LSTM based Fetal Distress Classification

Hrithik Vijay Auchar

School of Computer Engineering and Technology

MIT Academy of Engineering, Alandi

hvauchar@mitaoe.ac.in/ +917219086680

ABSTRACT

Clinical medical data, especially during pre-labour and labour, consist of multivariate time series of observations. While potentially containing a wealth of insights, the data is difficult to mine effectively, owing to varying length, irregular sampling and missing data. Recurrent Neural Networks (RNNs), particularly those using Long Short-Term Memory (LSTM) hidden units, are powerful and increasingly popular models for learning from sequence data. They effectively model varying length sequences and capture long range dependencies. This paper presents the study to empirically evaluate the ability of LSTMs to recognize patterns in multivariate time series of clinical measurements during childbirth that mainly consist of two vital parameters FHR and UC. Specifically, this paper considers binary classification for diagnoses, and prior detection of Fetal Distress. A novel Fetal Distress classification algorithm is proposed for continuous Labour monitoring. The proposed solution employs a novel architecture consisting of signal resampling and multiple LSTM recurrent neural networks. First, we establish effectiveness of a simple LSTM network for modelling clinical data. Then demonstrate a straight forward and effective training strategy in which we resample training signal of FHR and UC so as to compact training data and then train the LSTM model with two fully connected layers with the data followed by model evaluation on various parameters. The proposed LSTM architecture accurately diagnoses fetal distress cases with classification accuracy 88.21%, Precision 0.89136, F1 score 0.88118.

Keywords

Cardiotocography (CTG), FHR (Fetal Heart Rate), UC (Uterine Contraction), Down Sampling, LSTM (Long Short-Term Memory)

Chapter 1 INTRODUCTION

In well developed and financially stable country with good economy, modern technology, and access to healthcare services the chances of dying during pregnancy, in childbirth, or in the days and weeks after delivery are very low. In developed countries such as US, UK, Canada women have healthy pregnancies and child births. Of course, there exists still a small risk of maternal death or fetal death, even in developed countries. However, that is not the case everywhere else around the world. Throughout the world, more than 300,000 women die each year from problems that arise during pregnancy and childbirth. Where about 700 women die each year in the US, the World Health Organization (WHO) reports that approximately 830 women die each day worldwide, most of them come from week economical background and generally belong to underdeveloped areas.

There exists several factors contributing to such unwelcomed condition such as age, socioeconomic status, parity and so on but major is the availability of skilled obstetrician and doctors, and one of the major cause for such condition is fetal distress (Murray, 2019). Fetal distress refers to signs before and during childbirth indicating that the fetus is not well. Fetal distress is an uncommon complication of labor. It typically occurs when the fetus does not receive enough oxygen for prolonged period of time. It is oftentimes detected through abnormal pattern in fetal heart rate and uterine contraction by a skilled obstetrician.

Recording Cardiotocography (CTG) during labour is a common practise done by obstetrician for fetal monitoring called as electronic fetal monitor. CTG consist of two vital parameters i.e. fetal heart rate and uterine contraction which is continues time series signal. A well trained obstetrician can detect fetal distress by visually inspecting the CTG waveform.

However, fetal distress occurs intermittently, especially in early stages of the labour. Hence, it is difficult to detect them in a short time window of the CTG waveform. Therefore, continuous monitoring of patients' CTG during childbirth is crucial to detect fetal distress. Our approach is to locally execute the CTG classification algorithm that can categorise between normal and distress waveform of FHR and UC prior to that of complications. This task can be simply solved using various methods and techniques of signal processing and Artificial Intelligence.

There exists number of algorithms to solve this task but since we are dealing with time series signal classification LSTM (Long Short Term Memory) Architecture from RNN (Recurrent Neural Networks) is considered to be the standard and also the most popular approach to deal with time series classification problems. One of the appeals of RNNs is the idea that they are able to connect previous information to the present task. LSTM are a special kind of RNN, capable of learning long-term dependencies. LSTM offers a special memory unit to retain past information and do future computation simultaneously. As soon as the model (in our case LSTM) detects any uncommon pattern in CTG signal it will immediately intimate doctor. This can drastically make a huge impact on the death rate during child birth and also could safeguard many lives that are most often lost due to lack of skilled obstetrician and medical services.

I. The “normal” fetal heart rate

Current international guidelines recommend for the normal fetal heart rate (FHR) baseline different ranges of 110 to 150 beats per minute (bpm) or 110 to 160 bpm. Starting with a precise definition of “normality” and performed a retrospective computerized analysis of electronically recorded FHR tracings. American Congress of Obstetricians and Gynaecologists this is been referred from 2009 Practice Bulletin No. 109: Intrapartum fetal heart rate monitoring: nomenclature, interpretation, and general management principles. Obstetrics and Gynaecology 114:192–202. Accelerations are periodic, transient increases in FHR, defined as an increase in FHR >15 bpm for more than 15 seconds

II. Distress Detection

Decelerations are periodic, transient decreases in FHR, usually associated with uterine contractions. Prolonged decelerations/ bradycardia. A deceleration with a reduction in FHR of greater than 30 bpm that lasts for at least 2 minutes is termed a prolonged deceleration They are caused by a decrease in oxygen transfer to the fetus so can arise as a consequence of a wide variety of disorders including:

- Maternal hypotension
- Umbilical cord compression
- Uterine hypertonia

But the major factor that determines the distress condition is the pH of amniotic fluid of the mother if the pH of the fluid is greater than 7.1 then the CTG signal that was associated to the mother is normal else it is a fetal distress condition. So why cannot it, be detect using pH rather than go for the complex machine learning approach? The pH of amniotic fluid can only be tested after childbirth no prior detections and fluid testing can be done before it.

III. Key Differences

Our experiments show that LSTMs can accurately classify binary time series of clinical measurements during childbirth, a topic not addressed in any prior work. Additionally, while some papers use LSTMs for binary classification such as ECG classification for arrhythmia detection [2], ours is the first to address this problem in the medical context. Moreover, for classifying varying length sequences with fixed length output vectors, this paper is the first, to our knowledge, to demonstrate the efficacy of a data resampling and signal downsampling strategy, achieving both faster training and better generalization.

Chapter 2 MATERIAL AND METHODS

I. Dataset Description

This paper is making use of database from the Czech Technical University (CTU) in Prague and the University Hospital in Brno (UHB), contains 552 cardiotocography (CTG) recordings, which were carefully selected from 9164 recordings collected between 2010 and 2012 at UHB.

The CTG recordings start no more than 90 minutes before actual delivery and each is at most 90 minutes long. Each CTG contains a fetal heart rate (FHR) time series and a uterine contraction (UC) signal, each sampled at 4 Hz.

The priority was to create as homogeneous a set as possible; thus only recordings fulfilling the following criteria were included:

- Singleton pregnancy
- Gestational age >36 weeks
- No a priori known developmental defects
- Duration of stage 2 of labor ≤ 30 minutes
- FHR signal quality (i.e. percentage of the recording during which FHR data were available) > 50% in each 30 minute window
- Available analysis of biochemical parameters of umbilical arterial blood sample (i.e. pH)
- Majority of vaginal deliveries (only 46 cesarean section (CS) deliveries included)
- Additional parameters were collected for all recordings, and are available in the (text) .hea files of the records:
- Maternal data: age; parity; gravidity;
- Delivery data: type of delivery (vaginal; operative vaginal; CS); duration of delivery; meconium stained fluid; type of measurement (i.e. ultrasound or direct scalp electrode);
- Fetal data: sex; birth weight;
- Fetal outcome data: analysis of umbilical artery blood sample (i.e. pH; pCO2; pO2; base excess and computed BDecf); Apgar score; neonatology evaluation (i.e. need for O2; seizures; admission to NICU)
- Expert evaluation of the CTG data "Gold Standard" evaluation based on annotation of the signals by 9 expert obstetricians including variability/confidence for each signal.

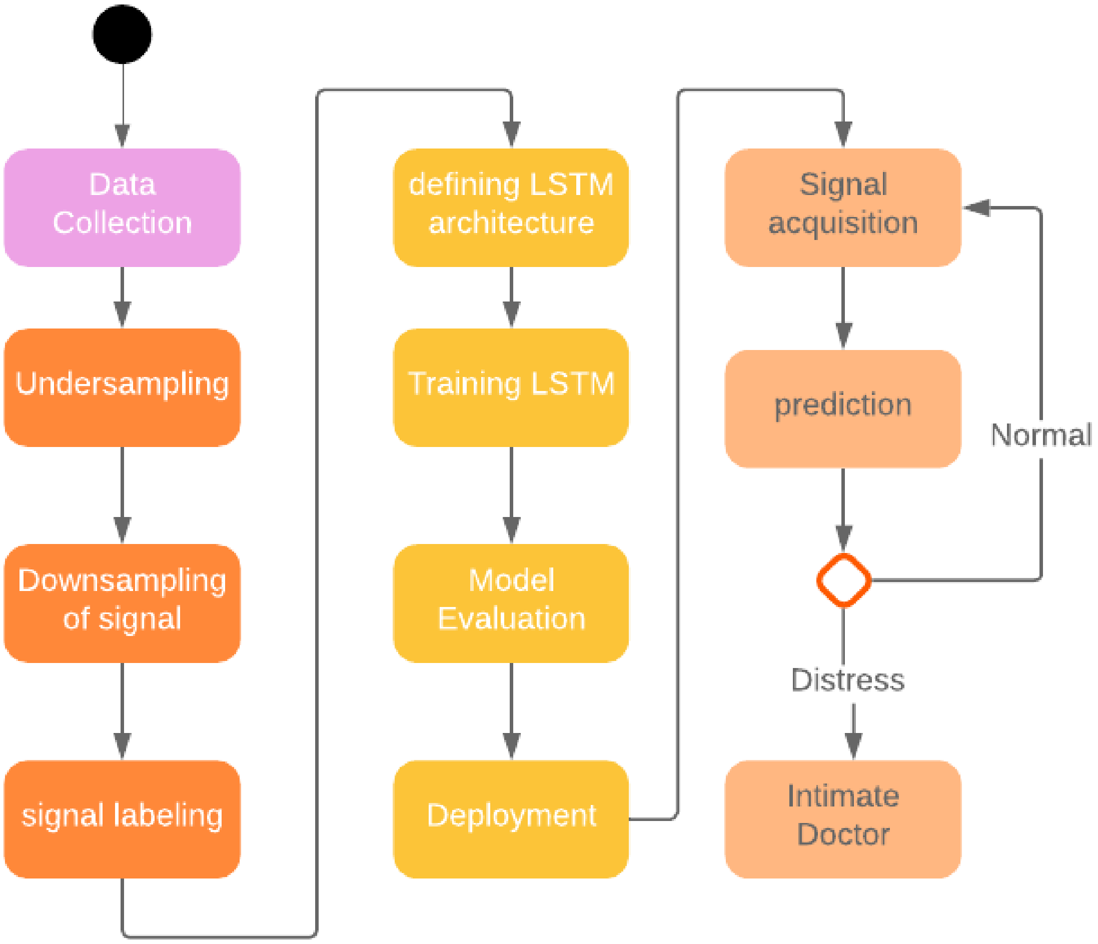

Fig. 1: Workflow

II. The Problem of Imbalance Classes (Badr)

This is the most common problem machine learning engineers typically deal with in a classification task. Data imbalance generally reflects an unequal distribution of classes within a dataset. For example, in a credit card fraud detection dataset, most of the credit card transactions are not fraud and extremely few counts of transactions are fraud transactions. This leaves us with something like 50:1 ratio between the fraud and non-fraud classes. The same problem exists in our case in which normal delivery are in abundance and the fetal distress cases are sparingly encountered. The CTG dataset that we are referring to consist of 496 normal deliveries and 56 fetal distress cases, if binary classifier is trained on this imbalance dataset, the model will be completely biased. And this will also impact on the correlation between features. The referred dataset has 9:1 ratio between normal and distress classes.

There are several methods to overcome this problem but most successful of them is dataset resampling. This method can be applied in two ways

1. Undersampling in which some observations from the majority class are randomly delete in order to match the numbers with the minority class.
2. Oversampling which is a bit complicated process, in which synthetic data is generated that tries to randomly generate a sample of the attributes from observations in the minority class.

But oversampling is extremely risky since we are dealing with medical system it's most important for the training data to be authentic and unaltered, and as discussed oversampling generates observations synthetically. After all appropriate and accurate data will lead us to an optimal and trustworthy solution, but synthetic data has high probability of misleading the classifier and training process. Due to this we go for undersampling of dataset which will retain the originality and essence of the original dataset. Although it has a drawback of reduction in the number of training data points but it's more or less negligible.

III. Signal Downsampling

In digital signal processing, downsampling and decimation are terms related with the process of resampling in a multi-rate digital signal processing system. When downsampling (decimation) is performed on a sequence of samples of a *signal* or other continuous function, it produces an approximation of the sequence that would have been obtained by sampling the signal at a lower rate. This can be clearly illustrated by an example, consider an input digital signal.

x = [1 2 3 4 5 6 7 8 9 10]

y = downsample(x,n) decreases the sample rate of x by keeping the first sample and then every nth sample after the first. If x is a matrix, the downsample treats each column as a separate sequence. The output signal when n=3 i.e. y = downsample(x,3) will be

y = [1, 4, 7, 10]

As mentioned in the dataset description, the given signal are sampled are 4Hz that means we have 4 corresponding observations of that particular signal each second that makes processing computationally expensive and also there are high of redundancy in the signal.

These facts have direct relevance with time and space complexity of the algorithm. Thus downsampling is necessary to make data compact and process computationally cheap.

IV. Understanding LSTM Network

LSTMs are explicitly designed to avoid the problem of long-term dependency. All RNN's have the form of a chain of repeating modules of neural network. In standard recurrent neural networks, this repeating module will have a very simple structure, such as a single *tanh* layer. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way. We cast the problem of phenotyping clinical time series as binary classification. Given a series of observations $\boldsymbol{x}$ (1) , ..., $\boldsymbol{x}$ (T), we learn a classifier to generate hypotheses $\boldsymbol{y}$ˆ of the true labels $\boldsymbol{y}$. Here, t indexes sequence steps, and for any example, T stands for the length of the sequence. Our proposed LSTM RNN uses memory cells with forget gates. As output, we use a fully connected layer atop the highest LSTM layer followed by an element-wise sigmoid activation function, because our problem is binary classification. We use *binary cross entropy* as the loss function at each output.

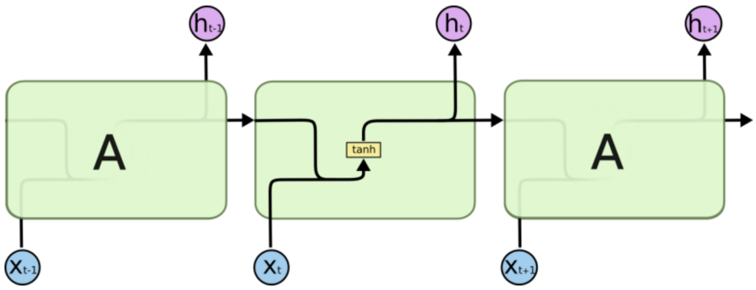

Fig. 2: Stacked LSTM units **(Olah, 2015)**[14]

The following equations give the update for a layer of memory cells ***h(*** lt) where ***h*** ($l-t$)1 stands for the previous layer at the same sequence step (could be a previous LSTM layer or the input layer $\boldsymbol{x}^{(t)}$) and $\boldsymbol{h}_l^{(t-1)}$ stands for the same layer at the previous sequence step:

$$\boldsymbol{g}_l^{(t)} = \phi(W_l^{\mathrm{gx}}\boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{gh}}\boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{g}})$$

$$\boldsymbol{i}_l^{(t)} = \sigma(W_l^{\mathrm{ix}}\boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{ih}}\boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{i}})$$

$$\boldsymbol{f}_l^{(t)} = \sigma(W_l^{\mathrm{fx}}\boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{fh}}\boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{f}})$$

$$\boldsymbol{o}_l^{(t)} = \sigma(W_l^{\mathrm{ox}}\boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{oh}}\boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{o}})$$

$$\boldsymbol{s}_l^{(t)} = \boldsymbol{g}_l^{(t)} \odot \boldsymbol{i}_l^{(i)} + \boldsymbol{s}_l^{(t-1)} \odot \boldsymbol{f}_l^{(t)})$$

$$\boldsymbol{h}_l^{(t)} = \phi(\boldsymbol{s}_l^{(t)}) \odot \boldsymbol{o}_l^{(t)}.$$

In the equations, σ stands for an element-wise application of the *sigmoid* (*logistic*) function φ stands for an element-wise application of the *tanh* function and Θ *is* the Hadamard (element-wise) product. The input, output, and forgets gates are denoted by ***i***, ***o***, and ***f*** respectively, while ***g*** is the input node and has a *tanh* activation. LSTM can be entitled as an extend version of GRUs that are Gated Recurrent Units. A common architecture is composed of a **cell** (the memory part of the LSTM unit) and three major "regulators", usually called gates, of the flow of information inside the LSTM unit: an **input gate**, an **output gate** and a **forget gate**. Some variations of the LSTM unit do not have one or more of these gates or maybe have other gates. For example, as that of gated recurrent units (GRUs) do not have an output gate. A cell can be diagrammatically represented as

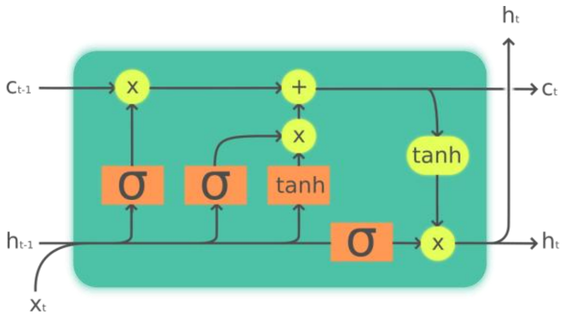

Fig. 3: LSTM Unit **(Chevalier, 2018)**[11]

IV. Different Forms of LSTM Networks

There exist several forms of RNN ranging from one to one, many to one, one to many, many to many to name a few. Each of them has their own significance and application

(1) Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification).

(2) Sequence output (e.g. image captioning takes an image and outputs a sentence of words).

(3) Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment).

(4) Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French).

(5) Synced sequence input and output (e.g. video classification where we wish to label each frame of the video). Notice that in every case are no pre-specified constraints on the lengths sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like.

The red blocks denote the input layer, green the LSTMs cell and blue the output layer.

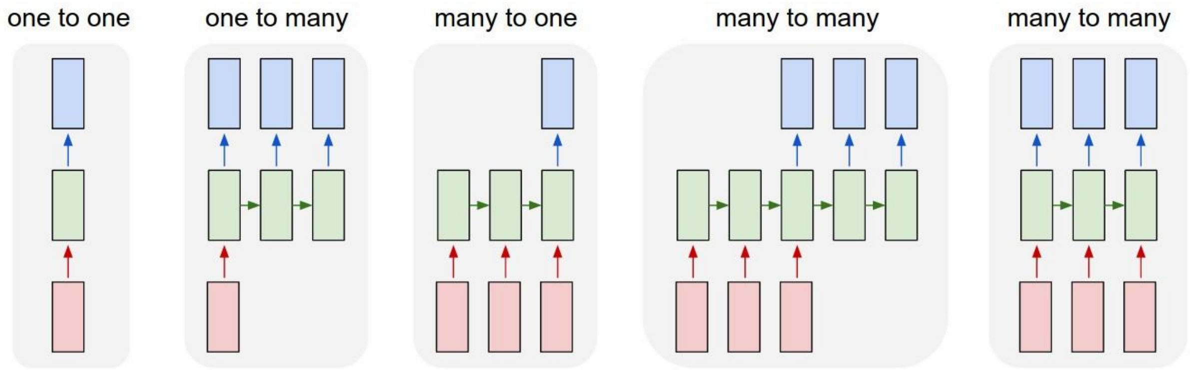

Fig. 4: Types of RNN architecture **(Karpathy, 2015)**[12]

Considering our problem many to many architecture is the most suitable to model solution. The input vector will consist of two elements FHR and UC at that particular instance of time. The digital signal will be sequentially passed to the corresponding LSTM cell at that particular time instance and the LSTM unit will make computation considering the current input and the past values as well. So each unit will make predication so that as soon as we find any abnormality the system will immediately intimate the doctor to take further steps.

V. Training of LSTM

Back propagation (BP) is a well known method for training feed-forward neural networks such as convolutional neural networks (CNNs). This method cannot be applied to RNNs because of the existing temporal dependencies in the model, i.e., the feedback loops in which carry previous information through time. To train RNN models, train data is split into batches of several heartbeats each. The heartbeats are processed sequentially as the following. The weights are updated upon completion of every batch. In the beginning of every batch, h is set to zero and c is set randomly. Then the input data is forward propagated over the network, and error is calculated until the batch finishes. Next, the error is back-propagated over the unfolded network in time, the weight matrices change in all instances and their mean is set as the updated weight.

This is repeated until all batches are processed. This method is known as back propagation through time (BPTT). The optimization method employed in our work is adaptive moment estimation algorithm (Adam).

VI. Model Evaluation

Before the deployment of the model it needs to be tested on unseen data prior to that of real time implementation. By model evaluation we understand how good the algorithm does on data that it has never seen before and it also gives estimation how well will it do after deployment. There is various evaluation parameters on which the model can be evaluate. Each one highlights the significant feature in model that differs from domain to domain.

The output of system is either **fetal distress** (+ve) or **normal** (-ve). There are only 4 cases any lady ***X*** could end up with. We'll be using the following as a reference

- True positive (*TP*): Prediction is +ve and X is fetal distress, *we want that*
- True negative (*TN*): Prediction is -ve and X is normal, *we want that too*
- False positive (*FP*): Prediction is +ve and X is normal, *false alarm, bad*
- False negative (*FN*): Prediction is -ve and X is fetal distress, *the worst*

1. *Classification Accuracy*

   It's the ratio of the correctly labeled subjects to the whole pool of subjects. Accuracy is the most intuitive

   Accuracy = (TP+TN)/(TP+FP+FN+TN)

2. *Precision*

   Precision is the ratio of the **correctly** +ve labeled by our program to all +ve labeled. Count of instance those who we labeled as fetal distress are actually fetal distress.

   Precision = TP/(TP+FP)

3. *Recall*

   Recall is the ratio of the correctly +ve labeled by our program to all who are diabetic in reality. Of all the people who are fetal distress, number of those we correctly predict

   Recall = TP/(TP+FN)

4. *F1 Score*

   F1 Score considers both precision and recall. It is the harmonic mean (average) of the precision and recall.

   F1 Score = 2(Recall * Precision) / (Recall + Precision)*

5. *Specificity*

   Specificity is the correctly -ve labeled by the program to all who are normal in reality. Of all the ladies who are normal, number of those did classifier correctly predicted.

   Specificity = TN/(TN+FP)

Chapter 3 RESULTS AND DISCUSSION

I. Data Analysis

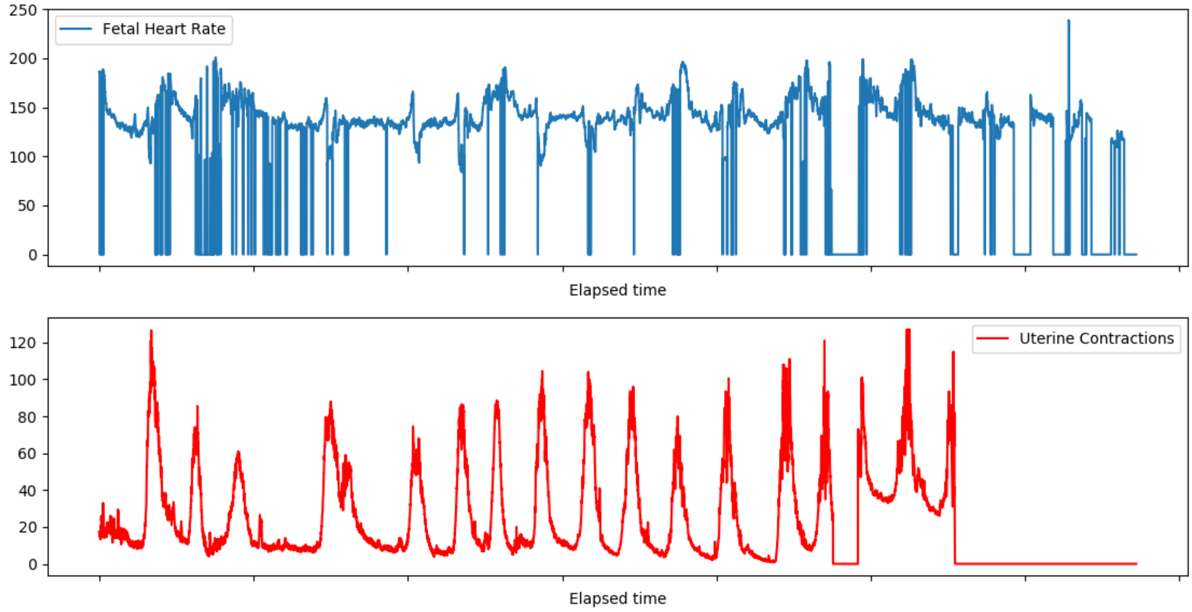

Fig. 5: Input signal

The above plotted is a sample CTG signal out of 552 cases in the training set which is the preliminary input to the system. This signal is sampled at 4Hz i.e. each observation is collected every ¼ second. The signal in the dataset is from the initial stage of labour pain. The labour occurs in 3 stages and last for 3 hours. The dilation of cervix leading to childbirth takes place in 2^{nd} stage of labour in case of normal deliveries. For training purpose referred dataset considers initial phase signal varying in length from 60 minutes to 90 minutes, on an average length of signal is 74 minutes. The time span of labour stages vary from lady to lady due to which it is difficult to fit a traditional ML classifier on it. RNN has the ability to model data varying in length but has constant features as in our case only 2 attributes FHR and UC

Factors	Value (Min)
Maximum	90.09166666666667
Minimum	60.00833333333333
Standard Deviation	7.551343271915915
Variance	57.02278521030975
Average	74.17179951690821

II. Data Pre-processing Strategies

As mentioned under sampling dataset advantageous [1] it made the model extremely well in terms of generalizing the dataset. Underseampling reduced the risk of overfitting to which algorithms are extremely susceptible and helps model to generalise data in better manner. Downsampling also had a great impact on many aspects such as memory, computational complexity and time complexity to name a few parameters [III]. After experimentation we found that downsampling with sampling rate of 4 units is optimum, since if we increase rate then there is high probability of data loss, due to which the LSTM architecture will fail to capture the temporal long term dependencies in CTG signal. And if we reduce sampling rate then there is high chances of data redundancy because extremely minor and unnecessary attributes will also be considered for training. This was also the reason for the poor performance of the model on training data because the original signal was sampled at higher rate. The signal plotted below is UC and only does a comparison between lengths of normal and original and resampled signal.

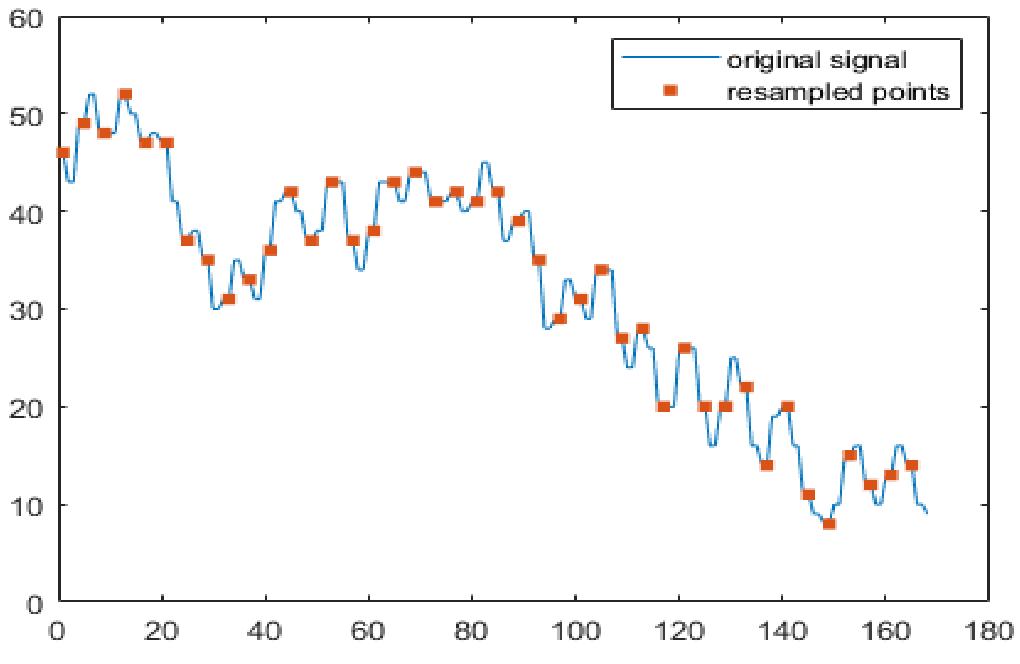

Fig. 5: Resampling

After resampling the signal was reduced to ¼ of original one that means the frequency of sampling signal was reduced from 4Hz to 1Hz. Thus ¾ of the memory space could be saved. Signal labelling is done on basis of the pH of the amniotic fluid if the pH is greater than 7.1 then its labelled as normal else it is considered as fetal distress case.

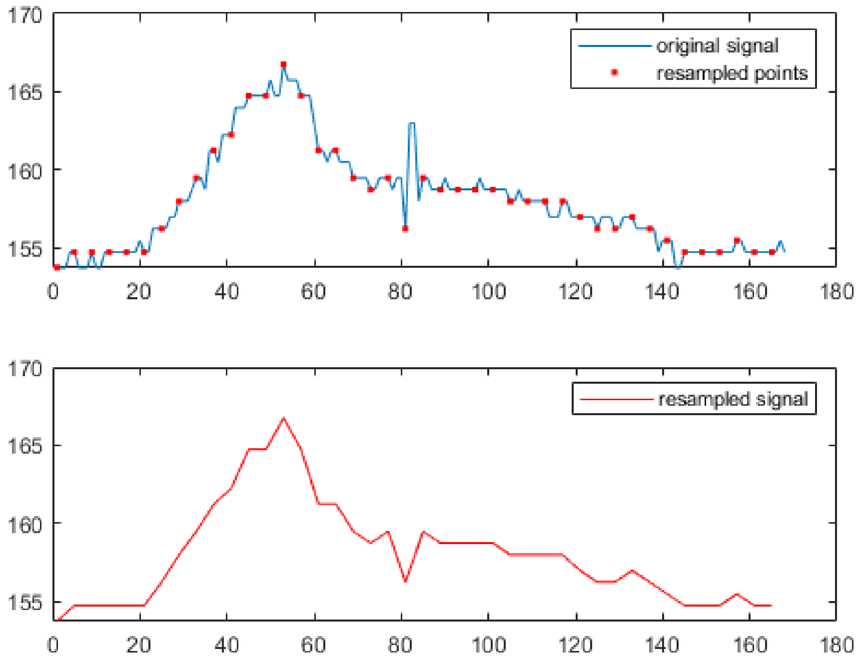

Fig. 6: Original signal Vs Resampled signal

III. Architecture

The below mentioned is the proposed architecture, for this classification task. The first layer is input layer that takes in the value of two CTG parameters FHR and UC which are sequentially inputted from time t=0 to t=N where N is the number of hidden LSTM units. LSTM models with one layers of 350 hidden units followed by 2 fully connected hidden layers and trained with target replication (LSTM-TR) perform best among models using only raw time series as inputs.

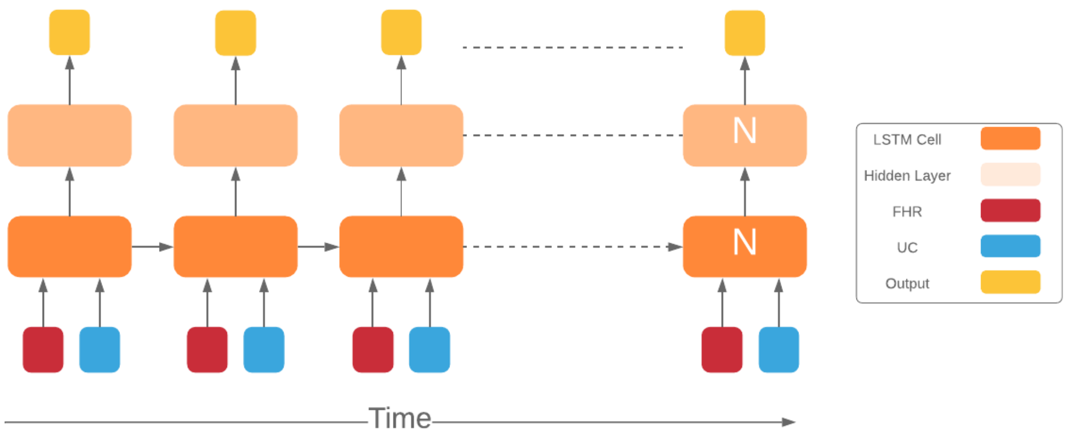

Fig. 7: Implemented Architecture

The proposed architecture improves performance over simple LSTMs on all metrics and accelerates learning. Backpropagating error from outputs at every step, due to target replication provides local targets to guide training, which may aid the many to many architecture in modeling long-term dependencies. Architecture with 350 units followed by 20 fully connected neurons per LSTM unit connected to fully connected layer of 2 neurons and finally stacked softmax layer completes the system. This architecture outperformed all others with highest accuracy of 88.21%.

IV. Network Analysis

Name	**Type**	**Activations**	**Learnables**
sequence input with 2 dimension	Sequence Input	2	-
LSTM with 350 hidden units	LSTM	350	InputWeights 1400x2 RecurrentWeights 1400x350 Bias 1400x1
1st Fully Connected Layer 20 neurons	Fully Connected	20	Weights 20x350 Bias 20x1
2nd Fully Connected Layer 5 neurons	Fully Connected	5	Weights 5x20 Bias 5x1
3rd Fully Connected Layer 2 neurons	Fully Connected	2	Weights 2x5 Bias 2x1
Softmax	Softmax	2	-
Class output Cross Entropy with classes normal and distress	Classification Output	-	-

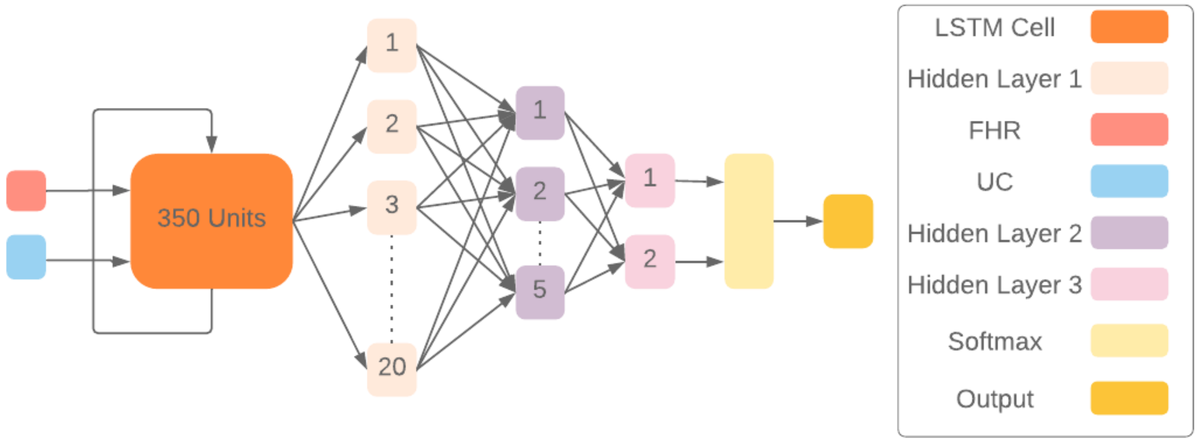

Fig. 8: Network Analysis

V. Training

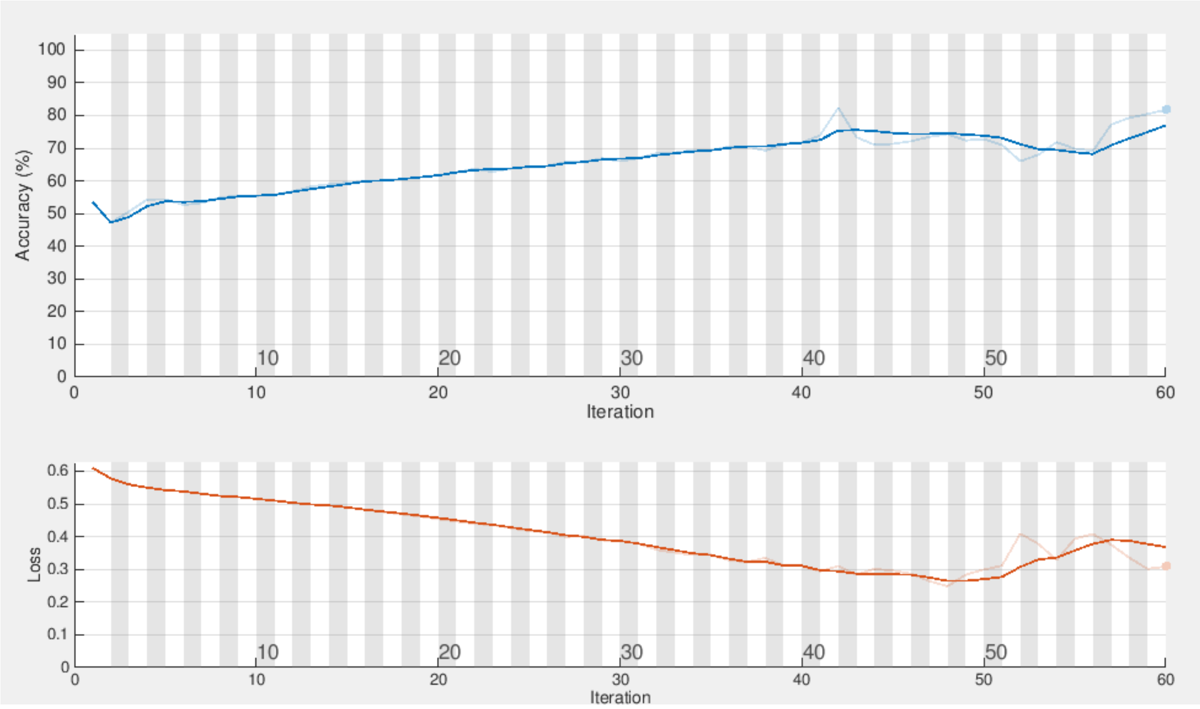

Fig. 9: Traning Progress

When we start off with our LSTM network we initialize our weights randomly. Obviously, it won't give very good results as above training plot of Accuracy Vs Iterations initially performs extremely poor and Loss Vs Iterations initially starts with an extreme high value of loss function. The process of training starts with a bad performing neural network and wind up to network with good accuracy. In terms of loss function, we want our loss function to much lower in the end of training. Improving the network is possible, because we can change its function by adjusting weights. We want to find another function that performs better than the initial one. The problem of training is equivalent to the problem of minimizing the loss function. This is done by *Adam* (short for Adaptive Moment Estimation) that is an iterative method to optimize an objective function especially for neural networks. This algorithm uplifts the power of adaptive learning rates methods to find individual learning rates for each parameter. Which means there exist individual learning rates for each input as well as intermediate inputs to neuron. The randomly initialized weights are then updated using backpropogation algorithm which is about determining how changing the weights impact the overall cost in the neural network. Single Epoch is when an ENTIRE dataset is passed forward and backward through the neural network only ONCE. After several experimentations we found that 60 epochs are sufficient for generalization because if increased then model is likely to overfit.

VI. Model Evaluation Metrices

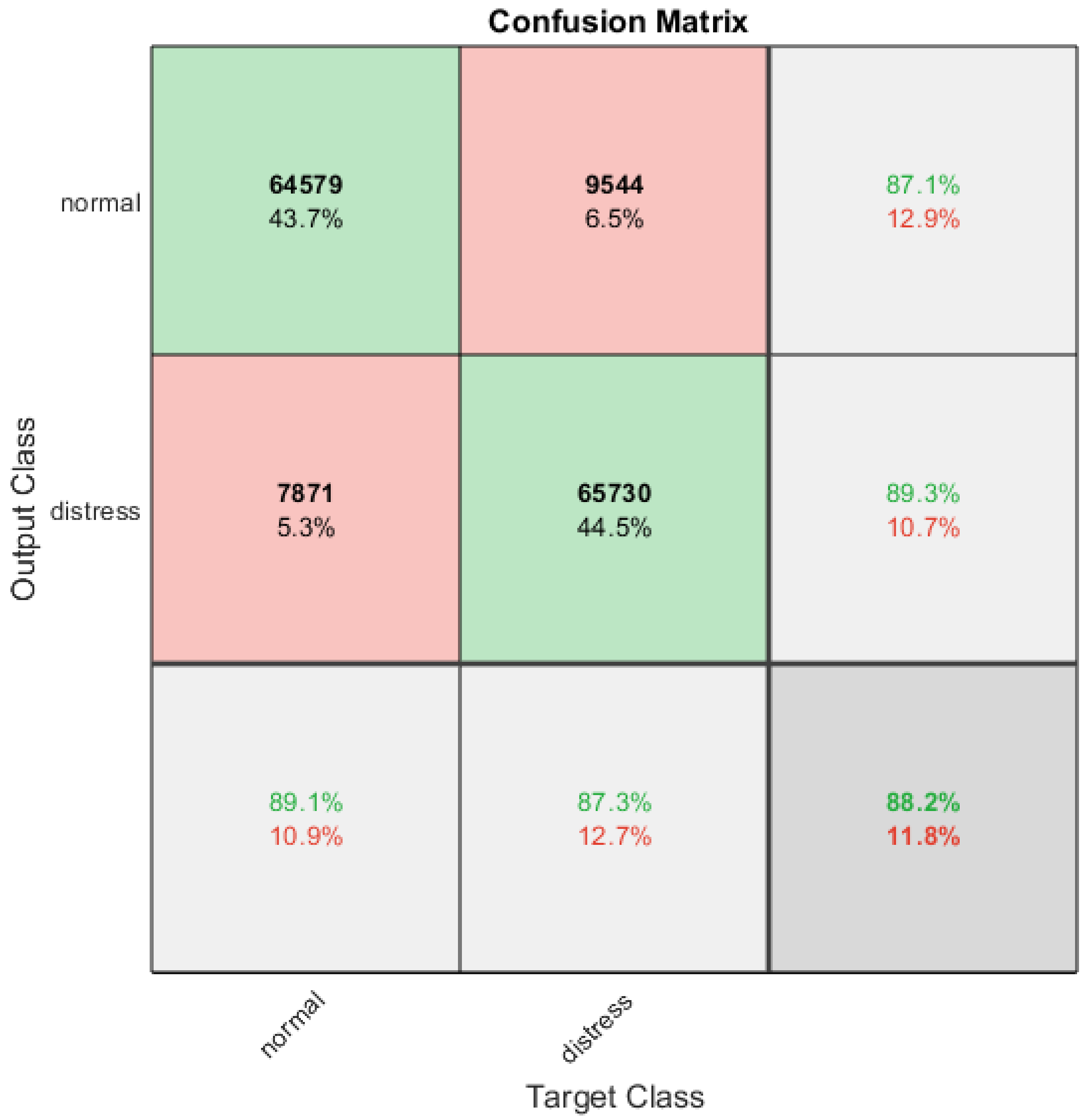

Fig. 8: Confusion Matrix

Performance parameters	Value
Classification accuracy	88.21%.
Precision	0.89136
Recall	0.87124
F1 score	0.88118
Specificity	0.8725

Chapter 4 CONCLUSION

Our results indicate that LSTM RNNs, especially many to many architecture, can successfully classify diagnoses of critical care mothers given clinical time series data before and during childbirth. Our experiments demonstrate a clear advantage over all linear baselines and over traditional feedforward architectures applied to raw time series CTG data.

However, this is only a first step in this line of research in the domain of health care. On the methodological side, we would like to both improve and better exploit the capabilities of LSTMs and also make use of some signal processing strategies. Results from speech recognition have shown that LSTMs shine in comparison to other models using raw features. In the clinical setting, LSTMs may allow us to exploit previously difficult to mine sources of data while minimizing the pre-processing and feature engineering required. In future work, we plan to introduce indicator variables to allow the LSTM to distinguish actual from missing or imputed measurements from CTG Signals.

Additionally the flexibility of the LSTM architecture should enable us to eliminate age-based corrections and to incorporate non-sequential inputs, such as age, weight, and height of the lady, into predictions. We also look forward to consider Fetal outcome data: analysis of umbilical artery blood sample (i.e. pH; pCO2; pO2; base excess and computed BDecf); Apgar score; neonatology evaluation (i.e. need for O2; seizures; admission to NICU) in order to make prediction. We also are encouraged by the success of many to many architecture and plan to explore other variants of this technique and to apply it to other domains and tasks. Additionally, we acknowledge that there remains a debate about the interpretability of neural networks when applied to complex medical problems. We are developing methods to interpret the representations learned by LSTMs in order to better expose patterns of health and severe risks to clinical users. We also hope to make practical use of the distributed representations of patients for tasks including patient similarity search.

REFERENCES

1. Source code is available at https://github.com/hvauchar/LSTM-based-Fetal-Distress-Classification
2. Saeed Saadatnejad, Mohammadhosein Oveisi, and Matin Hashemi, "LSTM-Based ECG Classification for Continuous Monitoring on Personal Wearable Devices" 11 May 2019.
3. J. M. Bote, J. Recas, F. Rincn, D. Atienza, and R. Hermida, "A modular low-complexity ecg delineation algorithm for real-time embedded systems,"*IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 2, pp. 429–441, March 2018.
4. X. Wang *et al.*, "Enabling smart personalized healthcare: A hybrid mobile-cloud approach for ecg telemonitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 739–745, May 2014.
5. J. M. Lillo-Castellano *et al.*, "Symmetrical compression distance for arrhythmia discrimination in cloud-based big-data services," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 4, pp. 1253–1263, July 2015.
6. T. Teijeiro, P. Flix, J. Presedo, and D. Castro, "Heartbeat classification using abstract features from the abductive interpretation of the ecg," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 2, pp. 409–420, March 2018.
7. P. de Chazal, M. ODwyer, and R. B. Reilly, "Automatic classification of heartbeats using ecg morphology and heartbeat interval features,"*IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1196–1206, July 2004.
8. K. Minami, H. Nakajima, and T. Toyoshima, "Real-time discrimination of ventricular tachyarrhythmia with fourier-transform neural network,"*IEEE Transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 179–185, Feb. 1999.
9. M. Lagerholm *et al.*, "Clustering ecg complexes using hermite functions and self-organizing maps," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 7, pp. 838–848, 2000.

10. **Badr, Will.** Having an Imbalanced Dataset? Here Is How You Can Fix It. *towardsdatascience.* [Online] https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb.

11. **Chevalier, Guillaume. 2018.** Long short-term memory. [Online] Wikipedia, May 16, 2018. https://en.wikipedia.org/wiki/Long_short-term_memory#/media/File:The_LSTM_cell.png.

12. **Karpathy, Andrej. 2015.** The Unreasonable Effectiveness of Recurrent Neural Networks. [Online] May 21, 2015. http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

13. **Murray, Donna. 2019.** Maternal Mortality Rate, Causes, and Prevention. *verywellfamily.* [Online] 05 June 2019. https://www.verywellfamily.com/maternal-mortality-rate-causes-and-prevention-4163653.

14. **Olah, Christopher. 2015.** Understanding LSTM Networks. [Online] August 27, 2015. https://colah.github.io/posts/2015-08-Understanding-LSTMs/.