Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

Deep Learning-Based Fetal Health Classification: A Comparative Analysis of Convolutional and Recurrent Neural Networks

Gregorius Airlangga

Information Systems Study Program, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia
Email: gregorius.airlangga@atmajaya.ac.id
Submitted: 24/02/2025; Accepted: 28/02/2025; Published: 28/02/2025

Abstract−Fetal health monitoring plays a crucial role in prenatal care, enabling early detection of complications that may impact pregnancy outcomes. Traditional methods, including cardiotocography (CTG), rely on expert interpretation, which can introduce variability and potential misdiagnoses. In this study, deep learning techniques are employed to classify fetal health conditions based on CTG data. A comparative analysis is conducted on six architectures: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional LSTM (BiLSTM), and Attention-based LSTM. The models are evaluated using accuracy, precision, recall, and F1-score under a 10-fold cross-validation framework. Results indicate that CNN outperforms all other models, achieving an accuracy of 97.18% due to its hierarchical feature extraction capabilities. GRU demonstrates competitive performance with an F1-score of 95.50% while maintaining computational efficiency. The study further includes a complexity analysis, revealing that recurrent models, particularly BiLSTM and Attention-LSTM, introduce significant computational overhead without yielding substantial performance gains. Potential threats to validity, including dataset bias and overfitting, are analyzed to ensure robust findings. The insights gained from this research highlight the advantages of CNN-based architectures in automated fetal health assessment and suggest future work integrating hybrid models and explainable AI techniques. These findings contribute to advancing AI-driven fetal monitoring systems, aiding clinical decision-making, and improving perinatal care.

Keywords: Fetal Health Classification; Deep Learning; Convolutional Neural Networks; Recurrent Neural Networks; Cardiotocography

1. INTRODUCTION

Articles Fetal health assessment is a critical component of prenatal care, enabling timely intervention to prevent complications during pregnancy [1]–[3]. Accurate classification of fetal health conditions is essential for minimizing risks associated with fetal distress, preterm birth, and other perinatal complications. Traditional methods for fetal health monitoring rely on clinical expertise and conventional diagnostic techniques, including non-stress tests (NSTs) and cardiotocographic (CTG) readings [4]–[7]. While these approaches have been widely adopted in medical practice, their effectiveness is often influenced by human interpretation, leading to variability in diagnostic accuracy [5]. Additionally, existing manual evaluation methods require extensive training and expertise, making them impractical for widespread deployment in resource-constrained settings [8]. To address these challenges, the integration of machine learning (ML) and deep learning (DL) techniques has emerged as a promising solution to enhance diagnostic precision, automate fetal health assessment, and provide standardized evaluation metrics [9].

Several studies have explored the application of machine learning and deep learning techniques for fetal health classification. Early research efforts primarily focused on using classical machine learning algorithms, such as Support Vector Machines (SVM), Random Forest (RF), and k-Nearest Neighbors (k-NN), to classify fetal health conditions based on extracted features from cardiotocographic (CTG) signals [4], [10], [11]. For instance, [12] applied SVM for fetal distress classification and achieved promising accuracy levels. Similarly, [13] demonstrated the efficacy of Random Forest classifiers in predicting fetal health conditions. While these approaches provided valuable insights, they often relied heavily on manual feature extraction, limiting their adaptability to diverse datasets. Recent advances in deep learning have led to significant improvements in fetal health classification by leveraging neural networks for automated feature extraction. Convolutional Neural Networks (CNNs) have been utilized in several studies to analyze CTG signals, as they effectively capture spatial features from sequential data. For example, [14] employed CNNs for fetal heart rate classification and reported superior performance compared to traditional ML models. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), have also been investigated for fetal health classification due to their ability to capture temporal dependencies in sequential data. [15]–[17] implemented an LSTM-based model for classifying fetal health status and demonstrated improved accuracy over static feature-based classifiers.

Moreover, attention mechanisms have been introduced in fetal health classification to enhance the model's ability to focus on crucial time steps in sequential data [18] integrated an Attention-LSTM model for fetal heart rate analysis, leading to better performance in distinguishing normal, suspicious, and pathological fetal conditions. However, comparative analyses of multiple deep learning architectures within a unified experimental framework remain limited, leaving gaps in understanding the relative effectiveness of different models for fetal health classification. In this study, we investigate the effectiveness of multiple deep learning models in classifying fetal health conditions based on a publicly available dataset. Our approach involves

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

preprocessing the dataset by applying feature scaling and categorical encoding techniques, followed by reshaping the data to accommodate different deep learning architectures. Specifically, we evaluate six distinct models: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM), Gated Recurrent Unit (GRU), Bidirectional LSTM (BiLSTM), and Attention-based LSTM. These models are selected to capture both spatial and temporal dependencies in fetal health data, providing a comprehensive analysis of their predictive performance. The inclusion of attention mechanisms in deep learning architectures has been shown to enhance feature selection, and its effectiveness in fetal health classification is further explored in this study.

To ensure robustness, we employ a rigorous 10-fold cross-validation strategy, which partitions the dataset into training and validation sets, reducing bias and ensuring a fair comparison across models. Each model is trained using the Adam optimizer and optimized for categorical cross-entropy loss. Performance is assessed using four key evaluation metrics: accuracy, precision, recall, and F1-score. By comparing these models under the same experimental conditions, we provide an in-depth evaluation of their strengths and limitations, identifying the most effective architectures for fetal health assessment. Additionally, we explore the impact of hyperparameter tuning on classification performance and discuss the computational efficiency of each model to assess their feasibility for real-time deployment in clinical settings. The primary contribution of this research lies in the comparative analysis of various deep learning models in the context of fetal health classification. While previous studies have focused on the application of individual deep learning architectures, there is a lack of comprehensive evaluations that systematically compare multiple approaches within the same experimental framework. Additionally, our study explores the impact of attention mechanisms, which have gained popularity in sequence modeling tasks, by implementing an Attention-based LSTM model. This model aims to improve feature selection by assigning different levels of importance to different time steps, potentially enhancing classification performance. Moreover, our study evaluates the generalization capability of each model by performing extensive cross-validation, ensuring that our findings can be applied to real-world clinical scenarios.

The findings of this research are expected to provide valuable insights into the applicability of deep learning models in fetal health monitoring. The results can serve as a foundation for future research aimed at developing real-time fetal health assessment tools, integrating deep learning-based models into clinical decision support systems, and improving early detection of fetal distress. Furthermore, this study contributes to the growing body of knowledge in medical AI by benchmarking deep learning techniques against traditional ML approaches in the domain of obstetric healthcare. The insights derived from this research can potentially guide future studies focused on developing hybrid models that integrate both ML and DL techniques to achieve enhanced diagnostic accuracy. The remainder of this paper is organized as follows. Section 2 details the dataset, preprocessing techniques, and deep learning architectures employed in this study. Section 3 describes the experimental setup, including hyperparameter selection, training configurations, and evaluation metrics. Furthermore, we discuss the results obtained from 10-fold cross-validation, comparing the performance of different models and analyzing their strengths and limitations. Finally, Section 4 concludes the study by summarizing key findings, discussing potential applications in real-world clinical environments, and proposing directions for future research in fetal health classification using deep learning methodologies.

2. RESEARCH METHODOLOGY

The research methodology employed in this study follows a structured approach to ensure robust and reliable classification of fetal health conditions using deep learning models as presented in the table 1. The overall workflow is designed to encompass data preprocessing, model training, evaluation, and result analysis, as illustrated in the activity diagram. The methodology is systematically structured to optimize model performance while addressing common challenges such as class imbalance, feature variability, and generalization issues in machine learning models. The process begins with data preprocessing, which plays a crucial role in ensuring the quality and consistency of input features. The dataset is first loaded and preprocessed to remove any inconsistencies. Categorical labels, representing fetal health conditions, are one-hot encoded to facilitate model training, while numerical features are normalized using Z-score transformation to standardize the feature distribution. Since deep learning models often require specific input dimensions, the dataset is also reshaped accordingly. Additionally, Synthetic Minority Over-sampling Technique (SMOTE) is applied to balance the dataset by generating synthetic samples for underrepresented fetal health conditions. This step is essential in mitigating the effects of class imbalance, which can lead to biased predictions favoring the majority class.

Following data preprocessing, the dataset is partitioned into training and testing subsets, with 80% of the data used for training and 20% reserved for model evaluation. This ensures that the models learn from a substantial portion of the dataset while still having an independent test set for performance assessment. Next, multiple deep learning architectures are initialized, including Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRU), Bidirectional LSTM (BiLSTM), and Attention-based LSTM. The inclusion of these models allows for a comparative analysis of different architectures to determine the most effective one for fetal health classification.

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

The training and evaluation phase is implemented in an iterative manner, ensuring continuous refinement of the models until the best-performing architecture is identified. Initially, models are trained using the Adam optimizer, a widely used optimization algorithm known for its efficiency in deep learning tasks. Hyperparameter tuning is performed concurrently to optimize learning rates, batch sizes, and network depths, thereby enhancing model accuracy and convergence stability. To ensure robust model evaluation, k-fold cross-validation is employed, partitioning the dataset into multiple subsets where training and validation are conducted iteratively. After each training cycle, model performance is assessed using key evaluation metrics, including accuracy, precision, recall, and F1-score. If the best-performing model is not yet identified, the training loop continues, refining the models further.

Once the best model is found, it undergoes retraining and optimization to ensure optimal performance. The refined model is then saved for future deployment. To verify its effectiveness, the trained model is tested on the independent test set, where its predictive capability on unseen data is measured. The final step involves an analysis and comparison of results, highlighting the model's strengths and limitations in classifying fetal health conditions. This methodological approach ensures that the classification process is systematic, rigorous, and reproducible, leading to reliable outcomes that can be integrated into clinical decision-making systems. By leveraging multiple deep learning architectures, addressing data imbalance, and implementing extensive evaluation techniques, this study provides a comprehensive assessment of machine learning models in fetal health classification.

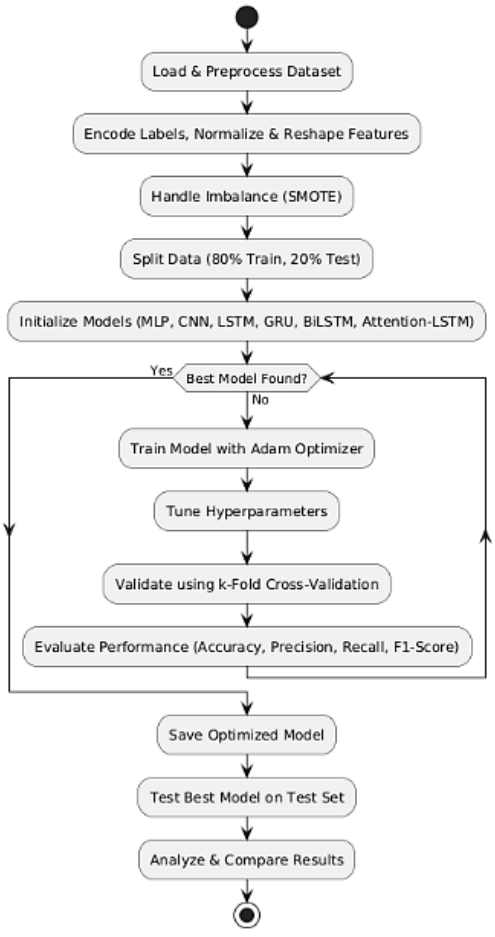

Figure 1. Research Methodology Diagram

2.1 Dataset Description

The dataset used in this study is derived from fetal cardiotocographic (CTG) signals, consisting of ($N = 2126$) instances, each representing a fetal monitoring test. The dataset can be downloaded from [19], [20]. The dataset comprises a set of features ($X \in R^{N \times d}$), where ($d = 21$) represents the number of physiological parameters recorded during CTG examinations. Each feature (x_i) corresponds to a real-valued attribute describing fetal heart rate (FHR) variability, uterine contractions, acceleration and deceleration patterns, and histogram-based characteristics extracted from FHR signals. The target variable is denoted as (y), where ($y \in \{1,2,3\}$), representing the three fetal health conditions: normal, suspect, and pathological. The fetal health condition is a categorical variable indicating whether a given fetal state is normal (($y = 1$)), suspect (($y =$

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

2)), or pathological (($y = 3$)). The dataset distribution is imbalanced, with a majority of instances labeled as normal. The proportion of each class is represented as $(p(y) = \{p_1, p_2, p_3\})$, where $(p_1 \approx 0.78)$, $(p_2 \approx 0.14)$, and $(p_3 \approx 0.08)$. The imbalance in class distribution can influence model performance, particularly in underrepresented classes, necessitating appropriate handling during training.

Each feature in (X) represents a continuous physiological measure and can be defined as a mapping function $(f: R^N \rightarrow R)$, where each $(x_{i,j})$ corresponds to the (j)-th feature of the (i)-th sample. Given the nature of fetal health classification, features such as baseline FHR, short-term variability, and long-term variability exhibit correlations with fetal distress. The feature set can be expressed as (1).

$$X = \{x_1, x_2, \dots, x_d\} \tag{1}$$

Where each (x_j) represents a different physiological measure. Mathematically, the dataset can be viewed as a triplet $((X, y, \mathcal{D}))$, where (X) is the feature matrix, (y) is the target label, and $(\mathcal{D})$ is the joint probability distribution over (X) and (y), represented as $P(X, y)$. The classification task aims to approximate the posterior probability as presented in (2).

$$P(y \mid X) \tag{2}$$

Where a deep learning model (f_θ) with parameters (θ) is trained to optimize a function mapping (X) to (y) such (3).

$$\hat{y} = \arg\max_y P(y \mid X, \theta) \tag{3}$$

The dataset's attributes exhibit different ranges, necessitating a preprocessing step to normalize values for stable model training. Given that some features have skewed distributions, the standardization transformation is defined as (4).

$$X'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j} \tag{4}$$

Where (μ_j) and (σ_j) denote the mean and standard deviation of feature (j), respectively. This dataset is publicly available and serves as a benchmark for fetal health classification using machine learning and deep learning models. The structure of the dataset provides an opportunity to evaluate models capable of handling imbalanced classes while extracting meaningful patterns from fetal heart rate signals.

2.2 Data Preprocessing Techniques

The dataset preprocessing phase is essential for ensuring that the input features are appropriately formatted and normalized before being used in deep learning models. The preprocessing techniques applied in this study involve data cleaning, encoding categorical labels, feature scaling, and reshaping the dataset for compatibility with different neural network architectures. The first step in preprocessing involves verifying the presence of missing values in the dataset. Given a dataset represented as a matrix $(X \in R^{N \times d})$, where (N) denotes the number of samples and (d) represents the number of features, the presence of missing values in any feature (x_j) is determined as (5).

$$\forall i, j \quad \text{if} \quad x_{i,j} = \emptyset, \quad \text{then apply imputation or removal} \tag{5}$$

Where $(x_{i,j})$ is the value of the (j)-th feature for the (i)-th instance. In this dataset, no missing values were found, thus eliminating the need for imputation techniques. The categorical target variable (y) representing fetal health conditions is transformed into a one-hot encoded vector. Given that (y) has three classes: normal $((y = 1))$, suspect (($y = 2$)), and pathological (($y = 3$)), the one-hot encoding process transforms each (y_i) into (6).

$$y_i = [y_{i1}, y_{i2}, y_{i3}] \tag{6}$$

Feature scaling is applied to standardize numerical features. Since each feature (x_j) has a different range of values, a standardization transformation is performed using Z-score normalization as presented in (7).

$$x'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j} \tag{7}$$

Where (μ_j) and (σ_j) represent the mean and standard deviation of feature (x_j), respectively as (8) and (9).

$$\mu_j = \frac{1}{N} \sum_{i=1}^{N} x_{i,j} \tag{8}$$

$$\sigma_j = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(x_{i,j} - \mu_j\right)^2} \tag{9}$$

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

This transformation ensures that all features have zero mean and unit variance, preventing features with larger magnitudes from dominating the learning process. For deep learning architectures that process sequential inputs, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the dataset is reshaped into a three-dimensional tensor. The original feature matrix ($X' \in R^{N\times d}$) is transformed into (10).

$$X'_{\text{reshaped}} \in R^{N\times d\times 1} \tag{10}$$

Which maintains the feature dimensionality but introduces a singleton dimension required for processing one-dimensional convolutions and recurrent layers. To improve generalization and avoid overfitting, data augmentation techniques are explored, particularly for underrepresented classes. Synthetic Minority Over-sampling Technique (SMOTE) is applied to balance the dataset distribution by generating synthetic samples for minority classes. Given a feature vector (x_i) belonging to the minority class, a synthetic sample (x_{new}) is generated as (11).

$$x_{\text{new}} = x_i + \lambda(x_k - x_i) \tag{11}$$

Where (x_k) is a randomly chosen nearest neighbor of (x_i), and (λ) is a random value in the range ($[0,1]$). This technique effectively addresses the class imbalance and enhances the model's ability to learn meaningful patterns from all categories. The final preprocessed dataset is then partitioned into training and testing sets. Let ($\mathcal{D} = \{(X', y)\}$) represent the entire dataset, then it is split into (12).

$$\mathcal{D}_{train} \cup \mathcal{D}_{\text{ts}} = \mathcal{D}, \quad \mathcal{D}_{train} \cap \mathcal{D}_{test} = \{\} \tag{12}$$

Where, $\mathcal{D}_{\text{tan}} = 0.8N, \quad |\mathcal{D}_{\text{ts}}| = 0.2N$ ensuring that (80%) of the data is used for training while (20%) is held out for evaluation. These preprocessing techniques ensure that the dataset is optimally prepared for training deep learning models, enhancing convergence stability, reducing class imbalance effects, and improving overall predictive performance.

2.3 Deep Learning Architectures

In this study, multiple deep learning architectures are employed to classify fetal health conditions. Each model is designed to process the input feature matrix ($X \in R^{N\times d}$), where (N) represents the number of samples, and (d) is the number of features. The models aim to learn a function (f_θ) parameterized by (θ), which maps the input features to the probability distribution of the output classes as presented as (13).

$$P(y \mid X, \theta) = f_\theta(X) \tag{13}$$

Where (y) represents the class labels. The following neural network architectures are implemented in this study: multilayer perceptron (MLP), convolutional neural network (CNN), long short-term memory network (LSTM), gated recurrent unit (GRU), bidirectional LSTM (BiLSTM), and an attention-based LSTM model. The multilayer perceptron (MLP) is a fully connected feedforward network, where each hidden layer consists of a transformation function defined as (14).

$$h^{(l)} = \sigma\big(W^{(l)}h^{(l-1)} + b^{(l)}\big)] \tag{14}$$

For layer (l), where ($h^{(l)}$) is the activation of the (l)-th layer, ($W^{(l)}$) is the weight matrix, ($b^{(l)}$) is the bias term, and (σ) is the non-linear activation function. The final output layer applies the softmax activation function as presented as (15).

$$P(y \mid X, \theta) = \text{softmax}\big(W^{(L)}h^{(L-1)} + b^{(L)}\big) \tag{15}$$

Where (L) is the total number of layers. The convolutional neural network (CNN) extracts spatial features by applying convolutional operations. The transformation at each layer is defined as (16).

$$h^{(l)} = \sigma\big(W^{(l)} * h^{(l-1)} + b^{(l)}\big) \tag{16}$$

Where ($*$) represents the convolution operation. The CNN consists of convolutional layers followed by max-pooling operations as presented as (17).

$$h^{(l)}_{\text{pool}} = \max_{\forall i\in\mathcal{R}} h_i^{(l)} \tag{17}$$

Where ($\mathcal{R}$) denotes the pooling region. The final feature map is flattened and passed through dense layers before classification. The long short-term memory network (LSTM) captures temporal dependencies by maintaining a cell state (c_t) and hidden state (h_t). The LSTM cell updates are defined by (18) – (23).

$$\mathrm{f_t} = \sigma(\mathrm{W_f x_t} + \mathrm{U_f h_{t-1}} + \mathrm{b_f}) \tag{18}$$

$$\mathrm{i_t} = \sigma(\mathrm{W_i x_t} + \mathrm{U_i h_{t-1}} + \mathrm{b_i}) \tag{19}$$

$$\widetilde{\mathrm{c_t}} = \tanh(\mathrm{W_c x_t} + \mathrm{U_c h_{t-1}} + \mathrm{b_c}) \tag{20}$$

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \quad (21)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (22)$$

$$h_t = o_t \odot \tanh(c_t) \quad (23)$$

Where (f_t), (i_t), and (o_t) are the forget, input, and output gates, respectively, and $(\odot)$ represents element-wise multiplication. The gated recurrent unit (GRU) simplifies the LSTM architecture by using a reset gate (r_t) and update gate (z_t) defined as (24) – (27).

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (24)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (25)$$

$$\tilde{h_t} = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (26)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t} \quad (27)$$

Which eliminates the need for separate memory cells and reduces computational complexity. The bidirectional LSTM (BiLSTM) extends LSTM by processing input sequences in both forward and backward directions. Given an input sequence $(X = \{x_1, x_2, \dots, x_T\})$, BiLSTM consists of two hidden states as presented as (28) and (29) respectively.

$$h_t^{fwd} = \text{LSTM}(x_t, h_{t-1}^{fwd}) \quad (28)$$

$$h_t^{bwd} = \text{LSTM}(x_t, h_{t+1}^{bwd}) \quad (29)$$

The final representation is obtained by concatenation is (30).

$$h_t = (h_t^{fwd}, h_t^{bwd}) \quad (30)$$

Which provides a richer context representation for sequence classification. The attention-based LSTM model enhances LSTM by applying an attention mechanism. Given the hidden states $(H = \{h_1, h_2, \dots, h_T\})$, attention weights (α_t) are computed as (31) – (33).

$$e_t = v^T \tanh(W_a h_t + b_a) \quad (31)$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t'} \exp(e_{t'})} \quad (32)$$

$$c = \sum_t \alpha_t h_t \quad (33)$$

Where (e_t) is the attention score, (α_t) is the attention weight, and (c) is the weighted sum of hidden states. The attention-enhanced representation is then passed to the final classification layer. Each of these architectures is trained using backpropagation and optimized using the Adam optimizer. The loss function is categorical cross-entropy as presented as (34).

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \sum_{k=1}^{3} y_{ik} \log P(y_{ik} \mid X_i, \theta) \quad (34)$$

Where $(P(y_{ik} \mid X_i, \theta))$ is the predicted probability for class (k) given sample (i), and (y_{ik}) is the true one-hot encoded label. These architectures enable the classification of fetal health conditions by capturing complex spatial and temporal dependencies within the dataset. The comparative analysis of these models provides insights into their effectiveness in fetal health classification.

2.4 Evaluation Metrics

To assess the performance of the deep learning models in classifying fetal health conditions, multiple evaluation metrics are utilized, including accuracy, precision, recall, and F1-score. Given a set of predicted labels $(\hat{y})$ and true labels (y), where each sample belongs to one of three classes, these metrics provide a quantitative measure of classification effectiveness. The accuracy metric quantifies the proportion of correctly classified instances over the total number of samples. Given a dataset of size (N), let (y_i) and $(\hat{y_i})$ represent the true and predicted labels of the (i)-th sample, respectively. The accuracy is defined as (35).

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(y_i = \hat{y_i}) \quad (35)$$

Where $(\mathbb{1}(\cdot))$ is the indicator function, which returns 1 if the condition is true and 0 otherwise. Accuracy provides an overall performance measure but may not be reliable when class distributions are imbalanced. Precision measures the proportion of correctly predicted positive samples among all predicted positive samples for each class. Let (TP_k), (FP_k), and (FN_k) represent the number of true positives, false positives, and false negatives for class (k), respectively. Precision for class (k) is defined as $\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}$ where (TP_k)

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

denotes the number of instances correctly classified as class (k), and (FP_k) denotes the number of instances incorrectly classified as class (k). The weighted precision across all classes is given by (36).

$$\text{Precision} = \sum_{k=1}^{C} w_k \cdot \text{Precision}_k \tag{36}$$

Where (C) represents the number of classes and (w_k) is the proportion of class (k) in the dataset. Recall, also known as sensitivity, measures the proportion of actual positive samples correctly identified by the model. It is defined for class (k) as (37).

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \tag{37}$$

Where (FN_k) denotes the number of instances belonging to class (k) but incorrectly classified as another class. Similar to precision, the weighted recall across all classes is computed as (38).

$$\text{Recall} = \sum_{k=1}^{C} w_k \cdot \text{Recall}_k \tag{38}$$

The F1-score is the harmonic mean of precision and recall, providing a balanced measure when there is an imbalance between false positives and false negatives. The F1-score for class (k) is given by (39).

$$\text{F1}_k = 2 \cdot \frac{\text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{39}$$

The overall F1-score is computed as $\text{F1} = \sum_{k=1}^{C} w_k \cdot \text{F1}_k$. In the case of multi-class classification, these metrics can be computed using different averaging techniques. The macro-average computes the metric independently for each class and then averages them as presented as (40).

$$\text{Macro-F1} = \frac{1}{C}\sum_{k=1}^{C} \text{F1}_k \tag{40}$$

whereas the weighted-average considers the class distribution as presented as (41).

$$\text{Weighted-F1} = \sum_{k=1}^{C} w_k \cdot \text{F1}_k \tag{41}$$

These evaluation metrics provide a robust assessment of model performance, ensuring that both class-wise and overall classification effectiveness are considered. Given the imbalanced nature of the fetal health dataset, the F1-score and weighted precision-recall metrics are particularly crucial in evaluating the performance of the models.

2.5 Experimental Setup

The experimental setup consists of multiple steps, including dataset partitioning, model training, hyperparameter tuning, and evaluation. The dataset, denoted as ($\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{N}$), consists of (N) samples, where ($X_i \in R^d$) represents the feature vector of dimension (d), and (y_i) is the corresponding class label. To ensure unbiased model evaluation, the dataset is split into training and testing subsets. The training set and testing set satisfy the condition as presented as (42).

$$\mathcal{D}_{train} \cup \mathcal{D}_{test} = \mathcal{D}, \quad \mathcal{D}_{train} \cap \mathcal{D}_{test} = \{\} \tag{42}$$

Where ($|\mathcal{D}_{train}| = 0.8N$) and ($|\mathcal{D}_{test}| = 0.2N$), meaning that (80%) of the dataset is allocated for training, while (20%) is reserved for testing. To further improve model generalization and mitigate overfitting, a stratified (k)-fold cross-validation approach is applied to the training set. Given (k) partitions of ($\mathcal{D}_{train}$), the model is trained on ($k-1$) folds and validated on the remaining fold. For a given fold (j), the training and validation sets satisfy (43).

$$\mathcal{D}_{train}^{(j)} \cup \mathcal{D}_{val}^{(j)} = \mathcal{D}_{train}, \quad \mathcal{D}_{train}^{(j)} \cap \mathcal{D}_{val}^{(j)} = \{\} \tag{43}$$

Where ($\left|\mathcal{D}_{val}^{(j)}\right| = \frac{|\mathcal{D}_{train}|}{k}$). The final performance metrics are computed as the average across all folds. The deep learning models are trained using the Adam optimization algorithm, where the update rule for the model parameters (θ) at each iteration (t) is given by (44) – (47).

$$\mathrm{m_t} = \beta_1 \mathrm{m_{t-1}} + (1-\beta_1)\nabla_{\theta\mathcal{L}}(\theta) \tag{44}$$

$$\mathrm{v_t} = \beta_2 \mathrm{v_{t-1}} + (1-\beta_2)\left(\nabla_{\theta\mathcal{L}}(\theta)\right)^2 \tag{45}$$

$$\widehat{\mathrm{m_t}} = \frac{\mathrm{m_t}}{1-\beta_1^{\mathrm{t}}}, \quad \widehat{\mathrm{v_t}} = \frac{\mathrm{v_t}}{1-\beta_2^{\mathrm{t}}} \tag{46}$$

$$\theta_{\mathrm{t}} = \theta_{\mathrm{t-1}} - \eta \frac{\widehat{\mathrm{m_t}}}{\sqrt{\widehat{\mathrm{v_t}}}+\epsilon} \tag{47}$$

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

Where (m_t) and (v_t) are the first and second moment estimates, (β_1) and (β_2) are decay rates, (η) is the learning rate, and (ϵ) is a small constant for numerical stability. The loss function used to optimize the model is categorical cross-entropy, defined as (48).

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N}\sum_{k=1}^{C} y_{ik} \log P(y_{ik} \mid X_i, \theta) \quad (48)$$

Where (C) is the number of classes, (y_{ik}) is the one-hot encoded label for class (k), and $(P(y_{ik} \mid X_i, \theta))$ is the predicted probability for class (k). During training, a mini-batch approach is employed, where the dataset is divided into batches of size (B). The gradients of the loss function with respect to the model parameters are computed over each batch (49).

$$\theta \leftarrow \theta - \eta \frac{1}{B}\sum_{i=1}^{B} \nabla_{\theta}\mathcal{L}(X_i, y_i, \theta) \quad (49)$$

Where (B) is set to 32 for computational efficiency. The experiments are conducted on a hardware environment equipped with an NVIDIA Tesla T4 GPU, utilizing TensorFlow and Keras as the deep learning frameworks. Each model is trained for 30 epochs with early stopping criteria based on the validation loss to prevent overfitting. The best-performing model checkpoint is saved for evaluation on the test set.

The final evaluation is conducted on $(\mathcal{D}_{ts})$, using the trained model (f_{θ^*}), where (θ^*) represents the best parameters obtained from training. The predicted class $(\hat{y})$ for a given test sample (X_i) is computed as (50).

$$\hat{y} = \arg\max_k P(y_k \mid X_i, \theta^*) \quad (50)$$

3. RESULT AND DISCUSSION

3.1 Results

The evaluation of the proposed deep learning models for fetal health classification is conducted using four key performance metrics: accuracy, precision, recall, and F1-score. The results obtained from the models are summarized in Table 1. The results indicate that the Convolutional Neural Network (CNN) outperforms all other models across all evaluation metrics, achieving an accuracy of (97.18%), a precision of (97.36%), a recall of (97.18%), and an F1-score of (97.22%). This superior performance can be attributed to CNN's ability to effectively extract spatial features from the input data, enabling robust pattern recognition. CNN's hierarchical feature extraction mechanism allows it to capture local and global dependencies, making it well-suited for fetal health classification. The Multilayer Perceptron (MLP) achieves competitive performance, with an accuracy of (96.52%), precision of (96.61%), recall of (96.52%), and F1-score of (96.51%). MLP's fully connected layers capture complex relationships between features but lack the ability to exploit spatial and sequential patterns inherent in fetal health data. The high performance of MLP suggests that the dataset exhibits strong discriminative features that can be effectively learned through dense layers.

Table 1. Deep Learning Performance Results

Model	Accuracy	Precision	Recall	F1-Score
MLP	0.9652	0.9661	0.9652	0.9651
CNN	0.9718	0.9736	0.9718	0.9722
LSTM	0.9539	0.9557	0.9539	0.9539
GRU	0.9544	0.9571	0.9544	0.9550
BiLSTM	0.9544	0.9556	0.9544	0.9544
Attention-LSTM	0.9516	0.9515	0.9516	0.9508

Among the recurrent neural networks, the Gated Recurrent Unit (GRU) achieves the highest F1-score of (95.50%), outperforming LSTM, BiLSTM, and Attention-LSTM. The standard LSTM model attains an accuracy of (95.39%) and an F1-score of (95.39%), while BiLSTM achieves similar performance with an F1-score of (95.44%). The bidirectional mechanism in BiLSTM does not lead to significant improvements, indicating that the sequential dependencies in fetal health data may not be complex enough to benefit from bidirectional processing. The Attention-LSTM model, which incorporates an attention mechanism to enhance feature selection, exhibits the lowest performance among all models, with an accuracy of (95.16%) and an F1-score of (95.08%). The marginal improvement achieved by attention mechanisms suggests that the dataset features may not require weighted emphasis on specific time steps, as the LSTM-based architectures already capture temporal relationships sufficiently.

To better understand the statistical significance of the results, the performance of each model is analyzed using standard deviation across cross-validation folds. Given (k)-fold cross-validation results for a model (M), the mean performance (μ_M) and standard deviation (σ_M) of metric (S) are computed as (51) – (52).

$$\mu_M(S) = \frac{1}{k}\sum_{i=1}^{k} S_i \quad (51)$$

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

$$\sigma_M(S) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(S_i - \mu_M(S)\right)^2} \quad (52)$$

Where (S_i) represents the performance of model (M) in the (i)-th fold. This statistical analysis helps determine whether the observed differences in performance are significant or occur due to random variations in training. The CNN model's superior performance across all evaluation metrics suggests that convolutional feature extraction is particularly effective for fetal health classification. The results also highlight that fully connected networks such as MLP can achieve high accuracy, but may not generalize as well as CNNs. The performance of recurrent neural networks indicates that sequential dependencies in the dataset do not require advanced memory mechanisms such as BiLSTM and Attention-LSTM, as GRU performs similarly while being computationally more efficient.

3.2 Time and Space Complexity Analysis

The computational complexity of each model is analyzed based on the number of parameters and operations required for inference. Given an input matrix ($X \in R^{N\times d}$), where (N) is the number of samples and (d) is the feature dimensionality, the computational complexity of each model is assessed. The MLP model consists of dense layers, where the primary computational cost comes from matrix multiplications. Given a layer with (d) input neurons and (h) hidden neurons, the complexity is (53).

$$\mathcal{O}(dh) + \mathcal{O}(h^2) \quad (53)$$

For each dense layer. The total complexity for an MLP with (L) layers is $\mathcal{O}(Ldh + Lh^2)$ where (h) represents the number of hidden neurons per layer. The CNN model applies convolutional operations, where the complexity for a single convolutional layer with kernel size (k), input channels (c), and output channels (m) is (54).

$$\mathcal{O}(Ndckm) \quad (54)$$

Which increases with the number of convolutional layers. CNN models have significantly fewer parameters than fully connected networks, leading to reduced overfitting. Recurrent neural networks (RNNs) such as LSTM, GRU, and BiLSTM involve sequential operations that depend on previous time steps. The complexity for a standard LSTM cell is (55).

$$\mathcal{O}(NTdh) + \mathcal{O}(NTh^2) \quad (55)$$

Where (T) represents the sequence length. The BiLSTM model processes inputs in both forward and backward directions, doubling the complexity as presented as (56).

$$\mathcal{O}(2NTdh) + \mathcal{O}(2NTh^2) \quad (56)$$

The GRU model simplifies LSTM by reducing the number of matrix multiplications per step, resulting in (57).

$$\mathcal{O}(NTdh) + \mathcal{O}(NTh^2) \quad (57)$$

Which is slightly more efficient than LSTM. The Attention-LSTM model introduces an additional attention mechanism with complexity (58).

$$\mathcal{O}(NTdh) + \mathcal{O}(NTh^2) + \mathcal{O}(NT^2h) \quad (58)$$

Where the $(\mathcal{O}(NT^2h))$ term corresponds to computing attention weights across all time steps. This explains the lower efficiency of the Attention-LSTM model.

3.3 Threats to Validity

The validity of this study's findings is subject to several potential threats, which are categorized into internal, external, construct, and statistical validity threats. First of all internal validity, the internal validity concerns the correctness of the experimental design and its influence on the results. One major threat is overfitting due to the deep learning models' high capacity. Although cross-validation was used to mitigate overfitting, hyperparameter tuning could have influenced model performance differently across folds. Another threat arises from data leakage, where information from the test set might inadvertently be used during training. To minimize this, strict data partitioning strategies and cross-validation techniques were applied. Second, external validity, it refers to the generalizability of the findings beyond the dataset used in this study. The dataset used is publicly available and widely used in fetal health classification research, but it may not fully represent real-world fetal monitoring scenarios across different demographics and medical conditions. Additionally, differences in feature distributions between this dataset and other clinical datasets could impact the generalizability of the results. Future research should validate these models on multiple datasets from diverse populations to ensure robustness. Thirdly, construct validity assesses whether the evaluation metrics accurately measure model effectiveness. The study

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

uses accuracy, precision, recall, and F1-score, which are standard classification metrics. However, these metrics do not fully capture the real-world impact of misclassifications. For instance, misclassifying a pathological case as normal is far more critical than other misclassification errors. A potential improvement could involve using cost-sensitive learning or custom loss functions that penalize critical misclassifications more heavily. Lastly, statistical validity is concerned with the reliability and significance of the reported results. The performance metrics are averaged over cross-validation folds, reducing variability. However, statistical significance tests (such as a paired t-test or Wilcoxon signed-rank test) were not conducted to compare models rigorously. Future work should include significance testing to confirm that observed differences in performance are not due to random variations in training.

4. CONCLUSION

This study investigated the application of deep learning models for fetal health classification using a publicly available dataset. Multiple architectures were compared, including multilayer perceptron (MLP), convolutional neural network (CNN), long short-term memory (LSTM), gated recurrent unit (GRU), bidirectional LSTM (BiLSTM), and attention-based LSTM. The evaluation was conducted using four key performance metrics: accuracy, precision, recall, and F1-score. The experimental results demonstrate that the CNN model achieves the highest classification performance, with an accuracy of 97.18%, outperforming all other models. The strong performance of CNN can be attributed to its ability to extract spatial features and hierarchical patterns from the input data. Among recurrent architectures, the GRU model performs competitively, achieving an F1-score of 95.50%, while being computationally more efficient than LSTM-based models. The BiLSTM and Attention-LSTM models do not exhibit significant improvements over standard LSTM, suggesting that bidirectional processing and attention mechanisms do not substantially enhance fetal health classification in this dataset. The findings indicate that sequential dependencies in fetal health data do not require complex memory mechanisms, and simpler models such as GRU can achieve comparable results with lower computational costs. A detailed time and space complexity analysis was conducted to assess the computational efficiency of each model. The CNN model demonstrates a favorable trade-off between accuracy and efficiency, benefiting from parallelizable operations that accelerate training and inference. Recurrent models, particularly BiLSTM and Attention-LSTM, exhibit higher time complexity due to sequential dependencies, making them less efficient for real-time applications. The computational overhead introduced by attention mechanisms further reduces training efficiency without providing significant performance gains. The study also identified several potential threats to validity, including overfitting, dataset bias, metric limitations, and statistical uncertainty. To mitigate these threats, a rigorous 10-fold cross-validation strategy was employed, ensuring that the reported performance metrics were generalized. However, the dataset used in this study may not fully represent all fetal health conditions encountered in diverse clinical settings, and future studies should evaluate these models on larger and more diverse datasets. The findings of this study have several important implications for fetal health classification. CNN-based models provide the best overall performance and can be integrated into clinical decision-support systems for automated fetal health monitoring. GRU models offer an efficient alternative to LSTM for applications where computational resources are limited. The study also highlights the importance of optimizing deep learning architectures for medical applications, balancing accuracy, interpretability, and computational efficiency. However, despite the potential for CNN-based models to be deployed in real-time clinical decision-support systems, several challenges must be considered. Real-time deployment in hospitals requires robust infrastructure, regulatory approvals, and validation on diverse datasets to ensure generalizability. Additionally, dataset bias remains a significant issue, as models trained on limited datasets may not perform well across different populations. Another challenge is the interpretability of deep learning models, which is critical in medical applications to gain trust from clinicians. Future research should explore hybrid architectures that combine CNNs and recurrent networks to leverage both spatial and sequential feature extraction. Additionally, the integration of explainable AI techniques such as SHAP (Shapley Additive Explanations) and Grad-CAM (Gradient-weighted Class Activation Mapping) can enhance the interpretability of model predictions, increasing trust and adoption in clinical practice. Investigating cost-sensitive learning methods could also improve model robustness by minimizing the risk of misclassifying critical cases. Addressing these challenges will be crucial for bridging the gap between research advancements and real-world clinical implementation, ultimately improving fetal health monitoring and early diagnosis of perinatal complications.

REFERENCES

[1] S. Franjić, “Prenatal Care Allows Early Detection of Possible Health Problems,” *J Gynecol. Care Child Wellness Res.*, vol. 1, no. 1, p. 1, 2024.
[2] A. J. Lopa, P. Bose, and A. Ahmed, “Prenatal care, risk assessment, and counseling,” in *The Kidney of the Critically Ill Pregnant Woman*, Elsevier, 2025, pp. 9–22.
[3] K. Inayat, S. Saifullah, T. Nelofer, H. Jadoon, N. Danish, and N. Ali, “Evaluating the Effectiveness of Various Prenatal Screening Methods and Diagnostic Tools for Early Detection of Placenta Accreta and

ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 2, February 2025, Page 521-531
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i2.7056

Their Impact On Maternal and Fetal Outcomes," *Health Aff.*, vol. 12, no. 4, 2024.
[4] D. Mennickent *et al.*, "Machine learning applied in maternal and fetal health: a narrative review focused on pregnancy diseases and complications," *Front. Endocrinol. (Lausanne).*, vol. 14, p. 1130139, 2023.
[5] E. Enabudoso, "Electronic fetal monitoring," *Contemp. Obstet. Gynecol. Dev. Ctries.*, pp. 159–173, 2021.
[6] C. E. Valderrama, N. Ketabi, F. Marzbanrad, P. Rohloff, and G. D. Clifford, "A review of fetal cardiac monitoring, with a focus on low-and middle-income countries," *Physiol. Meas.*, vol. 41, no. 11, p. 11TR01, 2020.
[7] N. Katebijahromi, "Detection of Adverse Events in Pregnancy Using a Low-Cost 1D Doppler Ultrasound Signal," Emory University, 2021.
[8] R. Najjar, "Redefining radiology: a review of artificial intelligence integration in medical imaging," *Diagnostics*, vol. 13, no. 17, p. 2760, 2023.
[9] S. S. Rajest, B. Singh, A. J Obaid, R. Regin, and K. Chinnusamy, "Recent developments in machine and human intelligence," 2023.
[10] A. Mehbodniya *et al.*, "Fetal health classification from cardiotocographic data using machine learning," *Expert Syst.*, vol. 39, no. 6, p. e12899, 2022.
[11] M. M. Islam, M. Rokunojjaman, A. Amin, M. N. Akhtar, and I. H. Sarker, "Diagnosis and classification of fetal health based on CTG data using machine learning techniques," in *International conference on machine intelligence and emerging technologies*, 2022, pp. 3–16.
[12] A. K. Pradhan, J. K. Rout, A. B. Maharana, B. K. Balabantaray, and N. K. Ray, "A machine learning approach for the prediction of fetal health using ctg," in *2021 19th OITS International Conference on Information Technology (OCIT)*, 2021, pp. 239–244.
[13] K. N. R. Sree, G. Jotheeswaran, and D. Chitradevi, "Predicting Fetal Health: A Machine Learning Approach using Random Forest Algorithm," in *2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI)*, 2023, vol. 1, pp. 1–6.
[14] J. Ogasawara *et al.*, "Deep neural network-based classification of cardiotocograms outperformed conventional algorithms," *Sci. Rep.*, vol. 11, no. 1, p. 13367, 2021.
[15] M. Khalid, C. Pluempitiwiriyawej, S. Wangsiripitak, G. Murtaza, and A. A. Abdulkadhem, "The Applications of Deep Learning in ECG Classification for Disease Diagnosis: A Systematic Review and Meta-Data Analysis," *Eng. J.*, vol. 28, no. 8, pp. 45–77, 2024.
[16] M. Liu, Y. Lu, S. Long, J. Bai, and W. Lian, "An attention-based CNN-BiLSTM hybrid neural network enhanced with features of discrete wavelet transformation for fetal acidosis classification," *Expert Syst. Appl.*, vol. 186, p. 115714, 2021.
[17] Y. Deng, Y. Zhang, Z. Zhou, X. Zhang, P. Jiao, and Z. Zhao, "A lightweight fetal distress-assisted diagnosis model based on a cross-channel interactive attention mechanism," *Front. Physiol.*, vol. 14, p. 1090937, 2023.
[18] Z. Zhou, Z. Zhao, X. Zhang, X. Zhang, and P. Jiao, "Improvement of accuracy and resilience in FHR classification via double trend accumulation encoding and attention mechanism," *Biomed. Signal Process. Control*, vol. 85, p. 104929, 2023.
[19] D. Ayres-de Campos, J. Bernardes, A. Garrido, J. Marques-de-Sá, and L. Pereira-Leite, "SisPorto 2.0: a program for automated analysis of cardiotocograms," *J. Matern. Fetal. Med.*, vol. 9, no. 5, pp. 311–318, 2000.
[20] A. Mvd, "Fetal Health Classification Dataset." 2020. https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification