(An Open Accessible, Fully Refereed and Peer Reviewed Journal)

A Framework for Classification of Fetal Health Using Deep Learning

May Tamara Stow

Department of Computer Science and Informatics, Federal University Otuoke, PMB 126, Yenagoa, Bayelsa State, Nigeria
E-mail: stowmt@fuotuoke.edu.ng

Abstract

Research on child and maternal mortality risk factors has identified limited access to quality healthcare services as a significant cause of adverse outcomes. Inadequate prenatal care and skilled birth attendance increase the risk of complications during pregnancy and childbirth. A lack of essential healthcare infrastructure in rural or remote areas further exacerbates this risk. By identifying these access-related risk factors, policymakers and healthcare providers can implement strategies to improve healthcare accessibility and save countless lives. The paper presents a deep learning framework for detecting the three stages of fetal health. The dataset was highly imbalanced but solved by oversampling using the random over sampling technique. The ten most essential features were selected using a random forest classifier, and the results were used to build a convolutional neural network model to classify the stages of fetal health with high accuracy.

Keywords: Fetal health; Mortality rate; Deep learning; Convolutional neural network

1. Introduction

Analyzing the factors that contribute to the risk of child and maternal mortality is crucial in understanding the root causes and developing effective interventions to prevent these tragic outcomes. One key factor that increases the risk of both child and maternal mortality is the limited availability of quality healthcare services. Insufficient prenatal care and skilled attendance during childbirth significantly raise the chances of complications, leading to adverse outcomes. In rural or remote areas, the absence of essential healthcare infrastructure, such as hospitals, clinics, and trained healthcare professionals, further exacerbates the risk [1].

Identifying these access-related risk factors enables policymakers and healthcare providers to implement strategies aimed at improving healthcare accessibility and saving numerous lives.

Socioeconomic factors also play a significant role in child and maternal mortality risks. Poverty, food insecurity, and limited education levels are associated with higher mortality rates among mothers and children. Poverty often hinders access to adequate nutrition, resulting in maternal malnutrition and increased vulnerability to complications. Moreover, limited educational opportunities contribute to a lack of knowledge about proper healthcare practices, family planning, and early childhood care, perpetuating the cycle of mortality risks. Addressing these socioeconomic risk factors requires comprehensive strategies to alleviate poverty, improve education, and ensure access to adequate nutrition and social support systems [2].

Another critical risk factor for child and maternal mortality is the prevalence of infectious diseases, particularly in low-income and developing countries. Diseases such as malaria, pneumonia, diarrhea, and HIV/AIDS pose significant threats to both mothers and children. Pregnant women and young children have weaker immune systems, making them more susceptible to infections and their severe consequences. Insufficient access to preventive measures, vaccinations, and appropriate treatment further escalates the risks. Countries can make substantial progress in reducing child and maternal mortality rates by prioritizing infectious disease control through vaccinations, public health campaigns, and improved healthcare infrastructure [3].

2. Literature Review

In a study, the performance of seven machine learning algorithms was compared to predict child and maternal mortality risk factors using clinical data [4]. Performance metrics such as accuracy, precision, and recall were utilized. The random forest algorithm emerged as the top performer, achieving an impressive accuracy of 99.98%. Despite the imbalanced initial dataset, applying under-sampling and oversampling techniques improved performance across all algorithms.

In another study, researchers focused on the classification of fetal health using different machine learning models and the cardiotocography dataset [5]. They compared algorithms such as random forest, logistic regression, decision tree, support vector classifier, voting classifier, and K-nearest neighbor. The random forest model exhibited the highest accuracy, achieving 97.51%.

(An Open Accessible, Fully Refereed and Peer Reviewed Journal)

In a different investigation, additive logistic regression, k-Nearest Neighbors (k-NN), and random subspace classifiers were employed to analyze CTG data to classify specific fetuses [6]. The Logiboost algorithm outperformed k-NN and random subspace in terms of accuracy, achieving an accuracy rate of 92.16% on both the training and testing sets. Logiboost and random subspace performed well and outperformed k-NN in feature selection.

In a proposal, a technique was presented to improve the accuracy of fetal weight estimation and identify potential risks before delivery [7]. The researchers analyzed a clinical dataset comprising 7875 singleton fetuses and addressed the imbalanced learning problem using the Synthetic Minority Oversampling Technique (SMOTE). The Support Vector Machine (SVM) algorithm was employed for fetal weight classification, while the Deep Belief Network (DBN) was utilized for fetal weight estimation based on ultrasound parameters. The proposed model outperformed commonly used regression formulas, yielding a Mean Absolute Percent Error (MAPE) of 6.09% and a Mean Absolute Error (MAE) of 198.55g.

In a comparative study, Rahmayanti *et al*. assessed fetal health using machine learning algorithms based on heart rate data [8]. They utilized a dataset from the UCI Machine Learning Repository containing data from 2126 pregnant women. Five of the seven algorithms achieved high accuracy, ranging from 89% to 99% across three different scenarios, with the LGBM algorithm consistently providing reliable results in all scenarios.

Another analysis focused on fetal heart rate analysis using cardiograph data and known classifiers [9]. The study employed characteristic measurements of FHR and uterine contractions obtained from cardiograms on CTG. The Naive Bayes approach exhibited promising results, with the survey achieving good performance at 85.50% f-measure, 84.88% assurance, 94.60% accuracy, 85.90% recall, and 94.60% precision under specific conditions.

Using rough neural networks, a study explored cardiotocography classification by employing various data mining algorithms such as recurrent neural networks, decision tables, bagging, nearest neighbors, decision trees, and support vector machines [10]. The Recurrent Neural Network (RNN) algorithm achieved a high accuracy of 92.95%.

In another investigation, machine learning techniques were utilized to classify fetal heart rate [11]. The dataset comprised 2126 instances of fetal health data, and SVM, ELM, ANN, RBFN, and RF algorithms were employed. The Artificial Neural Network (ANN) exhibited the best performance with 99.73% sensitivity and 97.94% specificity.

A study analyzed cardiotocographic data using machine learning algorithms for fetal health classification [12]. Different classification algorithms were compared using a dataset consisting of 21 features. Random Forest outperformed all other algorithms, achieving the highest accuracy of 93%, followed closely by SVM with a 93% accuracy rate.

In a study on tree-based ensemble learning, cardiotocography data was utilized to predict fetal health risks [13]. The dataset comprised 2126 observations, and the Random Forest classifier algorithm achieved an accuracy rate of 93.46%, representing an improvement over previous methods.

The classification of fetal heart rates was investigated using empirical mode decomposition and support vector machines [14]. The datasets were categorized into 'normal' or at risk' based on expert classifications. Cross-validation results revealed an accuracy rate of 86% and a geometric mean of 94.8% for sensitivity and specificity measures.

Another study evaluated the use of contraction-dependent fetal heart rate variability to detect distress in fetuses [15]. The researchers analyzed 100 recordings and employed support vector machines with a genetic algorithm for feature selection. The classification performance improved from 70% to 79% for segments closest to birth.

A research effort focused on detecting periodic changes in fetal heart rate using FHR and UCP recordings the proposed methods aimed to identify acceleration, deceleration, and sinusoidal heart rates. The Random Forest algorithm achieved an accuracy rate of 93% for sinusoidal heart rate classification.

Finally, a review article provided an overview of machine learning techniques in fetal cardiology, emphasizing their potential to enhance image acquisition, quantification, segmentation, and diagnosis of fetal cardiac abnormalities and remodeling during pregnancy.

3. Design Methodology

The design methodology shows the process and steps in building a model for detecting and classifying fetal health. This can be seen in Figure 1.
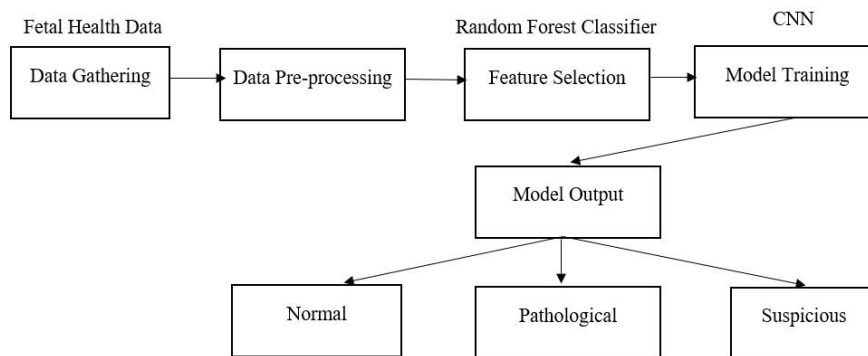

Figure 1. Architectural design of the proposed system.

3.1. Data Gathering

The dataset consists of 2126 entries that include features extracted from Cardiotocogram exams. Three experienced obstetricians evaluated these exams, classifying them into Normal, Suspicious, and Pathological categories. Cardiotocograms (CTGs) provide a convenient and affordable means to assess the well-being of the fetus, enabling

(An Open Accessible, Fully Refereed and Peer Reviewed Journal)

healthcare providers to take necessary measures to prevent infant and maternal mortality. The CTG equipment operates by transmitting ultrasound pulses and analyzing the resulting signals, providing information about Fetal Heart Rate (FHR), fetal movements, uterine contractions, and other relevant factors.

3.2. Data Preprocessing

We carried out a data cleaning to check if there were missing values on the dataset. We also used the MinMaxScaler function for the normalization of the data. The normalization of the data has to do with bringing the data to a unified form. The mathematical expression for data normalization using MinMaxScaler can be seen:

$$normalized\_value = a + (x - x\_min) \times (b - a) / (x\_max - x\_min) \tag{1}$$

In this formula, x represents the original value, x_min is the minimum value in the dataset, x_max is the maximum value in the dataset, and a and b are the desired minimum and maximum values for the normalized range.

3.3. Feature Selection

We utilized a random forest classifier to select the dataset's most essential features. The feature selection involves assessing the importance of each feature in the dataset and selecting the most relevant ones for model training. The process typically consists of evaluating the feature importance scores generated by the random forest algorithm. These scores indicate how much each feature contributes to the model's predictive power.

3.4. Model Training

The Convolutional Neural Network (CNN) model was used for the detection/classification of fetal health. The CNN model was trained using the following:

3.5. Input Features

The input features contain the features of the fetal health records. The records comprise results that were obtained from 2126 patients.

3.6. The Convolutional Layer

Maps the input features into an n x m matrix. This is achieved by using the Word2vec function in Python.

3.7. Maxpooling Layer

This selects the maximum malware features (fetal health features) from the mapped features and converts them into a 3D matrix.

05

(An Open Accessible, Fully Refereed and Peer Reviewed Journal)

3.8. Fully Connected Layer

Here, the vectorization of the maximum selected features is transformed to produce an output that can be normal, suspicious, and pathological.

4. Results

An experiment was carried out utilizing Jupyter Notebook technology to investigate the capabilities of detecting or classifying fetal health. The experiment consisted of two distinct phases. An Exploratory Data Analysis (EDA) was performed in the initial phase of the dataset. A CNN model was trained to detect or classify fetal health in the subsequent phase.

4.1. Exploratory Data Analysis

We performed an Exploratory Data Analysis (EDA) on the dataset to gain better insight. We first plotted a counterplot on the categorical column to determine if there is a minority class on the dataset. This made the dataset imbalanced. The count plot that displays the imbalanced dataset can be seen in Figure 2. The imbalance problem was solved by performing oversampling using the Random Oversampling technique in Python to have a better training result. We plotted a count plot to see if the imbalance problem had been solved. The visualized count plot can be seen in Figure 3. Next, we checked for the most essential features of the dataset. Because the dataset contains 19 features, excluding the categorical column, we employed the random forest classifier to select the ten (10) most important features. The random forest classifier calculated the most essential features by attaching a score to each feature. The ten (10) most important features of the dataset can be seen in Table 1 and Figure 4.
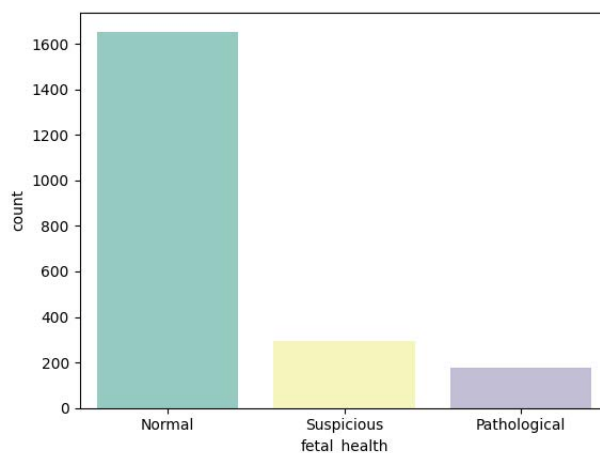

Figure 2. Count Plot of the imbalanced dsata.

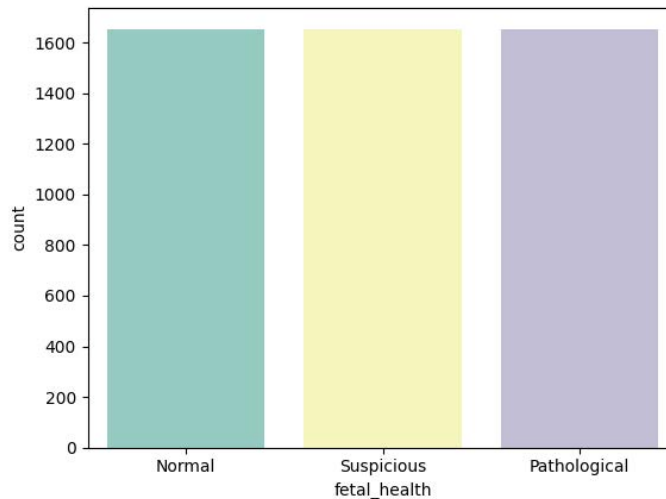**(An Open Accessible, Fully Refereed and Peer Reviewed Journal)**

Figure 3. Count Plot of the balanced data.

Table 1. Ten most important features.

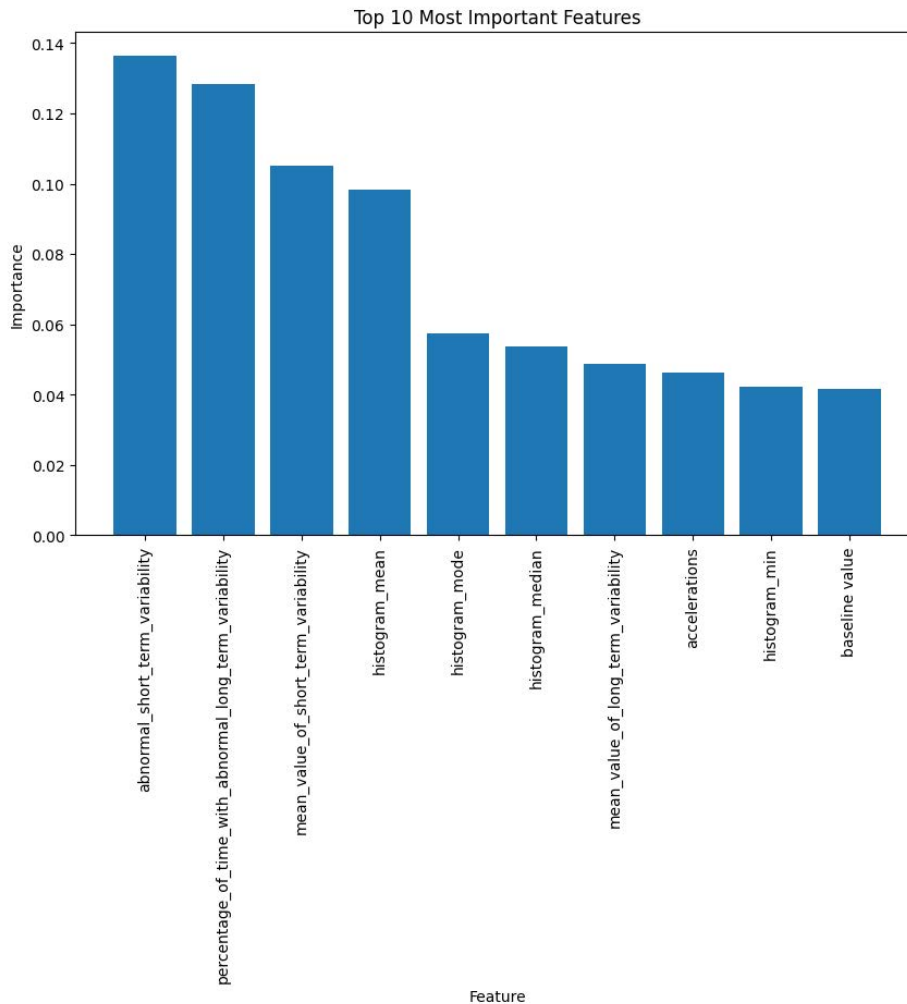| S/No	Features	Important Features
1	Abnormal_short_term_variability	0.136257
2	Percentage_of_time_with_abnormal_long_term_variability	0.128275
3	Mean_value_of_short_term_variability	0.105126
4	Histogram_mean	0.09835
5	Histogram mode	0.057339
6	Histogram_median	0.053573
7	Mean_value_of_long_term_variability	0.048899
8	Accelerations	0.046324
9	Histogram_min	0.042347
10	Baseline value	0.041563

Figure 4. Visualization of the ten most essential features of the dataset.

4.2. Model Training with CNN

The ten most important features were used as input parameters on the CNN model. The convolutional neural network was built using a fully connected layer of three. One input layer, one hidden layer, and an output layer show if the operations carried out on the operating system are malicious or normal. The Convolutional Neural

(An Open Accessible, Fully Refereed and Peer Reviewed Journal)

Network was trained using a batch size of 64, a training epoch of 2,0, and a learning rate of 0.01, a dense layer of 128 for the first layer, 64 for the second layer, and 64 for the third layer, and it's activation function was softmax. The training step for the first ten epochs can be seen in Figure 5. The evaluation matrix regarding training and validation loss can be seen in Figures 6 and 7. The confusion matrix and classification report can be seen in Figure 8 and Figure 9.

```
Epoch 1/50
125/125 [==============================] - 10s 46ms/step - loss: 0.8093 - accuracy: 0.6191 - val_loss: 0.7259 - val_accuracy: 0.6767
Epoch 2/50
125/125 [==============================] - 6s 45ms/step - loss: 0.6891 - accuracy: 0.6984 - val_loss: 0.6368 - val_accuracy: 0.7241
Epoch 3/50
125/125 [==============================] - 5s 41ms/step - loss: 0.5988 - accuracy: 0.7485 - val_loss: 0.5405 - val_accuracy: 0.7724
Epoch 4/50
125/125 [==============================] - 6s 48ms/step - loss: 0.4831 - accuracy: 0.7885 - val_loss: 0.4204 - val_accuracy: 0.8107
Epoch 5/50
125/125 [==============================] - 5s 40ms/step - loss: 0.4025 - accuracy: 0.8233 - val_loss: 0.4170 - val_accuracy: 0.8066
Epoch 6/50
125/125 [==============================] - 5s 44ms/step - loss: 0.3590 - accuracy: 0.8467 - val_loss: 0.3098 - val_accuracy: 0.8661
Epoch 7/50
125/125 [==============================] - 5s 40ms/step - loss: 0.3285 - accuracy: 0.8537 - val_loss: 0.2925 - val_accuracy: 0.8640
Epoch 8/50
125/125 [==============================] - 3s 22ms/step - loss: 0.2892 - accuracy: 0.8731 - val_loss: 0.3284 - val_accuracy: 0.8530
Epoch 9/50
125/125 [==============================] - 4s 28ms/step - loss: 0.2717 - accuracy: 0.8804 - val_loss: 0.2871 - val_accuracy: 0.8751
Epoch 10/50
125/125 [==============================] - 3s 22ms/step - loss: 0.2540 - accuracy: 0.8948 - val_loss: 0.2456 - val_accuracy: 0.9023
```
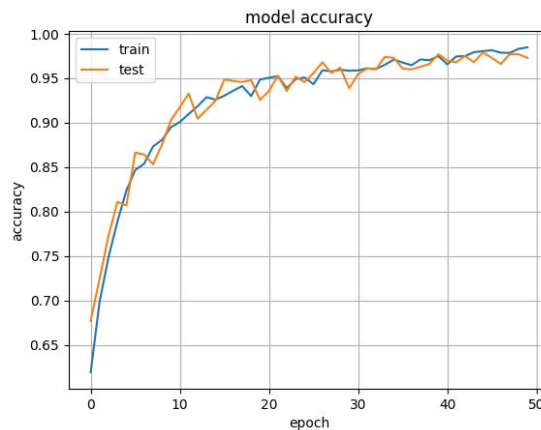
Figure 5. Training steps of the model.

Figure 6. Training and validation accuracy of the model.

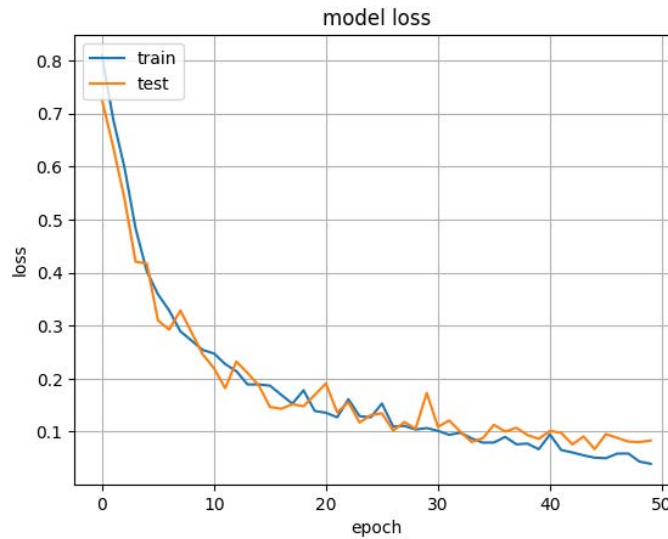**(An Open Accessible, Fully Refereed and Peer Reviewed Journal)**

Figure 7. Training and validation loss.

	precision	recall	f1-score	support
1.0	1.00	1.00	1.00	414
2.0	1.00	1.00	1.00	414
3.0	1.00	1.00	1.00	414
accuracy			1.00	1242
macro avg	1.00	1.00	1.00	1242
weighted avg	1.00	1.00	1.00	1242

Figure 8. Classification report.

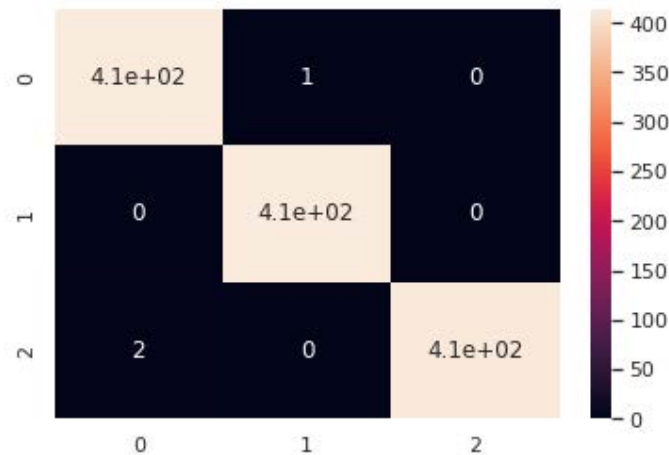**(An Open Accessible, Fully Refereed and Peer Reviewed Journal)**

Figure 9. Confusion matrix of the CNN model.

5. Discussion

Figure 2 shows the count plot of the dataset from the experiment conducted. The count plot indicates that the dataset is highly imbalanced. This is because the class with the highest number of instances is the regular class, and the minority class is that of the pathological class. The count plot in Figure 3 shows that the imbalanced problem of the dataset has been resolved, making them all equal number of instances. Table 1 shows the most essential features of the dataset. This offers the ten most important features of the dataset. This was achieved using a random forest classifier. The random forest classifier shows the most essential features by calculating a score. The scores are used to indicate the ten most important features. The graphical representation of the ten most important features of the dataset can be seen in Figure 4.

Figure 5 shows the training steps of the model for the first ten steps. The training step offers the accuracy and loss for both training and validation tests. It also shows the time spent in completing one training step. Figure 6 shows a line graph showing the accuracy of the training and validation tests. From the graph, the training model achieved a training accuracy of 99.99% for training and 99.98% for validation. Figure 7 shows the model's loss values for training and validation. Both the training and validation tests are below 0.1%. Figure 8 shows the classification report of the model. The classification report shows that the model achieved an approximated value of 100% for precision, recall, f1-score, and accuracy. Figure 9 shows the confusion matrix; from the confusion matrix, the model has a very minimal value for both the expected, suspicious, and pathological classes.

6. Conclusion

Child and maternal mortality risk factor analysis is crucial in identifying the underlying causes and developing effective interventions to reduce these tragic outcomes. This paper presents a deep learning framework for the

11

detection/classification of the three stages of fetal heath. In this paper, we carried out an exploratory data analysis to better understand the dataset. The experimental data analysis results show that the dataset needed more balanced. We solved the data imbalance by performing oversampling using the Random over sampling technique. We selected the ten most essential features using a random forest class for better training results. The random forest classifier was used to choose the ten most important features of the dataset. The result of the selected features was used in building a convolutional neural network model to classify the stages of fatality health. The result of the CNN model shows a better performance for both training and validation, having an accuracy result of 99.99% for training and 99.98% for testing.

7. Acknowledgment

References

[1] Troiano, Nan H., & Patricia M. Witcher. "Maternal mortality and morbidity in the United States: classification, causes, preventability, and critical care obstetric implications." *J Perinat Neonatal Nurs*. 32.3 (2018): 222-231.

[2] Fildes, John, et al. "Trauma: the leading cause of maternal death." *J. Trauma Acute Care Surg.* 32.5 (1992): 643-645.

[3] Kane, Thomas T, et al. "Maternal mortality in Giza, Egypt: magnitude, causes, and prevention." *Stud Fam Plan.* 23.1 (1992): 45-57.

[4] Sheakh, Md Alif, et al. "Child and Maternal Mortality Risk Factor Analysis Using Machine Learning Approaches." *2023 11th Int Symp Digit Forensics Secur (ISDFS)*. IEEE, (2023):1-6.

[5] Alam, Md Takbir, et al. "Comparative analysis of different efficient machine learning methods for fetal health classification." *Appl Bionics Biomech* (2022).

[6] Mehbodniya, Abolfazl, et al. "Fetal health classification from cardiotocographic data using machine learning." *Expert Syst.* 39.6 (2022): e12899.

[7] Feng, Miao, et al. "Fetal weight estimation via ultrasound using machine learning," *IEEE Access 7* (2019): 87783-87791.

(An Open Accessible, Fully Refereed and Peer Reviewed Journal)

[8] Rahmayanti, Nabillah, et al. "Comparison of machine learning algorithms to classify fetal health using cardiotocogram data." *Procedia Comput Sci.* 197 (2022): 162-171.

[9] Afridi, Razman, et al. "Fetal heart rate classification and comparative analysis using cardiotocography data and KNOWN classifiers." *Int J Grid Distrib Comput (IJGDC).* 12 (2019): 31-42.

[10] Amin, Belal, et al. "Classifying Cardiotocography Data based on Rough Neural Network." *Int J Adv Comput Sci App.* 10.8 (2019): 352-356.

[11] Cömert, Z., & A. D. N. A. N. Kocamaz. "Comparison of machine learning techniques for fetal heart rate classification." *Acta Phys Pol A.* 132.3 (2017): 451-454.

[12] Krupa, Niranjana, et al. "Antepartum fetal heart rate feature extraction and classification using empirical mode decomposition and support vector machine." *Biomed Eng Online* 10 (2011): 1-15.

[13] Warmerdam, G. J. J., et al. "Detection rate of fetal distress using contraction-dependent fetal heart rate variability analysis." *Physio Meas.* 39.2 (2018): 025008.

[14] Das, Sahana, et al. "Periodic change detection in fetal heart rate using cardiotocograph." *2020 IEEE 33rd Int Symp Comput-Based Med Syst.* 2020.

[15] Garcia-Canadilla, Patricia, et al. "Machine learning in fetal cardiology: what to expect." *Fetal Diagn The.* 47.5 (2020): 363-372.