standard deviation of the population, $FHR(i)$ is the input FHR value, and $FHR'(i)$ is the normalized FHR. Due to the varying duration of labor among patients, the FHR input signals vary in length. In order to obtain records of similar length, FHR signals are padded with 0 values. Performing Z-score normalization by ignoring the zero entries in the FHR improved the overall performance of classification.

3.4 1D-CNN

Due to the unique property of combining feature mining and labeling procedures within one learning frame, 1D-CNNs have recently gained popularity in a variety of signal-processing applications. Compared to traditional multilayer perceptron (MLP) networks, it can process inputs with high efficacy. CNN has high fault tolerance and robustness and is simple to train and optimize [13].

The core CNN design contains an input layer, a convolution layer, a pooling layer, a fully linked layer, and an output layer. Figure 1 depicts the model employed in this research. A pair of convolution and max pooling layers were employed, succeeded by a layer that flattened the 1D-CNN to facilitate the creation of the fully
connected MLP. The MLP comprised two dense layers consisting of 10 neurons and one neuron each, which were fully connected. Concerning binary classification, sigmoid activation was utilized in the output layer to classify the FHR signal input into normal and abnormal classes. At the convolution layer, rectified linear unit (ReLU) activation was applied. The parameter tuning was carried out for the convolution layer's two essential parameters: the number of filters (denoted by variable \textit{n\_filters}) and the kernel size (denoted by variable \textit{kernel\_s}). The values of those parameters were determined empirically, as discussed in Section 5. The \textit{He} initialization method was employed to initialize the kernel weights. The batch normalization enhanced the efficiency of the deep neural network concerning the training period and overall stability. The layer-wise parameter settings for the proposed 1D-CNN model are shown in Table 2. The detailed parameter tuning for \textit{n\_filters} and \textit{kernel\_s} used in the 1D-CNN layer is discussed in Section 5.2.

The Adam optimization algorithm was utilized with an initial learning rate of 0.005. The problem of local minima was resolved by implementing the exponential learning rate decay approach, which regulates the change in learning rate across all layers. The hyperparameter settings used during the training phase are shown in Table 3.
The equations (2)–(4) summarize the 1D-CNN model’s forward propagation operation. Below is a description of the convolution operation in the convolution layer:

\[ z_j^l = b_j^l + \sum_{i=1}^{N^{l-1}} \text{conv1D}(x_i^{l-1}, w_{ij}^{l-1}), \quad (2) \]

where \( z_j^l \) is the intermediate output from the \( l - 1 \)th convolutional layer, \( b_j^l \) is the bias for the \( j \)th neuron at the \( l \)th layer, \( x_i^{l-1} \) is the input to the convolutional layer, and \( w_{ij}^{l-1} \) is the kernel weight from the \( i \)th neuron in the \( l - 1 \)th layer to the \( j \)th neuron in the \( l \)th layer. \( N^{l-1} \) specifies the total number of filters in the \( l - 1 \)th layer and \( \text{conv1D} \) represents the 1D-CNN operation. The convolutional layer employed the ReLU activation function to convert the intermediate output \( z_j^l \) into its final output \( a_j^l \) using equation (3).

\[ a_j^l = \text{ReLU}(z_j^l). \quad (3) \]

The pooling layer’s objective is to minimize the size of feature maps. During max-pooling, the largest value was selected from the area of the feature space. Equation (4) represents the output \( y_j^l \) due to the max-pooling layer.

\[ y_j^l = \text{maxpooling}(a_j^l) \quad (4) \]

The fully linked MLP layer took its input from the flattened output of the pooling layer. The error rate for backpropagation is computed using binary cross-entropy, as shown in equation (5).

\[ E = \frac{1}{m} \sum_{i=1}^{m} (y_i \log(y_{\text{pred},i}) + (1 - y_i) \log(1 - y_{\text{pred},i})), \quad (5) \]

where \( m \) represents the total number of training examples, \( y_i \) is the actual label, and \( y_{\text{pred},i} \) is the predicted output of the \( i \)th training example. The weights and biases were changed during backpropagation using equations (6) and (7). In this process, weight and bias derivatives were applied.

\[ w_{ij}^l = w_{ij}^l - \alpha \frac{\delta E}{\delta w_{ij}^l} \quad (6) \]

\[ b_j^l = b_j^l - \alpha \frac{\delta E}{\delta b_j^l} \quad (7) \]

where \( \alpha \) is the learning rate in the proposed network. The backpropagation seeks to optimize the learning process by minimizing the value of the binary cross-entropy error specified in equation (5).

### 4 Performance evaluation

The performance of the 1D-CNN method was assessed using the parameters specified in equations (8)–(13). True positive, false positive, true negative, and false negative values are denoted as TP, FP, TN, and FN, respectively.
5 Experiments

This section describes the current study’s experimental setup, various experiments conducted for parameter tuning for the proposed 1D-CNN model, and methods for addressing data imbalance.

5.1 Experimental setup

The proposed method was employed using Python 3.9.7, TensorFlow 2.8.0, and Keras 2.8.0. All experiments were conducted with Intel R i5, RAM: 8 GB, x64-based processor.

5.2 Experiment one: parameter tuning of 1D-CNN layer

The proposed deep learning model is a layered architecture in which the core layer is 1D-CNN that decides classification performance. The most critical parameters that affect the convolution operation in the 1D-CNN model are the kernel_s and n_filters required for the optimal performance of the model. The following experiments were conducted to decide the optimal settings for these two parameters. The base condition of the CTU-UHB dataset, without any class balancing method, was used for performance tuning experiments. The layer-wise setting and hyperparameter settings were as per Tables 2 and 3. Considering initial experimentation, the model was trained using 50 epochs and 64 training examples per batch.

5.2.1 Experiment for n_filters

The dataset consists of 506 normal delivery records and 46 cesarean section records, as explained in Section 3.1. This unbalanced condition is considered for deciding on two crucial parameters, kernel_s and n_filters. During the initial preliminary experimentation, the performance was observed to be highly dependent on the n_filters. Therefore, initial experimentation for parameter tuning is conducted for the appropriate value of n_filters. The layer-wise parameters were in accordance with Table 2, except for n_filters. The training parameters were as mentioned in Table 3. The initial setting for kernel_s was kept as kernel_s = 15, and the n_filters = [1,2,3,4,5,6] were considered for tuning. The values are selected as per initial preliminary experiments. The model was trained using the cross-fold technique with 10 folds, and the result was averaged across 10 folds.
Thirty-one simulations were carried out for each choice of $n_{\text{filters}}$, and statistical analysis of Acc, Se, Sp, QI, and AUC parameters was done by plotting the results using boxplots, as illustrated in Figure 2. The topmost plots denote the Acc and Pre plots; the middle row represents the Se and Sp boxplots, whereas the QI and AUC

\begin{figure}
\centering
\begin{tabular}{cc}
\includegraphics[width=0.45\textwidth]{accuracy_boxplot} & \includegraphics[width=0.45\textwidth]{precision_boxplot} \\
\includegraphics[width=0.45\textwidth]{sensitivity_boxplot} & \includegraphics[width=0.45\textwidth]{specificity_boxplot} \\
\includegraphics[width=0.45\textwidth]{qiu_boxplot} & \includegraphics[width=0.45\textwidth]{auc_boxplot}
\end{tabular}
\caption{The boxplots for averaged performance parameters using different values of $n_{\text{filter}} = [1, 2, 3, 4, 5, 6]$ across 10 folds. From top-left to top-right: Acc, Pre; from middle-left to middle-right: Se, Sp; from bottom-left to bottom-right: QI, and AUC. (a) Acc boxplot, (b) Pre boxplot, (c) Se boxplot, (d) Sp boxplot, (e) QI boxplot, and (f) AUC boxplot.}
\end{figure}
boxplots are represented in the bottommost row. The Acc and Pre boxplots suggested that \( n_{\text{filters}} = 1 \) is the optimal choice, whereas the Se parameter suggested that \( n_{\text{filters}} = 5 \) is an excellent choice.

The Se parameter is helpful to avoid unnecessary cesarean sections, while the Sp parameter is significant to avoid fetal compromise. The Sp parameter boxplot suggested that \( n_{\text{filters}} = 1 \) will be the best option. Boxplots of QI and AUC parameters also confirmed the same value. The \( n_{\text{filters}} = 1 \) value also yielded promising results in the acceptable range regarding the Se parameter and, therefore, was selected for further tuning the \( \text{kernel}_s \) parameter, as explained in Section 5.2.2.

5.2.2 Experiment for \( \text{kernel}_s \)

The following experiment was conducted to determine the value of the \( \text{kernel}_s \) parameter used in the convolutional layer. The \( n_{\text{filters}} = 1 \) was considered for choosing the appropriate value of \( \text{kernel}_s \). The various values of \( \text{kernel}_s = [5, 10, 15, 20, 25, 30] \) were considered for experimentation. The cross-fold technique was used to train the model with 10 folds, averaging the results. A total of 31 simulations were carried out for each choice of \( \text{kernel}_s \). The resulting boxplots of Acc, Pre, Se, Sp, QI, and AUC parameters are shown in Figure 3. The model’s performance in terms of the Se parameter was excellent for \( \text{kernel}_s = 10 \); similar performance was observed for \( \text{kernel}_s = [15, 20, 25] \). Although the Sp parameter’s value was low due to very few cesarean section records in the unbalanced dataset, it was better for \( \text{kernel}_s = 10 \) with the highest median value indicated by the orange line in the boxplot. The Sp parameter value was good for \( \text{kernel}_s = 5 \), but there was a high variance from the median value to Q3, so this choice of \( \text{kernel}_s \) was not considered further. The choice of \( \text{kernel}_s = 10 \) can be further verified using boxplots of other parameters such as Acc, Pre, QI, and AUC, as illustrated in Figure 3. The optimal values selected for \( n_{\text{filters}} = 1 \) and \( \text{kernel}_s = 10 \) were used further for various dataset imbalance handling scenarios, as discussed subsequently.

5.3 Data balancing methods

The data imbalance in CTU-UHB regarding 506 normal delivery records (controls) and 46 cesarean section records (cases) was addressed by adopting the strategies outlined in subsections from Sections 5.3.1 to 5.3.4. The 1D-CNN model’s layer-wise settings and training hyperparameters were as per Tables 2 and 3, with \( n_{\text{filters}} = 1 \) and considering \( \text{kernel}_s = 10 \) for the 1D-CNN layer in the proposed model. A total of 31 experiments were conducted for each strategy. Statistical parameters were used to assess the model’s performance following Section 3.

5.3.1 Unbalanced dataset

The first method can be viewed as a base condition in which the dataset was assessed without class balancing. The model was trained using 50 epochs and 64 training examples per batch. The training loss function and training Acc function are depicted in Figure 4 in parts (a) and (b), respectively, indicating training stability and convergence. The model was trained using the cross-fold technique with 10 folds, as explained in Section 5.2.

5.3.2 Undersampling

In the undersampling scenario, all 46 case records were chosen, and 46 control records were chosen randomly from 506 records to match the total number of case records. Thus, 92 recordings were considered for the experiment. These 92 records were separated into a training split of 80%, a testing split of 20% of total records, and a validation split of 10% from the training split. The model underwent training with 200 epochs and 64 records per batch.
5.3.3 The weighted binary cross-entropy method

The imbalance in the dataset was addressed using the weighted binary cross-entropy approach. The weights were assigned to each class considering the proportion of 8.33% minority class vs 91.66% majority class. The weights resulted in one misclassification from the case category, contributing to as many as 11

Figure 3: The boxplots for averaged performance parameters using different values of $\text{kernel}_s = [5, 10, 15, 20, 25, 30]$ across 10 folds. From top-left to top-right: Acc, Pre; from middle-left to middle-right: Se, Sp; from bottom-left to bottom-right: QI, and AUC. (a) Acc boxplot, (b) Pre boxplot, (c) Se boxplot, (d) Sp boxplot, (e) QI boxplot, and (f) AUC boxplot.

5.3.3 The weighted binary cross-entropy method

The imbalance in the dataset was addressed using the weighted binary cross-entropy approach. The weights were assigned to each class considering the proportion of 8.33% minority class vs 91.66% majority class. The weights resulted in one misclassification from the case category, contributing to as many as 11
misclassifications in the control category (considering 46 cases vs 506 controls in the dataset); the cross-fold method with 10 folds, a batch size of 64 examples. The 1D-CNN model was converged with 200 epochs during the training phase. The training logloss and Acc functions can be observed in Figure 5.

5.3.4 Oversampling using SMOTE

In the current approach, using SMOTE for oversampling, the minority class was made proportionate during the training phase. The cross-fold validation method was used with 10 folds, of which ninefolds were sent for SMOTE and then utilized for training, and onefold was preserved for testing. The model was empirically examined with different epoch values, and its performance was most remarkable for epoch 30, which is an outstanding accomplishment for the suggested model. The training logloss and Acc functions are illustrated in Figure 6.
6 Results and discussion

The section enlists the results obtained in each data balancing approach. As discussed in Section 5, 31 simulations were carried out for each scenario, and the median (based on QI value) observation’s statistical parameters are reported here.

6.1 Unbalanced dataset

The unbalanced dataset was used as input for the 1D-CNN model, and Table 4 shows the various statistical parameters’ values during the training and testing phases. The median Se, Sp, and QI values were 0.9980, 0.1900, and 0.4355, whereas the AUC was 0.7140. The high value of Se shows that the model accurately detected cases of normal delivery, whereas low Sp suggests that cesarean section delivery cases were inadequately classified. The unbalanced data may be a contributing factor to this bias. The model demonstrated exceptional.

Table 4: 1D-CNN model result for different class balance scenarios

<table>
<thead>
<tr>
<th>Phase</th>
<th>Loss</th>
<th>Acc</th>
<th>Pre</th>
<th>Se</th>
<th>Sp</th>
<th>QI</th>
<th>F1-Score</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unbalanced dataset</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training (fold = 1)</td>
<td>0.2774</td>
<td>0.9173</td>
<td>0.9173</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.9569</td>
<td>0.6027</td>
</tr>
<tr>
<td>Training (fold = 10)</td>
<td>0.1547</td>
<td>0.9517</td>
<td>0.9576</td>
<td>0.9912</td>
<td>0.5100</td>
<td>0.7125</td>
<td>0.9741</td>
<td>0.9058</td>
</tr>
<tr>
<td>Testing</td>
<td>0.2304</td>
<td>0.9312</td>
<td>0.9324</td>
<td>0.998</td>
<td>0.1900</td>
<td>0.4355</td>
<td>0.9641</td>
<td>0.7140</td>
</tr>
<tr>
<td><strong>Undersampling</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training</td>
<td>0.2119</td>
<td>0.9538</td>
<td>0.9429</td>
<td>0.9706</td>
<td>0.9355</td>
<td>0.9529</td>
<td>0.9565</td>
<td>0.9853</td>
</tr>
<tr>
<td>Validation</td>
<td>0.7354</td>
<td>0.50</td>
<td>0.5714</td>
<td>0.80</td>
<td>0.00</td>
<td>0.00</td>
<td>0.6666</td>
<td>0.30</td>
</tr>
<tr>
<td>Testing</td>
<td>0.8042</td>
<td>0.5263</td>
<td>0.4286</td>
<td>0.8571</td>
<td>0.3333</td>
<td>0.5345</td>
<td>0.5714</td>
<td>0.6667</td>
</tr>
<tr>
<td><strong>Weighted binary cross-entropy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training (fold = 1)</td>
<td>0.0148</td>
<td>0.9839</td>
<td>0.9956</td>
<td>0.9868</td>
<td>0.9512</td>
<td>0.9688</td>
<td>0.9912</td>
<td>0.9967</td>
</tr>
<tr>
<td>Training (fold = 10)</td>
<td>0.0079</td>
<td>0.9940</td>
<td>1.00</td>
<td>0.9934</td>
<td>1.00</td>
<td>0.9967</td>
<td>0.9967</td>
<td>0.9953</td>
</tr>
<tr>
<td>Testing</td>
<td>0.2062</td>
<td>0.9458</td>
<td>0.9784</td>
<td>0.9625</td>
<td>0.7650</td>
<td>0.8581</td>
<td>0.9704</td>
<td>0.9180</td>
</tr>
<tr>
<td><strong>SMOTE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training (fold = 1)</td>
<td>0.0601</td>
<td>0.9758</td>
<td>0.9656</td>
<td>0.9868</td>
<td>0.9648</td>
<td>0.9758</td>
<td>0.9761</td>
<td>0.9990</td>
</tr>
<tr>
<td>Training (fold = 10)</td>
<td>0.0003</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Testing</td>
<td>0.0550</td>
<td>0.9875</td>
<td>0.9891</td>
<td>0.9981</td>
<td>0.8800</td>
<td>0.9372</td>
<td>0.9935</td>
<td>0.9510</td>
</tr>
</tbody>
</table>
Acc, Pre, and F1-score performance during the test phase. Figure 7 shows the individual fold and overall mean AUC values.

Figure 7: Fold-wise AUC values (unbalanced dataset).

6.2 Undersampling

The undersampling scenario considered 46 records from both classes. Table 4 lists the results for the undersampling example during the training, validation, and testing phases. This experiment produced low Se, Sp, QI, and AUC levels. The low results may be attributed to the small number of training instances used to build the model.

6.3 The weighted binary cross-entropy method

The weighted binary cross-entropy method resulted in boxplots for Se, Sp, and QI, as shown in the top row of Figure 8. The Se parameter ranged from 0.8757 (min) to 1.00 (max). The range of the Sp parameter was found to be 0.48 (min) to 0.9 (max). The low variance in the Se parameter suggests the model's stability. The median Se, Sp, and QI are reported in Table 4 as 0.9625, 0.7650, and 0.8581, respectively. The Se value is adequate for identifying normal delivery records. The results obtained for the Sp and QI parameters are much better than those of the unbalanced dataset. Other metrics, such as Acc, Pre, and F1-Score, are also improved in the current scenario. Figure 9 provides the fold-wise AUC and mean AUC values.

6.4 Oversampling using SMOTE

The SMOTE scenario used in the training phase resulted in Se, Sp, and QI boxplots, as represented by the bottom row in Figure 10. The range of Se parameter was 0.9881 (min) to 1.00 (max). The low variance from median to Q3 and zero variance from Q3 to max in the Se boxplot indicate a stable and consistent result. The range of the Sp parameter was found to be 0.81 (min)–0.92 (max). Once again, a low variance from the median to the Q3 value assures the model's stability. The median Se value of 0.9981 and the Sp value of 0.88 are well balanced enough to discriminate between normal and cesarean sections, as supported by the outstanding QI of 0.9372 and AUC of 0.9510. The median values are represented in Table 4. Figure 11 depicts the fold-based and mean AUC values.
Figure 8: The boxplots for averaged performance parameters using $n_{\text{filter}} = 1$ and $kernel_s = 10$ across 10 folds under the weighted binary cross-entropy scenario. From top-left to top-right: Se, Sp; from bottom-left to bottom-right: QI, and AUC. (a) Se boxplot, (b) Sp boxplot, (c) QI boxplot, and (d) AUC boxplot.

Figure 9: Fold-wise AUC values (weighted binary cross-entropy).
Figure 10: The boxplots for averaged performance parameters using $n_{\text{filter}} = 1$ and $k_{\text{ernel}} = 10$ across 10 folds under SMOTE scenario. From top-left to top-right: $\text{Se}$, $\text{Sp}$; from bottom-left to bottom-right: QI, and AUC. (a) $\text{Se}$ boxplot, (b) $\text{Sp}$ boxplot, (c) QI boxplot, and (d) AUC boxplot.

Figure 11: Fold-wise AUC values (SMOTE).
6.5 Comparison with existing literature

The findings of the suggested methodology are compared with the cutting-edge methods, with the CTU-UHB dataset serving as the fundamental criterion for the comparison. Table 5 summarizes the comparison’s results. The comparison uses the pertinent metrics Se, Sp, QI, and AUC. Case and control values are also mentioned since they influence the interpretation of the Se and Sp parameters. In the SMOTE oversampling situation, the Se and QI of the proposed technique are greater than those described in [17], while Sp is slightly lesser. In addition, complex engineering of features was required in [17]. Considering the scenario with weighted binary cross-entropy, the proposed approach has shown superior results than AUC [19] and [23]. Although Sp and AUC parameters in [21] are superior to the proposed method in the SMOTE scenario, Se is much better in the suggested method, and QI is almost equal to that in [21]. In addition, Saleem et al. [21] required complex feature engineering. Given the condition of an unbalanced dataset in [22] and the current work, the Se is still higher in this situation. Although [24] yielded better results regarding Sp, QI, and AUC, a few records from the CTU-UHB dataset were chosen for this study. In contrast, the present research used all available records, and Se is still performing better. The current analysis used a one-dimensional FHR signal, while a two-dimensional image generated from a one-dimensional FHR signal was used as input for the 2D-CNN method in [24]. The generation of two-dimensional images is an overhead and will require more computational power. In addition, one notable limitation when employing 2D-CNN is high computational requirements, which demand specialized hardware, especially for training. In this regard, 2D-CNN is not appropriate for applications that operate in real-time on mobile and low-power/low-memory devices [13]. All the parameters are higher in the proposed method than those in the study by Alsaggaf et al. [25] with the requirement of significant feature engineering. The segmentation approach was employed for class balancing in [27], and the current system’s performance utilizing the SMOTE scenario is superior for all assessment parameters. GAN addressed the imbalance in the dataset [28], but the suggested scheme excelled in all parameters.

6.6 Discussion

The CTG is an important fetal monitoring instrument during the intrapartum and antepartum phases. The CTG tracings are evaluated according to the FIGO criteria, yet inter-observer and intra-observer variability remains and may lead to unnecessary surgical births and cesarean sections. Recently, machine learning techniques have improved classification performance by reducing variance. An optimal set of input features, which remains challenging and time-consuming, influences the classification performance. The proposed model employed a 1D-CNN, suitable for a one-dimensional input, and has the notable trait of combining the feature mining and categorization methods in a single unit. The class balance is rarely present in medical datasets and is the second aspect influencing classification performance. The current study addresses the class imbalance problem using various scenarios: (i) considering the original unbalanced dataset; (ii) undersampling the dataset; (iii) employing the weighted binary cross-entropy method; and (iv) oversampling the dataset with SMOTE. The parameter tuning for the 1D-CNN model is done for two essential parameters: n\textunderscore filters and kernel\_s, as it highly impacts the overall performance of convolutional operation. The Se for the imbalanced dataset is 99.80%, which is an excellent classification rate for true positives (normal delivery cases) and suggests that unnecessary operational deliveries can be avoided. The bias introduced by an unbalanced dataset may account for a lower Sp value in this scenario. Due to the insufficient training examples, the outcome of the undersampling scenario needed improvement. During the weighted binary cross-entropy scenario, the 1D-CNN model performs significantly better than the unbalanced scenario, as Sp increases by 57.5%, QI by 42.26%, and AUC by 20.4%. However, a slight decrease in the Se by 3.55% can be observed here. The overall outcome due to the weighted cross-entropy method is significantly better than the unbalanced scenario. Compared to the weighted binary cross-entropy technique, SMOTE significantly increased around 11.5% in Sp, 7.91% in Q1, 3.3% in AUC, and 3.56% in Se. Figure 12 shows a comparison graph of the four crucial parameters, Se, Sp, QI, and AUC, generated using four class balancing scenarios: unbalanced dataset,
Table 5: Comparison with state-of-the-art

<table>
<thead>
<tr>
<th>Method</th>
<th>Data division criteria</th>
<th>Case and control value</th>
<th>Class balancing approach</th>
<th>Se (%)</th>
<th>Sp (%)</th>
<th>QI (%)</th>
<th>AUC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensemble classifier (FLDA, RF, SVM) [17]</td>
<td>Type of delivery</td>
<td>Case:1 Control:0</td>
<td>SMOTE</td>
<td>87.00</td>
<td>90.00</td>
<td>88.49</td>
<td>96.00</td>
</tr>
<tr>
<td>CNN [19]</td>
<td>pH value</td>
<td>Case:1 Control:0</td>
<td>Weighted binary cross-entropy</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>82.00</td>
</tr>
<tr>
<td>AdaBoost [21]</td>
<td>Type of delivery</td>
<td>Case:0 Control:1</td>
<td>SMOTE</td>
<td>91.80</td>
<td>95.50</td>
<td>93.63</td>
<td>98.00</td>
</tr>
<tr>
<td>SVM [22]</td>
<td>pH value</td>
<td>Case:1 Control:0</td>
<td>—</td>
<td>77.40</td>
<td>93.86</td>
<td>85.23</td>
<td>88.74</td>
</tr>
<tr>
<td>MCNN and stacked MCNN [23]</td>
<td>pH value</td>
<td>Case:1 Control:0</td>
<td>Weighted binary cross-entropy</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>82.00</td>
</tr>
<tr>
<td>CNN [24]</td>
<td>pH value</td>
<td>Case:0 Control:1</td>
<td>Down sampling + image augmentation</td>
<td>99.29</td>
<td>98.10</td>
<td>98.69</td>
<td>98.70</td>
</tr>
<tr>
<td>SVM [25]</td>
<td>pH value, BDecf, Apgar score (1m &amp; 5m)</td>
<td>Case:1 Control:0</td>
<td>—</td>
<td>74.29</td>
<td>99.55</td>
<td>86.00</td>
<td>89.30</td>
</tr>
<tr>
<td>CNN [27]</td>
<td>Type of delivery</td>
<td>Case:0 Control:1</td>
<td>Segmentation</td>
<td>80.00</td>
<td>79.00</td>
<td>79.50</td>
<td>86.00</td>
</tr>
<tr>
<td>CNN [28]</td>
<td>pH Value</td>
<td>Case:1 Control:0</td>
<td>GAN data augmentation</td>
<td>67.64</td>
<td>71.97</td>
<td>69.77</td>
<td>—</td>
</tr>
<tr>
<td>Proposed method (1D-CNN)</td>
<td>Type of delivery</td>
<td>Case:0 Control:1</td>
<td>1. Unbalanced dataset</td>
<td>99.80</td>
<td>19.00</td>
<td>43.55</td>
<td>71.40</td>
</tr>
<tr>
<td>Type of delivery</td>
<td>Case:0 Control:1</td>
<td>2. Undersampling</td>
<td>85.71</td>
<td>33.33</td>
<td>53.45</td>
<td>66.67</td>
<td></td>
</tr>
<tr>
<td>Type of delivery</td>
<td>Case:0 Control:1</td>
<td>3. Weighted binary cross-entropy</td>
<td>96.25</td>
<td>76.50</td>
<td>85.81</td>
<td>91.80</td>
<td></td>
</tr>
<tr>
<td>Type of delivery</td>
<td>Case:0 Control:1</td>
<td>4. SMOTE</td>
<td>99.81</td>
<td>88.00</td>
<td>93.72</td>
<td>95.10</td>
<td></td>
</tr>
</tbody>
</table>
undersampling, weighted binary cross-entropy, and SMOTE. The graph demonstrates that the 1D-CNN model outperforms in terms of Sp, QI, and AUC in the SMOTE scenario compared to the rest.

The boxplots for Se, Sp, QI, and AUC also mark stability and consistent performance due to the very low variance in the acceptable range during the SMOTE scenario, as shown in Figure 10.

The Se plays a crucial role in preventing unnecessary interventions during labor. Apart from undersampling, the suggested model is robust enough to demonstrate consistent Se in all circumstances. The Sp is a crucial metric as it helps avoid fetal compromise; hence, a rise in Sp in the SMOTE scenario is paramount. Although the Sp value in the training phase is relatively high in the SMOTE scenario due to the availability of oversampled case records, it dropped during the testing phase. The reason might be a smaller number of case records available during the testing phase. The learning rate decay and batch normalization method lead to early network convergence, ultimately lowering the number of training epochs in unbalanced and SMOTE conditions. This feature makes the proposed network worth using with portable devices, where less memory and power requirements must be satisfied. Second, the low computational requirement of the 1D-CNN model, without any additional overhead for input preparation during the testing phase, makes it suitable for real-time applications. The system’s stable and consistent performance may be due to correctly handling zero values in the FHR signal during the input normalization phase. The most crucial factor significantly impacting the overall performance is the proper selection of n\_filters and kernel\_s for the convolutional layer. After conducting many trials in each case, the authors carefully selected these values during fine-tuning experimentation. The fine-tuning of these values leads to satisfactory results regarding Se, Sp, QI, and AUC. The experiments for hyperparameter setting and layer-wise parameters also contributed to the overall results. However, the authors strongly feel that other deep learning methods combined with CNN might help increase overall predictive capability in SP, QI, and AUC values. The overall outcomes are promising for using this SMOTE method to avoid unnecessary cesarean sections and to detect fetal compromise. The comparison with the state-of-the-art methods is also encouraging toward the model’s usefulness in clinical practices.

7 Conclusion

The proposed framework automatically extracts features from an FHR signal using a 1D-CNN method. The experimentation employed a dataset with open access. The dataset was balanced using undersampling,
weighted binary cross-entropy, and SMOTE methods. In all circumstances, besides undersampling, the sensitivity is very promising. The SMOTE scenario resulted in a Se of 99.81%, a Sp of 88%, a QI of 93.72%, and an AUC of 95.10%. Thus, the model has applicability in avoiding unnecessary cesarean sections and fetal compromise during the antepartum and intrapartum phases. In addition, the SMOTE scenario required a few epochs during the 1D-CNN training phase, significantly reducing training time. The proposed one-dimensional method is simple and does not require much input preparation compared to the two-dimensional model, and due to this lack of overhead, it will undoubtedly be helpful in real-time applications on portable low-memory/power devices.

In future work, we will evaluate the robustness of the suggested model using other datasets. In addition, different deep-learning variants adapted to time-series input can be used.

Acknowledgement: The authors thank the editor and referees for their insightful comments and suggestions.

Funding information: The authors state no funding involved.

Author contributions: Vidya Sujit Kurtadikar: conceptualization & design, methodology, software, validation, writing-original draft. Himangi Milind Pande: supervision, writing – reviewing and editing.

Conflict of interest: The authors state no conflicts of interest.

Data availability statement: The datasets analysed during the current study are available in the physionet repository, https://physionet.org/content/ctu-uhb-ctgdb/1.0.0/.

References


