mDLSpiro: Hardware Efficient Deep Learning based Mobile Spirometry

Aakash Agarwal
Dept. of Electronics and Communication
IIIT, Naya Raipur
Chattishgarh, India
aakash19101@iiitnr.edu.in

Debanjan Das
Dept. of Electronics and Communication
IIT, Naya Raipur
Chattishgarh, India
debanjan@iiitnr.edu.in

***Abstract*—Respiratory diseases such as chronic pulmonary disease (COPD), asthma etc., are diseases related to the lungs which usually cause difficulty in breathing. The most common of all the diseases is COPD. Spirometry is the standard diagnostic test for diagnosing COPD. Like every other disease, COPD negatively impacts the sufferer's life, which worsens over time. This requires them to have regular spirometry tests, which are time-consuming or to buy expensive portable devices. Therefore either time or money, both are affected and highly utilized. Since there has been much advancement in the computational power of smartphones, in this work, we try to exploit this and, using built-in microphones, provide a cost-effective alternative to traditional spirometry. This paper proposes a classification system that could classify if the person has COPD or not based on the breathing audio signal. The convolutional neural network is trained and evaluated on the Respiratory Sound Database (ICBHI 2017 database). The model is then compressed using the post-training quantization technique so that it can be deployed on an edge device (smartphone, for instance). A training accuracy of 99.05% and a testing accuracy of 95.56% is achieved. The model size is reduced from an initial 44.822 Megabytes to 11.207 Megabytes which is a four times reduction.**

***Index Terms*—Convolutional Neural Network (CNN), mobile-based spirometry.**

I. Introduction

Asthma and Chronic Obstructive Pulmonary Disease (COPD) are considered as the front-runners in the leading cause of death among other common lung diseases [2]. It is measured that almost 235 million people are asthma sufferers worldwide. Asthma is also considered to be the most common communicable disease among children [3]. World Health Organization (WHO) reports say that more than three million people die due to COPDs, which according to [4] is not just a small number but is 6% of all the deaths that occur worldwide. One thing to note is that over 90 % of COPD deaths are seen in middle-class or lower-class households.

Due to the poor airflow caused by Chronic obstructive pulmonary disease (COPD), it becomes very hard for the sufferer to breathe. Smoking is the main factor that causes COPD. A list of symptoms, not limited to chronic cough, wheezing, and chest tightness, can be seen in a COPD patient, not necessarily all simultaneously. According to [5], we can define COPD as a disease that shows an unusual volume of expiratory/inspiratory flow that does not change substantially when observed for many months. To correctly diagnose this deadly disease, a patient must be very fast in reacting to the symptoms he/she is having. Medical tests, also known as pulmonary function tests (PFT), are required to confirm this diagnosis. One of the ubiquitous pulmonary function tests (PFT) is the spirometry test. Spirometry tests, or in general, pulmonary function tests, are used to check if the lungs are working as they should. This inference is made on the basis of the speed and the amount of air inhaled and exhaled.

Spirometry is the device that is used to perform a spirometry test. The patient is asked to blow inside a tube as hard as possible for them. This tube is then connected to a device. This device measures the amount and the intensity of the air breathed out by the patient. Using this information, doctors make informed decisions. Due to the severity of the disease and the fact that it worsens with time, patients should make a note of changes and progress in symptoms and should make regular spirometry tests to confirm their diagnosis. Sadly, these tests can be done only physically at the medical centres and thus, both time and money both become a significant factor considering traditional spirometry. Regular spirometry tests are time-consuming and, at the same time, costly. As mentioned in [4] 90% of the COPD deaths occurring worldwide are seen in middle-class households. This fact makes it important that the spirometers should be more affordable to the common person, are easy to use and should even be portable. This is the main reason why mobile-based spirometry has acted as the centre of attraction in the past few years in this field.

A mHealth [6] (mobile health) approach can be carried out in contrast to the traditional solution used in spirometry. This project aims to propose smartphone-based spirometry that could record the audio signals and predict if the user is suffering from COPD or not. Our aim is to develop a Deep Learning model that can classify whether the person is suffering from COPD or not and convert the said model into a lightweight counterpart to be deployed on regular mobile phones. This is done by post-training quantization technique using the tensorflow-lite library.

The major contributions of this paper as follows:

- A standalone method for mobile spirometry is proposed that does not require any external equipment.

978-1-6654-6601-1/22/$31.00 ©2022 IEEE

TABLE I
COMPARISON WITH EXISTING WORKS

Paper	Marker	Method	Dataset used	Accuracy	Edge Decisions	Generalized dataset	Computional Power
Sudipto Trivedy, et al.	Differential Pressure	CNN	Self-generated	98.98%	✓	✗	Moderate
Siddharth Gupta et al.	Audio Signal	Sensor Based	Self-generated	95%	✓	✗	Moderate
Fatma Zubaydi, et al.	Audio Signal	Regression	Self-generated	96%	✓	✗	Low
Truc Nguyen, et al	Audio Signal	CNN	ICBHI 2017 [1]	86%	✗	✓	High
Zeenat Tariq, et al.	Audio Signal	CNN	ICBHI 2017 [1]	96%	✗	✓	High
Proposed Method	**Audio Signal**	**CNN**	**ICBHI 2017 [1]**	**95.56%**	✓	✓	**Low**

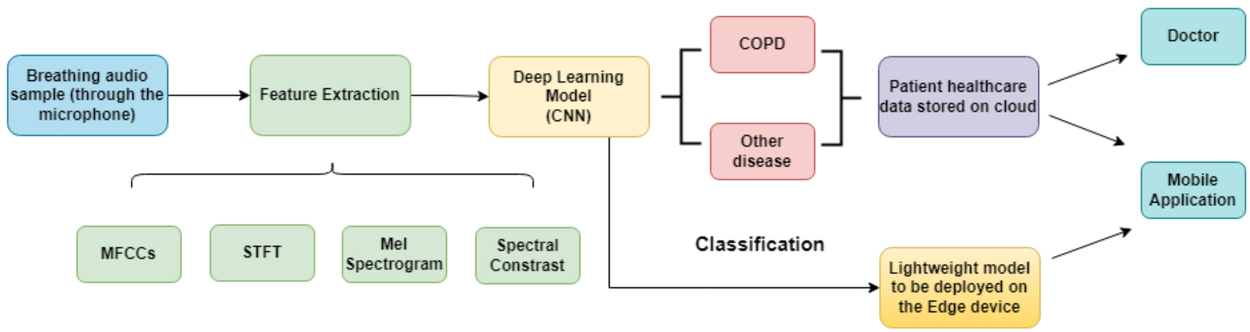

Fig. 1. A flowchart of the vision of the proposed framework

- A robust deep learning model has been designed and tested on the openly available ICBHI 2017 database.
- A method to reduce the computational complexity of the deep learning model is given which helps in making our solution easily deployable on mobile-edge devices for real time prediction

The rest of the paper is organized as follows. Some of the already done works are presented in section 2. The proposed methodology is given in section 3. Result is discussed and presented in section 4. The paper is concluded with future scope in section 5.

II. EXISITING SOLUTIONS

Accurate mobile-based spirometry for asthma was introduced in [7]. This device was designed specially and was able to connect to a smartphone device using an android application so that the patient can monitor their own spirometry readings. This system achieved an accuracy of almost 95%. The results, as well as the performance of this work, were good, but the main drawback is that external hardware, which included a pressure sensor, is used separately. This means that the patients will require to pay extra for the peripheral and must learn how to operate this external hardware setup.

A methodology that consisted of splitting of the breathing signal and then applying voice activity detection (VAD) was proposed in [8]. Features such as the average time duration of the breathing cycle and the energy of the signal are calculated. The accuracy of the work was approximately 86.42 %. This accuracy is achieved on the data produced from the subjects that were manually tested by the authors. In this work, too, no built-in microphone was used.

A sensor-based method using the gas flow based on the sound created by the turbulence is proposed in [9]. Using this method, the breathing cycle phases are detected using a microphone to measure the flow rate then. This work is related to mostly the circuit and the hardware part of the sensor design that they have proposed. Ambient noise and vibrations act as the main reason for this work to be less effective as the accuracy is very much dependent on the amount of noise present in the signal. This means that certain types of filters will also be required to remove the noise from the signal before making predictions.

Sayed Khushal Shah et al. proposed a Deep Convolutional Neural Network approach to classify the ICBHI 2017 database into different classes [10]. They ran the neural network by applying different data augmentation techniques, including, but not limited to, time-stretching, pitch shifting, Dynamic Range Compression. They reported an accuracy of approximately 97%, but they did not propose any method to make their system deployable.

Our work aims to create a standalone system that does not require any external hardware and can give good results. This work is mainly focused on the Deep Learning part of the solution and making that solution capable of being deployed on a mobile-edge device.col

III. PROPOSED MODEL

The proposed system is divided into majorly two phases - The deep learning phase and the model size reduction phase. In the deep learning phase, a model is created which takes in the breathing sound as the input and predicts whether the person has COPD or not, using the Convolutional Neural Networks. First of all, a breathing sample is recorded using the microphone of the mobile. This signal is then preprocessed to remove the surrounding noises. After the sample is ready to use, time-frequency domain features are extracted and fed into the convolutional neural network (CNN) classifier, which predicts whether the user has COPD or not.

In the second phase of our work we, aim to reduce the computational and the storage complexity of the deep learning model to ensure it can be deployed on low powered edge devices, such as mobile phones. After the model is compressed, it will be ready to be deployed on a mobile application.

A. Dataset

Two research teams in Portugal and Greece created a Respiratory Sound Database [1]. A total of 920 recording is present in the dataset. These recordings are of varying lengths, between 10 seconds and 90 seconds. These 920 recordings came from a total of 126 patients and is 5.5 hour long in total. 6898 respiratory cycles are present in all the dataset. Both clean respiratory sound as well as noisy recordings are present in the dataset. The age group of the patients was not at al limited and spanned across all ages – elderly, adults and the children. A detailed information about the dataset is given in table Table

The dataset includes

- 920 .wav sound files
- 920 annotation .txt files
- A text file listing the diagnosis for each patient
- A text file explaining the file naming format
- A text file listing 91 names (filename_differences.txt)
- A text file containing demographic information for each patient

The total 8 classes present in the dataset are mentioned in the table II

TABLE II
DIFFERENT CLASSES PRESENT IN THE DATASET

COPD	URTI
Bronchiectasis	Pneumonia
Bronchiolitis	Asthma
LRTI	Healthy

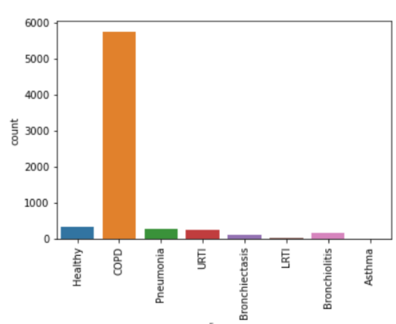

Fig. 2. Count Plot of sample files of different diseases present

The figure clearly shows that the classes are very skewed and the dataset is imbalanced.

B. Feature Extraction

Feature selection is a necessary step in every machine learning related problem [11]. Feature selection techniques have been explored in many studies [12]- [13]. There are three types of features that are calculated when dealing with an audio signal or any signal in general. The three types include the frequency domain, time domain, and time-frequency domain features. The simplest of the three is the Time-domain features which are very easy to extract compared to features extracted from the frequency-domain and time–frequency-domain. Even though time domain features are easy to calculate, they do not contain much information or can be easily hampered by noise. That is why we have used several frequency-domain features. The features considered in this research which are used for the classification of the audio samples, are shown in Table **??**

1) Short-Time Fourier Transform: The short-time Fourier transform is a potent time-frequency domain tool used in audio signal processing. [14]. STFT gives us information about the frequency and the phase of a signal changing with respect to time. For discrete Discrete-time STFT, we have,

$$\mathbf{STFT}\{x[n]\}(m,w) = \mathbf{X}(m,w) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-jwn} \tag{1}$$

2) Mel Frequency Cepstral Coefficient: The mel frequency cepstrum (MFC) Is nothing but a short-term power spectrum representation of an audio signal. This representation is based on the linear cosine transform of a log power spectrum on a non-linear mel scale of frequency. The process of calculating the MFCCs is simple. First of all, the Fourier transform of a windowed signal is calculated. Now the powers of the spectrums that are calculated are mapped onto a mel scale which is a non-linear scale. This is done using overlapping triangular windows. Now logarithm of these powers at different mel frequencies is calculated. Considering these mel-log-powers as signals, discrete cosine transform is them calculated. Finally, the MFCCs are the amplitudes of the resulting spectrum.

3) Mel Spectrogram: A mel spectrogram is a spectrogram that is calculated by converting the frequencies to a mel scale. The mel scale is a non-linear scale, and the mel-scale conversion is defined by the transformation given below.

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2}$$

where f is in hertz and m is in mels.

4) Spectral Contrast: Spectral contrast considers the spectral peak, the spectral valley, and difference in each frequency subband.

C. Model Architecture and Structure

The convolutional neural network (CNN) is made up of the following layers and its structure is given in Table. III

TABLE III
STRUCTURE OF THE PROPOSED CNN

Layer	Output Shape	Parameter
(Conv1D)	(None, 189, 64)	384
(Conv1D)	(None, 185, 128)	41088
(MaxPooling1D)	(None, 92, 128)	0
Conv1D	(None, 88, 256)	164096
Dropout	(None, 88, 256)	0
Flatten	(None, 22528)	0
Dense	(None, 512)	11534848
Dense	(None, 6)	3078

1) Convlutional layer: The operation of convolution is performed in this layer. This layer extracts the hidden features. Convolution operation can be mathematically expressed as,

$$f(x,y)*g(x,y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f(x,y)\times g(x-m,y-n) \quad (3)$$

where $f(x,y)$ and $g(x,y)$ are the input signal and the filter respectively. The weights of these layers are initialized randomly and are updated through the back-propagation learing.

2) Rectified Linear Unit (ReLU): ReLU is the activation function used for the convoluitonal layers. It performs thresholding on each element and makes any value lower than zero, to be zero as given by,

$$f(z) = z \qquad for\ z \geq 0 \quad (4)$$

$$= 0 \qquad otherwise \quad (5)$$

3) Maxpooling layer: This layers returns the maximum value present in a set of values equal to the kernel size.

4) Dropout layer: Dropout is a regularization technique to reduce over-fitting. In a dropout, a fraction k of the nodes given to the layer are made zero at random. This makes the network more robust and reduces over-fitting.

D. Computing Platform

The deep learning model is developed in python 3.6, using the open source library, tensorflow v2.7.0.. The model took 2 minutes and 25 seconds to train on a RYZEN 9 5900HX processor with 16GB of RAM and 8GB RTX 3070 laptop GPU. Other parameters related ot the model training phase are mentioned in the table IV.

TABLE IV
PARAMETERS OF THE CNN MODEL

Parameter	**Value**
Loss Function	Categorical Cross Entropy
Optimizer	Adams' Optimzer
Learning Rate	0.0001
Number of Epochs	80
Activation Function	Rectified Linear Unit (ReLU)

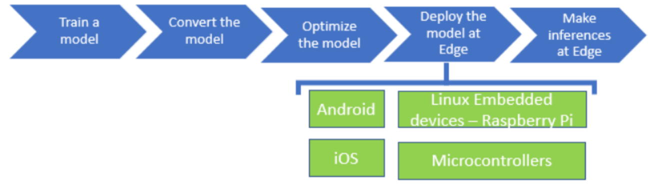

Fig. 3. Flowchart to represent the working steps to convert a heavyweight model to a Tensorflow Lite model [15]

E. Model Compression

The Tensorflow model that we have created is a large model in terms of size. We need a lighter version of this model for the purposes mentioned below.

- Edge devices are low on resources such as storage and computational power. CNN models can be resource-intensive and therefore we need a lightweight model for our edge device.
- We need our system to make quick inferences and reducing the size of the model helps increase the speed and reduce the latency.
- A network requires power for other sensors to work so the power consumed for the inference purpose must be low and hence it is required to have a small size.

There are mainly two techniques that are used to reduce the size of a model.

1) Weight Pruning: Pruning, as it sounds, is the process of removal of parameters of the model that contribute less to the performance of the model. Weight pruning helps in making the model sparse. In this technique, the parameters are ranked according to their importance, and the least ones are removed.

2) Quantization: Quantization is the process of reducing the precision of the numbers that are used for the representation of different parameters in the CNN model. We can apply quantization to both weight and activations. Weights are generally 32-bit floating points that can be downgraded to 16-bit or even 8-bit, which helps in reducing the size of the model. We can even quantize a floating point to an integer.

The above techniques help in reducing the size of the model, but at the same time, this is achieved at a trade-off of accuracy. So one must be very careful while using the said techniques and must set the accuracy limits beforehand. Fig. 3 shows the architecture of the conversion process of a TensorFlow model to a TensorFlow-lite model.

IV. Result and Discussion

A. Phase 1 - Deep Learning

The dataset is split into 80% and 20% fo training and testing purpose respectively.IV shows different parameters and their values used in this study. The model achieves an accuracy of 99.05% on the training data and an accuracy of 95.56% on the testing data. Fig 4 shows the variation of accuracy and the training loss with number of epochs. We can see the training and validation accuracy both increases as the neural network learns better after every epoch. A decreasing trend in the training loss can also be seen.

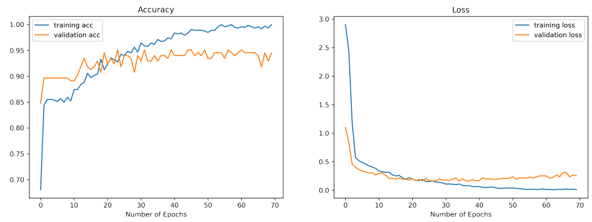

Fig. 4. Variation of model accuracy and loss with number of epochs

Fig. 5 shows the confusion matrix of the classifier that is used to calculate different performance evaluation metrics..

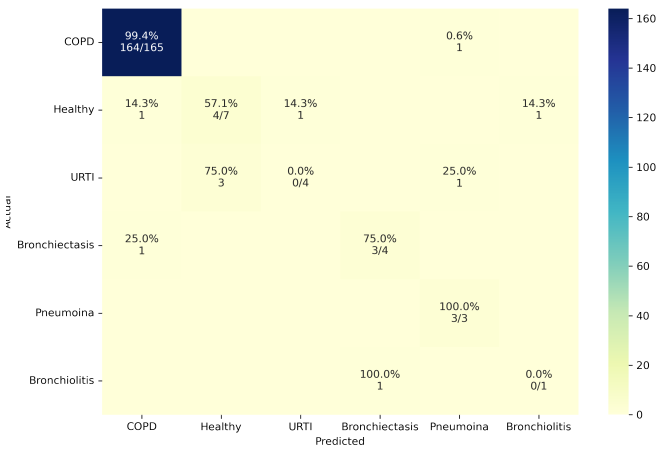

Fig. 5. Confusion Matrix of the proposed deep learning architecture

The performance metrics such as Precision, Recall, F1-Score, and Accuracy [16] are obtained from the Confusion Matrix by the Equations (6)-(9).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{9}$$

where True Positive, False Positive, True Negative and False Negative are denoted by TP, FP, TN and FN.

Precision is defined as the percentage of truly positive among all the predicted positives, whereas Recall is defined as the percentage of predicted positive out of all the positive. F1 score is the harmonic mean of the Precision and Recall.

TABLE V
DIFFERENT CLASSIFICATION METRIC OF THE MODEL

Class	Precision	Recall	F1-score	Support
COPD	0.99	0.99	0.99	165
Healthy	0.57	0.57	0.57	7
URTI	0.33	0.25	0.29	4
Bronchiectasis	1.00	0.75	0.86	4
Pneumonia	0.50	1.00	0.67	3

These metrics are useful in general case but can be used to evaluate the efficiency of the model even on an imbalanced dataset. Clearly from the equation (6)- (7), we can see that the value of precision and recall must be close to 1 for the model to work well. The performance metric values of the model is shown in Table V. The model is trained on different training and testing splits to test the robustness of the system. Considering the class imbalance present in the dataset, we can say that out model performed good even with very less training data. Table VI shows the accuracy at different training and testing spilts (trained for 70 epochs).

TABLE VI
DIFFERENT CLASSIFICATION METRIC OF THE MODEL

Training (%)	Testing (%)	Accuracy (%)
80	20	95.56
70	30	93.39
60	40	92.45
40	60	90.28

B. *Phase 2 - Model Compression*

The model is then converted to a lightweight model, using the tensorflow-lite library. The initial size of the model is 44.822 Megabytes and after the conversion, the model size is reduced to 11.207 Megabytes. We can see a reduction of almost 4 times in the model size.

V. Conclusion and Future Work

A robust system for mobile based spirometry that does not require any external equipment is proposed in this work. A deep learning model is developed based on Convolutional Neural Networks (CNN) that achieves a training accuracy of 99.05% and a testing accuracy of 95.56%. Different evaluation metrics are also calculated for the said model. This model is further compressed by post-quantization technique, which gives us a reduction of approximately equal to 4 times. For future work, we need to connect the front end and the back end of the mobile application to create a full-fledged deployable mobile application. We can also do the computations over a cloud platform rather than doing it on a local machine. This will help in reducing the load on the local machine.

References

[1] N. J. Pithadia and V. D. Nimavat, “ Respiratory Sound Database for the Development of Automated Classification,” *In Precision Medicine Powered by pHealth and Connected Health*, pp. 51–55, 2018.
[2] “The 10 leading causes of death in the world, 2000 and 2012,” May 2014. [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/the-top-10-causes-of-death

[3] "Asthma," November 2013. [Online]. Available: https://www.who.int/respiratory/asthma/
[4] "Chronic Respiratory Diseases," March 2019. [Online]. Available: https://www.who.int/respiratory/en/
[5] "Standards for the diagnosis and care of patients with chronic obstructive pulmonary disease and asthma," *American Thoracic Society Journal*, vol. 136(1), pp. 225–244, 1987.
[6] A. S. Fatma Zubaydi and F. A. , "MobSpiro: Mobile Based Spirometry for Detecting COPD," in *IEEE Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2017, pp. 1–4.
[7] P. R. M. Sudipto Trivedy, "Design and Development of Smartphone-Enabled Spirometer With a Disease Classification System Using Convolutional Neural Network." *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, vol. 69, no. 9, 2020.
[8] A. Abushakra and M. Faezipour, "Lung capacity estimation through acoustic signal of breath," in *Proc. of the IEEE 12th International Conference on Bioinformatics Bioengineering (BIBE)*. IEEE, November 2012, pp. 386–391.
[9] M. F. Ahmad Abushakra, "Lung Capacity Estimation Through Acoustic Signal of Breath." *Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics Bioengineering (BIBE)*, 2018.
[10] Y. L. Zeenat Tariq, Sayed Khushal Shah, "Lung Disease Classification using Deep Convolutional Neural Network," *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019.
[11] F. D. A. B. T. W. Rauber and F. M. Varejáo, "Heterogeneous feature models and feature selection applied to bearing fault diagnosis,," *IEEE Trans. Ind. Electron*, vol. 62, no. 1, pp. 637–646, 2015.
[12] W. J. S. Ding, H. Zhu and C. Su, "'A survey on feature extraction for pattern recognition," *' Artif. Intell. Rev*, vol. 37, no. 3, pp. 169–180, 2012.
[13] N. J. Pithadia and V. D. Nimavat, "A review on feature extraction techniques," *n Proc. WiSPNET*, vol. 1, no. 3, p. 1263–1268.
[14] J. B. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. ASSP, no. 25, pp. 235–238, 1977.
[15] "https://towardsdatascience.com/a-basic-introduction-to-tensorflow-lite-59e480c57292."
[16] J. Brownlee, "How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification," 2020. [Online]. Available: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification