## **Anomaly Detection Project Report**

### **1. Introduction**

Cardiotocography (CTG) is a critical technique used during pregnancy to monitor fetal health by measuring fetal heart rate (FHR) and uterine contractions (UC). Anomaly detection in CTG data is crucial for identifying potential risks to the fetus. In this project, we aim to develop an anomaly detection model using an autoencoder trained on CTG data sourced from the UC Irvine Machine Learning Repository with Tensorflow & Keras with Python.

### **2. Dataset Description**

The dataset is taken from open-source platform named UC Irvine Machine Learning Repository [(Dataset Link)](https://archive.ics.uci.edu/dataset/193/cardiotocography). The dataset utilized in this project originates from the UC Irvine Machine Learning Repository, a reputable source for machine learning datasets. With a focus on health and medicine, this dataset comprises 2126 instances of time series data related to Cardiotocography (CTG). It includes 21 features, primarily consisting of measurements of fetal heart rate (FHR) and uterine contractions (UC), which are essential indicators of fetal health. Expert obstetricians have meticulously classified these features, ensuring the reliability and accuracy of the dataset for analysis. The CTGs were also classified by three expert obstetricians and a consensus classification label assigned to each of them. Classification was both with respect to a morphologic pattern (A, B, C. ...) and to a fetal state (N, S, P). Therefore, the dataset can be used either for 10-class or 3-class experiments.

A blue bar graph with numbers

Description automatically generated

Our dataset is an imbalanced dataset. We have considered the 1st and 3rd feature that is 1 as Normal and 3 as pathologic. Normal values as inliers and Pathology as outliers.

**Distribution of each feature without anomalies:**

**A group of graphs showing different types of data

Description automatically generated**

**Distribution of each feature with anomalies:**

**A group of graphs showing the number of data

Description automatically generated with medium confidence**

### **3. Preprocessing Steps**

Before proceeding with model development, several preprocessing steps were undertaken to ensure the quality and suitability of the data. First, missing values were carefully addressed, and fortunately, no instances of missing data were found within the dataset, eliminating the need for imputation techniques. As the dataset did not contain any insignificant features, all available features were retained for further processing. Furthermore, to facilitate model training and testing, the data was appropriately scaled using the StandardScaler to normalize the feature values. Finally, the dataset was split into training, validation, and test sets, with proportions of 70%, 10%, and 20%, respectively, ensuring an adequate distribution of data for model evaluation.

### **4. Model Architecture**

The anomaly detection model employed in this project utilizes an autoencoder architecture, a popular choice for unsupervised learning tasks. The autoencoder consists of two main components: the encoder and the decoder. The encoder compresses the input data into a lower-dimensional representation, also known as the bottleneck layer. In this implementation, the encoder comprises two dense layers with Rectified Linear Unit (ReLU) activation functions, followed by a dropout layer to prevent overfitting. The decoder, on the other hand, aims to reconstruct the input data from the compressed representation produced by the encoder. It consists of two dense layers with ReLU activation functions. The entire model is optimized using the Adam optimizer and trained to minimize the Mean Squared Error (MSE) loss function. This architecture is well-suited for capturing complex patterns and anomalies within the CTG data, allowing for accurate detection of deviations from normal fetal health patterns.

**Loss curve after model training:**

A graph of loss curves

Description automatically generated

### **5. Hyperparameter Settings**

The hyperparameters were carefully selected to optimize the performance of the anomaly detection model. The model was trained for 50 epochs with a batch size of 32, aiming to strike a balance between convergence and computational efficiency. Encoding dimensions of [8, 16, 32] were explored to capture varying levels of data complexity, while dropout rates of [0.2, 0.3, 0.4] were employed to mitigate overfitting. A learning rate of 0.03 was utilized to govern the step size during optimization.

### **6. Evaluation Results**

The model achieved an impressive accuracy of 91.55% in detecting anomalies within the CTG data. The best-performing hyperparameters were determined to be encoding dimensions of 16 and a dropout rate of 0.3. Along with accuracy we also measure the other metrics values that is 0.60 precision score, 0.46 recall value, 0.52 F1-score, and 0.97 ROC AUC score comprehensively, providing a deeper understanding of the model's performance.

A graph of a function

Description automatically generatedA blue and white graph

Description automatically generated

### **7. Visualizations**

Several visualizations were employed to analyze the model's performance and gain insights into anomaly detection. These included the loss curve, illustrating training and validation loss trends over epochs, and the histogram of reconstruction errors, highlighting the distribution of errors. Additionally, scatter plots of anomalies, confusion matrices, and ROC curves provided further visualization of model performance and anomaly detection capabilities.

A graph of data and data index

Description automatically generated with medium confidenceA graph of a blue line

Description automatically generated with medium confidence

A graph of a bar graph

Description automatically generated

### 

### **8. Conclusion**

The autoencoder model demonstrates significant potential in accurately detecting anomalies in CTG data, as evidenced by the high accuracy achieved. However, further analysis is essential to comprehensively interpret evaluation metrics such as precision, recall, F1-score, and ROC AUC score. This will provide deeper insights into the model's efficacy and areas for improvement.

### **9. Recommendations**

To enhance the model's performance and robustness, exploration of additional evaluation metrics is recommended. Investigating the impact of varying thresholds on anomaly detection accuracy can provide valuable insights into model behavior. Additionally, considering ensemble methods or advanced deep learning architectures may further improve anomaly detection capabilities.

### **10. Future Work**

Future work entails incorporating domain knowledge to refine anomaly detection algorithms, ensuring their alignment with clinical practices. Exploring interpretability techniques can help elucidate model decisions, enhancing trust and understanding. Furthermore, investigating the generalizability of the model across diverse CTG datasets will be crucial for broader application and adoption in clinical settings.

|  |
| --- |
| **Contributors:**  Esnapur, Mallikarjun, m\_esnapur |
| Gondalia, Jaykumar Vitthalbhai, j\_gondalia |
| Parekh, Zeel, z\_parekh212512 |
| Puthiyaveettil Ashraf, Ashiq, a\_puthiyaveettilashr |
| Sathiya, Parth Deepakbhai, p\_sathiya |