REPUBLIC OF TÜRKİYE
HARRAN UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER OF SCIENCE (MSc) THESIS

LUNG SOUND SIGNAL CLASSIFICATION BY USING COSINE SIMILARITY-BASED MULTILEVEL DISCRETE WAVELET TRANSFORM DECOMPOSITION WITH CNN-LSTM HYBRID MODEL

KHABAT HASAN ABDULLAH

COMPUTER ENGINEERING DEPARTMENT

ŞANLIURFA
2023

CONTENTS

Page No

ÖZET

Yüksek Lisans Tezi

CNN-LSTM HIBRIT MODELIYLE KOSINÜS BENZERLIGINE DAYALI ÇOK DÜZEYLI AYRIK DALGACIK DÖNÜSÜMÜ AYRISIMI KULLANARAK AKCIGER SES SINYALI SINIFLANDIRMASI

Khabat Hasan ABDULLAH

Harran Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Bölümü

Danışman: Dr. Öğr. Üyesi Mehmet Bilal ER
Yıl: 2023, Sayfa: 62

Akciğer ses sinyalleri, solunum yolu rahatsızlıklarının ilk evrelerini teşhis etmek için kullanılabilir. Yapay zeka teknikleri, bu dijital veriler kullanılarak astım, zatürre ve bronşiyolit gibi solunum yolu hastalıklarını veya normal ve anormalleri otomatik olarak belirlemek için kullanılabilir. Bu çalışma, sınıflandırma işlemi için derin bir öğrenme tekniği kullanmaktadır. Uzun kısa süreli belleğe (LSTM) sahip bir Konvolüsyonel Sinir Ağı (CNN) yapısı, derin özellikleri çıkarmak ve sinyalleri normal ve anormal ana kategorilerine ayırmak için kullanılmıştır. Verilerin ön işleme adımlarında Butterworth filtresi adı verilen gürültü giderme yöntemi kullanılarak akciğer sesinden kaynaklanan gürültü giderilmiştir. Özniteliklerin çıkarılması için Kosinüs Benzerliği (CS) tabanlı Çok Düzeyli Ayrık Dalgacık Dönüşümü Ayrışımı (CS-MDWTD) adlı yeni bir yöntem önerilmiştir. DWT, bir sinyali farklı frekans bantlarına ayırabilir, farklı ölçeklerdeki sinyallerin CS'lerini karşılaştırarak, sinyalleri daha kapsamlı bir şekilde anlayabiliriz. Önerilen yöntem Solunum Sesi Veri Seti (ICBHI 2017) üzerinde test edilmiştir. Önerilen CS-MDWTD yöntemi, ICBHI veri seti üzerinde CNN-LSTM hibrit modeli kullanılarak değerlendirilmiştir. Deneysel sonuçlar, akciğer sesi sınıflandırmasının doğruluk oranlarının %80'ın üzerine ulaştığını göstermiştir. Akciğer sinyali ses sınıflandırma tekniklerinin diğer çalışmalarına/kıyaslamalarına kıyasla, sınıflandırma başarısı olasılık açısından yaklaşık olarak (%5) iyileştirilmiştir.

ANAHTAR KELİMELER: Akciğer ses sinyali, ICBHI Veri Seti, derin öğrenme, Butterworth filtresi, CS-MDWTD

ABSTRACT

MSc Thesis

LUNG SOUND SIGNAL CLASSIFICATION BY USING COSINE SIMILARITY-BASED MULTILEVEL DISCRETE WAVELET TRANSFORM DECOMPOSITION WITH CNN-LSTM HYBRID MODEL

Khabat Hasan ABDULLAH

Harran University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assist. Prof. Dr Mehmet Bilal Er
Year: 2023, Page: 62

It is possible to diagnose the early stages of respiratory illnesses by using lung sound signals. By analyzing this digital data, artificial intelligence techniques can be implemented to recognize respiratory ailments automatically such as asthma, pneumonia, bronchiolitis, and normal and abnormal lung sounds. To classify these sounds, a deep learning technique was used in this research to classify lung sound signals into the primary categories of normal and abnormal, a structure consisting of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) is utilized to extract deep features. In the data preprocessing stage, a denoising method known as the Butterworth filter is utilized to remove noise from lung sound signals. For the extraction of the features, a novel-new technique termed Cosine Similarity (CS) based Multilevel Discrete Wavelet Transform Decomposition (CS-MDWTD) is suggested. By comparing the CS of signals at various scales, we can acquire a more thorough knowledge of the signals and the Discrete Wavelet Transform (DWT) can divide a signal into various frequency bands. The proposed technique was evaluated on the Respiratory Sound Database as per the ICBHI 2017 challenge. On the ICBHI dataset, the suggested CS-MDWTD approach is assessed utilizing the CNN-LSTM hybrid model. According to the findings of experimental research, lung sound categorization accuracy rates can reach 80%. Comparing the lung signal sound classification methodologies to other research and benchmarks, the classification success rate has increased by about 5%.

KEYWORDS: Signal of the Lung sound, ICBHI Dataset, deep learning, Butterworth filter, CS-MDWTD

ACKNOWLEDGEMENTS

I would like to begin by expressing my gratitude to Almighty Allah, the ultimate source of energy and blessings, for enabling me to complete this study. Without divine guidance that guided me, this achievement would have been impossible.

I would also like to sincerely thank my project supervisor, **Assist. Prof. Dr Mehmet Bilal Er**, for his invaluable guidance and unwavering support throughout my MSc. project work. His encouragement and feedback were instrumental in the successful completion of this thesis.

I am also deeply indebted to my family for their unfailing love and support over the course of my studies. Their continuous support and encouragement provided me with the strength and determination to overcome any obstacles.

Furthermore, I am grateful to my friends who have been a constant source of support and encouragement, especially my best friend JEHAT, the software developer in Iraq. Their unwavering belief in me and their words of encouragement kept me motivated throughout the project.

Last but not least, I would like to express my sincere thanks to Almighty Allah for granting me the strength and ability to successfully accomplish this project.

LIST OF FIGURES

Page No

LIST OF TABLES

Page No

LIST of SYMBOLS and ABBREVIATIONS

ACC	Accuracy Rate
AE	Auto-Encoder
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AUC	Area Under the Curve
AUC-ROC	Area Under the ROC Curve
BFF	Band Pass Filter
BIGRU	Bi-GRU Bidirectional Gated Recurrent Unit
BIRNN	BI-RNN Bidirectional Recurrent Neural Network
CGAN	Conditional Generative Adversarial Network
CNN-LSTM	Convolutional Neural Network, Long Short-Term Memory of Hybrid Model
COPD	Chronic Obstructive Pulmonary Disease
CWT	Continuous Wavelet Transform
DL	Deep Learning
DS-CNN	Depth-Wise Separable Convolution Neural Network
DWT	Discrete Wavelet Transform
EBT	The Ensemble Bagging Tree (EBT) classifier
ELM	Extreme Learning Machine
ELM-W-AE	Extreme Learning Machine Wavelet Auto-Encoder
EMD	Empirical mode decomposition
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FS	Function Selection
GR	Gated Recurrent
GRU	Gated Recurrent Unit
ICBHI	International Conference on Biomedical and Health Informatics
IMFs	Intrinsic Mode Functions
LDA	Linear Discriminant Analysis
LOOCV	Leave-one-out cross-validation
LRTI	Lower Respiratory Tract Infection
LS	Lung Sound
LTF	Long-term features
MFCC	Mel Frequency Cepstral Coefficient
ML	Machine Learning
NN	Neural Network
RF	random forest.
RNN	Recurrent Neural Networks
ROC	Receiver Operator Characteristic
RSA	Random Subspace Ensembles
RSD	Respiratory Sound Database-Dataset
RSE	Random Subspace Ensembles
SC-MDWTD	Cosine Similarity-based Multilevel Discrete Wavelet Transform Decomposition
STF	Short-term features
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TN	True Negative
TP	True positive
TPR	True Positive Rate
URTI	Upper Respiratory Tract Infection
VTLP	Vocal Tract Length Perturbation
WHO	World Health Organization
WST	Wavelet Scattering Transform structure

1. INTRODUCTION

1.1. Overview

Classifying lung sound signals is a crucial area of research, particularly for identifying abnormal respiratory conditions and preventing lung diseases and obstructive conditions. Lung disease is the third leading cause of morbidity and mortality worldwide, with respiratory disorders causing nearly 3 million deaths annually according to the World Health Organization (WHO). Therefore, analyzing lung sound characteristics and diagnosing respiratory disorders are essential in pulmonary pathology (World Health Organization, 2016).

The classification of lung sounds distinguishes between two categories of lungs: normal and abnormal. A healthy lung signifies the absence and resistance of pulmonary disease, while the presence of abnormal lung sounds is indicative of such illnesses (Chang and Cheng, 2008)(Chang and Lai, 2010). The occurrence of an additional respiratory sound along with the normal lung sound can be considered an abnormal lung sound, which could indicate the presence of pulmonary disease. If the abnormal sound contains wheezing, it is known as a continuous sound, whereas if it contains crackling, it is referred to as a discontinuous sound. Such sounds almost often indicate the existence of lung pathology (Reichert et al., 2008).

A set of mathematical functions known as wavelets can be used to partition data into different components. The discrete wavelet transform (DWT) is a method that splits/ segments time series data points into different frequencies across different scales (Chaovalit et al., 2011). The field of machine learning has a subset called "deep learning" that employs algorithms to imitate intricate data abstractions through model architectures made up of multiple non-linear transformations (Bengio, 2009). Convolutional neural networks (CNNs) excel in manipulating image processing and classification tasks across a wide range of applications. They are specifically designed to effectively analyze and understand visual data, making them an ideal choice for

such tasks (Qayyum et al., 2019). CNNs excel at extracting spatial features from data, while long short-term memory (LSTM) networks are more effective at extracting temporal features (Chollet, 2017).

The specialized kind of neural network architecture designed to process multi-dimensional arrays of data is called Convolutional neural networks (CNNs). They incorporate four key concepts that leverage the characteristics of signals found in nature: shared weights, local connections, pooling, and the use of many layers (deep layering). The typical CNN architecture is composed of a series of stages, with the initial stages consisting of convolutional and pooling layers. The convolutional layers consist of units sorted in feature maps, which are interconnected to local patches in the preceding layer through a set of weights known as a filter bank. The main objective of the convolutional layer is to identify local combinations of features from the previous layer. In contrast, the role of the pooling layer is to combine similar features into a single entity with increased semantic meaning (Lecun et al., 2015).

In this study, several deep learning libraries such as (TensorFlow(tf) and Keras(k)) are used in the model. The important modules and libraries used in the project are Librosa, Pandas, NumPy, seaborn, IPython.display, and import scipy with scipy. signal import butter, lfilter, glob.os, matplotlib. pyplot. And This work is executed on a deep learning server such as Google Collaboratory, Kaggle, And Jupyter Notebook (formerly iPython notebook) interface as a webserver hosted. Many institutions host and configure their Jupyter webservers suitable for Python programming. So, this study used the jupyter notebook (iPython Notebook) as a web server for the project to write live code, Equations, visualizations, and presentations, (which supports Python, Julia, and R). Python programming language and several deep learning libraries, such as TensorFlow and Keras, were employed to create a model for the project using CNN, analyze data, and visualize the accuracy of the results.

1.2. Research problem

The present thesis's big problem is tackling the issue and answering the main questions, for instance, how can lung sound signals be accurately classified? Or how do we extract/select a feature from the lung sound signal? how to remove noise from the lung sound signals based on the dataset used. It is solved by using the Butterworth filter (Bandpass filter) for removing noise, extracting features by CS-MDWTD method and classifying it by using the CNN-LSTM hybrid model for getting enhancement of the patient and normal lung sound. In the present research, the Butterworth filter is utilized to remove the noise present in lung sound signals during the pre-processing stage (Nguyen and Pernkopf, 2020).

1.3. Objectives of the thesis

The objective of this research is to enhance the current work by utilizing deep learning techniques, specifically CNN methods, to select and extract feature, and analyze audio data features for evaluation purposes, removing noise in sound signals in the lung through preprocessing steps, and then classifying lung sounds with make a robust model for getting and integrating of both temporal-spatial information in the classification process by using CNN-LSTM. According to our study, the lung sound signal classified into 2- class, and 3 class based on the ICBHI 2017 data set. And so, the primary goal of this study is to improve the overall accuracy of lung sound classification standards by employing the suggested deep neural network. Another objective is to contribute to the body of knowledge in the field of lung sound analysis by proposing this novel approach for lung sound signal classification, validating its performance on the widely used ICBHI 2017 dataset, and providing insights for further research in this area. Early detection of abnormal lung sounds can be crucial in saving numerous lives.

1.4. Thesis layout

The organization and work of my thesis are going through five chapters:

Chapter 1: This chapter provides the introduction to the Lung Sound Signal Classification by utilizing the CS-MDWTD method with the CNN-LSTM Hybrid model and the objective and organization of the thesis are also described.

Chapter 2: This chapter covers the literature review of some published articles and published books, in this chapter, several types of research will be discussed with taking in mind the author's conclusions about the Lung Sound Signal Classification by using CS-MDWTD with CNN-LSTM Hybrid model.

Chapter 3: This chapter discusses material and method, gives information about the ICBHI 2017 dataset, data preprocessing, feature extraction steps with CS-MDWTD, and Classification steps with CNN-LSTM hybrid model, in general explaining the methods used in this thesis, component description, and the structure of the system and full explanation of the work of the system.

Chapter 4: This part gives the implementation result and discussion.

Chapter 5 presents the conclusion and provides recommendations for future endeavors.

2. LITERATURE REVIEW

The signal categorization and extraction feature for lung sound is more important by using good methods with good models as applied on the big datasets to get the highest accuracy rate. In this context, some important studies (for instance, articles, and books) will be presented in this section.

The authors sought to create a non-invasive diagnostic approach for dividing respiratory sounds into different categories. The researchers utilized two distinct machine learning techniques in their study: the first one employed a convolutional neural network (CNN) with spectrogram images that were generated using short-time Fourier transform (STFT), while the second method utilized a support vector machine (SVM) approach with Mel frequency cepstral coefficient (MFCC) features. The authors then created eight data sets, four for CNN and four for the SVM method, which was used to categorize respiratory sounds. The study found that both the CNN and SVM machine learning algorithms are effective in accurately classifying and predicting respiratory sounds with a large amount of data. Both techniques are effective for classifying spectrogram images. The classification of spectrogram images was found to be more effective using the CNN algorithm compared to the SVM algorithm (Aykanat et al., 2017).

The authors proposed a new CNN model for deep feature extraction, which utilizes pre-trained neural connections. Their approach involves using Linear Discriminant Analysis (LDA) classifier along with the Random Subspace Ensembles (RSE) method and the Deep Feature with CNN and parallel-poling technique. This approach was implemented on the ICBHI 2017 Challenge dataset, which comprises a total of 920 lung sounds. The researchers divided the dataset into 90% for training and validation of the CNN model and the remaining 10% for the LDA classifier and RSE method in the classification stage. The accuracy of the proposed CNN model with the LDA and RSE classifier was 71.15%. The authors conducted their experiments using software named MATLAB (R2019a) on a computer with an Intel (R) Core (TM) i7-

5500 CPU @ 2.4 GHz, an NVIDIA GeForce 840M GPU, and 8 GB of RAM (Demir et al., 2020).

The authors of this study proposed a classification of lung sound system, utilizing a snapshot ensemble of CNNs and evaluated its performance on the ICBHI 2017 database. To enhance the CNN model data, the researchers employed audio pre-processing and feature extraction techniques to extract significant features from log Mel spectrograms. These features were then fed into a robust CNN architecture that incorporated a cosine cycle learning rate schedule. The suggested system's performance was boosted by the snapshot ensemble of CNN models with a modest additional training cost. To address the class imbalance of the dataset, the researchers used temporal stretching/compressing and vocal tract length perturbation (VTLP) for augmentation of data and focal loss to improve model performance (Nguyen and Pernkopf, 2020).

The authors introduced a new attention mechanism as an alternative to RNN, LSTM, and GRU models. Additionally, they proposed a novel data preprocessing method that outperforms previous approaches in dealing with sequence-based problems such as audio and text. The proposed network model combines CNN and attention mechanisms to classify audio data and differentiate between abnormal and normal respiratory sounds, using the ICBHI benchmark dataset. Results show that the proposed attention mechanism outperforms other models such as RNN, BIRNN, GRU, and BIGRU in terms of classification accuracy (Li et al., 2020).

The authors aimed to create an automated classification system for breath sounds that could overcome the limitations of traditional auscultation techniques. They utilized a (CNN) based on approach of deep learning to classify 1918 sounds of respiratory as normal, crackles, wheezes, or rhonchi, which were recorded in a clinical setting. By employing deep learning-based classification, the authors intended to improve the accuracy of clinical auscultation and assist in the speedy diagnosis and proper treatment of respiratory diseases (Kim et al., 2021).

Recently, deep learning has become increasingly popular in various medical fields, such as electroencephalogram and chest X-ray analysis (Tang et al., 2020)(Altan et al., 2021). In a separate study, a deep convolutional neural network (CNN) was employed to classify lung sounds in the International Conference on Biomedical and Health Informatics (ICBHI) database from 2017. The researchers used a time-frequency approach to convert lung sound signals into spectrogram images, but the classification accuracy was only around 65% (Demir and Bajaj, 2020). A subclass of deep neural networks known as a(CNNs), are utilized in search applications for image processing and categorization (Qayyum et al., 2019).

While the LSTM model excels at extracting temporal characteristics, the CNN model excels at extracting spatial features. The LSTM model operates similarly to the RNN(Chollet, 2017). The researchers incorporated gates into the recurrent neurons, including the forget gate, update gate, and output gate, along with the cell, which is the internal processing unit (Pedrycz and Chen, 2020).

The objective was to propose a feature engineering approach to extract specific features for the DS-CNN to efficiently and accurately classify lung sounds into four types. The researchers developed shrunk DS-CNN models based on Mobile Net to keep storage space on edge devices. Three features were extracted for the shrunk model: the MFCC feature, the STFT feature, and a combination of these two features. The researchers discovered that training DS-CNN models using the MFCC or STFT features individually resulted in accuracies of 73.02% and 82.27%, respectively. However, when combining both features, the accuracy increased to 85.74% (Jung et al., 2021).

In their study, the authors utilized pre-trained models to classify lung sounds, incorporating co-tuning and stochastic normalization. They also addressed imbalanced dataset issues through data augmentation. Their model achieved an average score of 92.72±1.30% and 93.77±1.41% for classification of respiratory disease in the 3- and 2-class tasks, respectively(Nguyen and Pernkopf, 2022). The scientists used a conditional generative adversarial network (CGAN) to augment the ICBHI 2017

dataset and other datasets. Their classification model achieved accuracy rates of 92.50% and 92.68% (Jayalakshmy and Sudha, 2021). Gairola et al. proposed the RespireNet model, which uses smart padding and concatenation-based augmentation to enhance performance. They achieved a 2.2% improvement over the previous state-of-the-art results for four-class classification on the ICBHI dataset (Gairola et al., 2021). They put forward an inception-based network and a multi-spectrogram ensemble. They conducted experiments using the ICBHI benchmark dataset and achieved competitive scores of 0.53/0.45 and 0.87/0.85 for respiratory anomaly and disease detection, respectively (Pham et al., 2021).

They introduced a specialized convolutional neural network (CNN) architecture that is lightweight and designed specifically for the detection of respiratory illnesses using individual breath cycles. They utilized a combination of hybrid scalogram-based features derived from lung sounds, incorporating both continuous wavelets transform (CWT) and empirical mode decomposition (EMD). By utilizing a patient-independent train-validation-test set obtained from the ICBHI 2017 lung sound dataset, the authors examined the performance of their methodology. The outcomes revealed impressive weighted accuracy scores of 98.70% for six-class chronic classification and 98.92% for three-class chronic classification (Shuvo et al., 2021).

Khan and Pachori presented an approach to classify lung sounds automatically into non-chronic and chronic categories. They used EMD to extract intrinsic mode functions (IMFs) from the lung sound signals. Their method was evaluated on the ICBHI 2017 dataset, where the Ensemble of Bagged Tree achieved an accuracy of 97.14% (Khan and Pachori, 2021). They employed ensemble classifiers to classify lung sound signals by representing them as features utilizing Shannon, energy of logarithm, and spectrogram-based spectrum entropy. The thesis investigated the application of ensemble techniques, specifically adaptive boosting and bootstrap aggregation. These ensembles utilized discriminant classifiers and decision trees as the foundational base models. Using the boosted decision trees ensemble classifier, they obtained an accuracy rate of 98.27% (Fraiwan et al., 2021)

3. MATERIAL and METHOD

3.1. Dataset

The database of respiratory sounds was first compiled for the scientific challenge held at the International Conference on Biomedical Health Informatics (ICBHI) in 2017. The collection of respiratory sound data consists of 920 annotated audio samples from 126 subjects, containing 5.5 hours of recordings including 6898 respiratory cycles. Among these, 1864 cycles contain crackles, 886 cycles contain wheezes, and 506 cycles contain both crackles and wheezes. The database includes both private and public datasets from the ICBHI 2017 challenge and is freely available for research purposes. Hence, in our study, we utilized the publicly accessible ICBHI 2017 dataset for experimentation by using 126 subjects with 920 audio sample before splitting the data and after that the number of samples changed according the classification distribution as 2 class and 3 class classifications.

3.2. Data preprocessing

Preprocessing is a crucial step in preparing data for extracting feature and categorizations. The researchers utilized a band-pass filter, specifically a Butterworth filter, to eliminate noise in the respiratory sound signals. The audio sample recordings in the dataset were converted to wave format, with different sampling frequencies ranging from 4000 Hz to 41100 Hz. Since the signal of interest falls below 2000 Hz, all recordings were safely resampled at 4000 Hz. However, the duration of the respiratory cycles varied from 0.2 s to 16 s with an average of 2.7 s as observed in the study (Nguyen and Pernkopf, 2020). To begin with, the audio files in .wav format were imported and read as part of the preprocessing step. The "librosa" Python library was used to load the sound files and convert them into digital signals. With lower and upper cut-off frequencies of 50 Hz and 2000 Hz, respectively, a fifth-order Butterworth band-pass filter , was applied to eliminate any noise present in the signals. During this process, the sampling rates of each audio file were considered.

In this study, a Band Pass Filter (BPF) is used for signal processing and noise reduction. The BPF allows the signals within a specified frequency passband to pass through and attenuates signals at frequencies outside that band. This filtered signal can then be used for various signal feature extraction applications, including noise reduction, frequency boosting, digital audio equalization, and digital crossover. Compared to high and low pass filters, the BPF is more suitable for denoising and can play a significant role in speech signal processing applications such as automatic speech recognition, speech filtering, speech enhancement, and reduction of noise. Therefore, the BPF is a more appropriate choice for preprocessing lung sound signal classification (Kamble and Phadke, 2016) (Patel, 2013).

Python's bandpass filters combine high pass and low pass filters to only allow sounds inside a specific frequency range. The high-pass, low-pass, band-stop, and bandpass filters can be specified using the btype argument, which is a part of the method butter (). So, using Scipy's butter () method, the Butterworth filter may be applied to a signal. To flatten the frequency, we can create a digital or analogue Nth order Butterworth filter. The bandpass filter benefits from low and high filters to permit a certain frequency range (Fortune, 2018)(Lynch, 2018)(Pra and Fontana, 2020). Figure 3.1 demonstrates the results of applying a Butterworth band-pass filter with a frequency range of 50–2000 Hz on the URTI wave file called (101_1b1_pr_sc_Meditron.wav) as an illustration of the study's methodology.

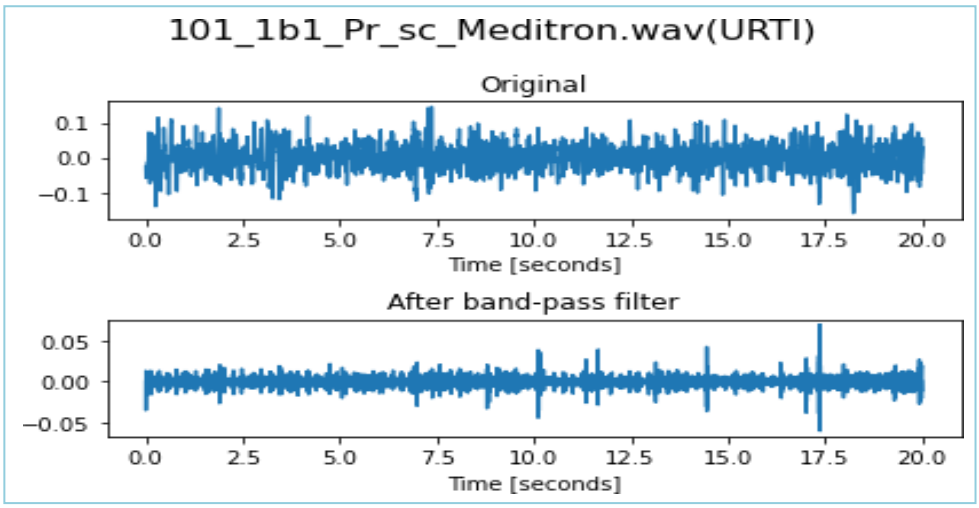

Figure 3. 1. URTI in original and after band pass-filter applied

By applying a Butterworth- band-pass filter to the overall lung sound and defining features with the sound's sample rate of 50-2000 Hz, we can eliminate noise and extract features using (CS-MDWTD) approach with the library named librosa for all data frames. So, the following lung sound signal of the patient person (target) samples with a different class name as converted the sound to the original image-digital signal and then filtered by BPF we can see accurate results without noise in the following samples figure that start from figure 3.2 to the figure 3.9 respectively. All figures illustrate the comparison between the signal in its original form and after applying a bandpass filter.

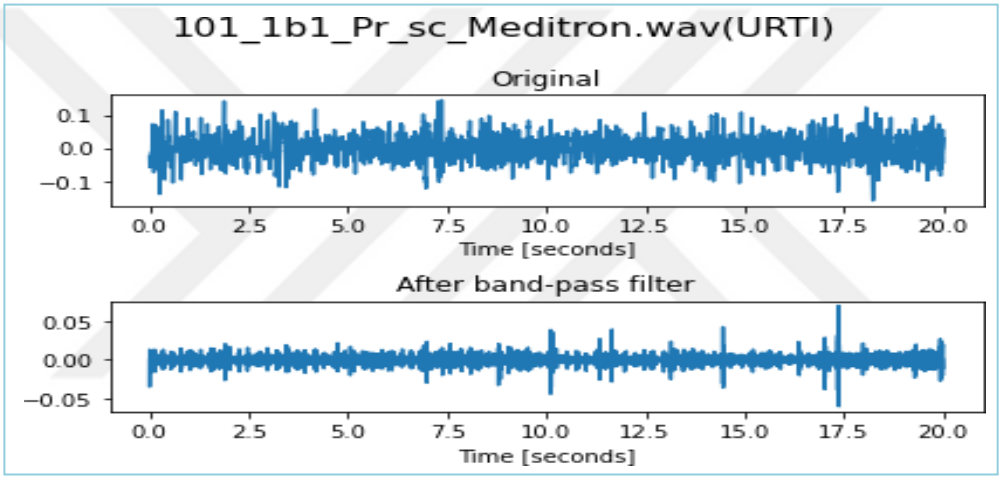

Figure 3. 2. URTI observed before and after implementing a bandpass filter

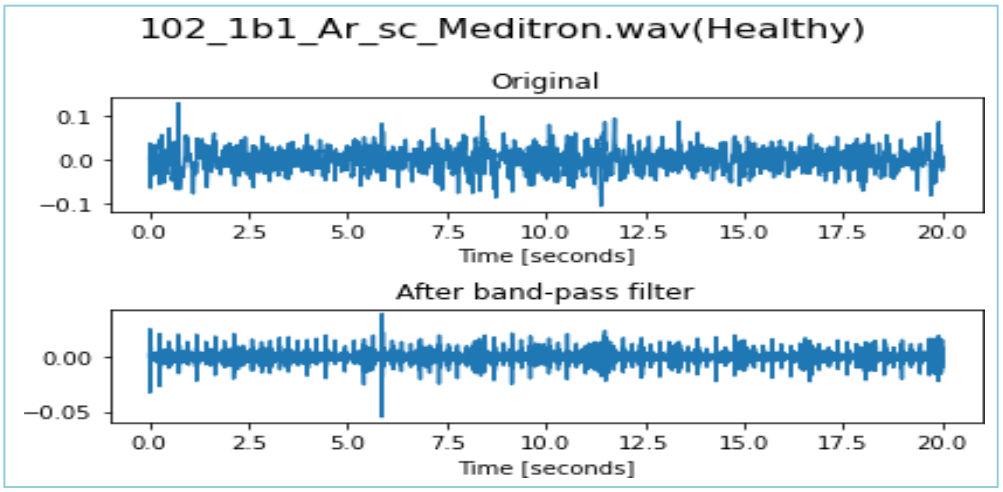

Figure 3. 3. Healthy observed before and after implementing a bandpass filter

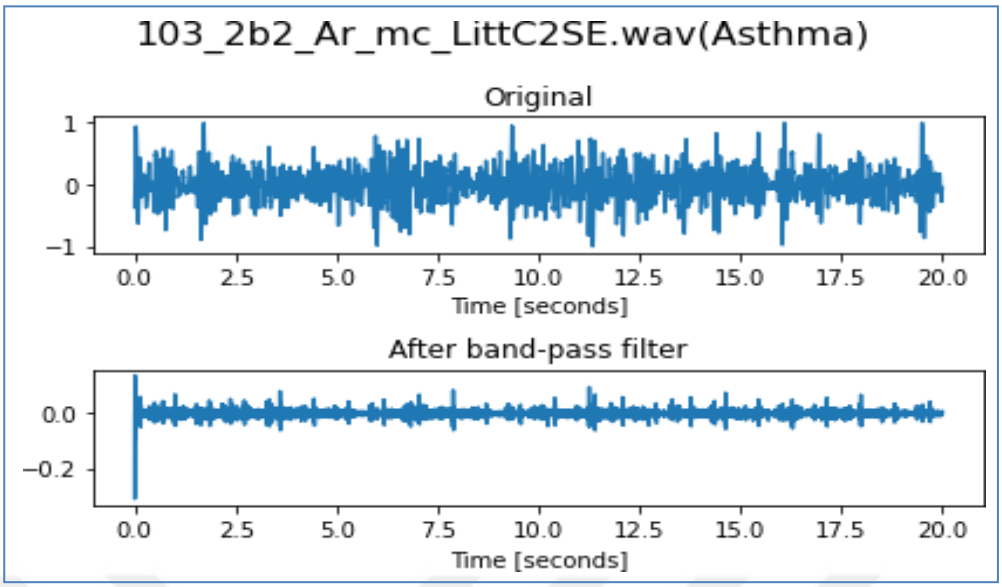

Figure 3. 4. Asthma in original and after band pass-filter applied

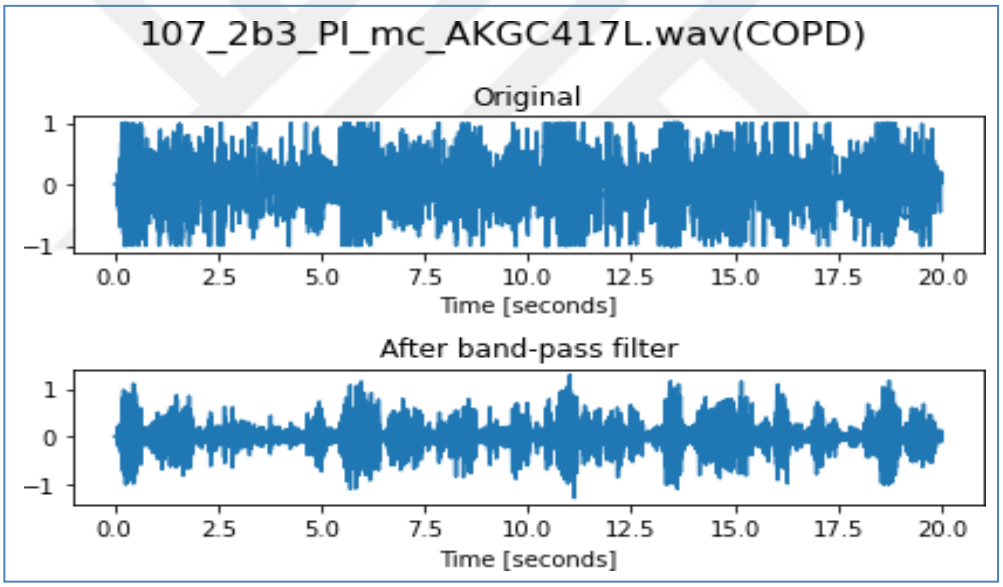

Figure 3. 5. COPD in original and after band pass-filter applied

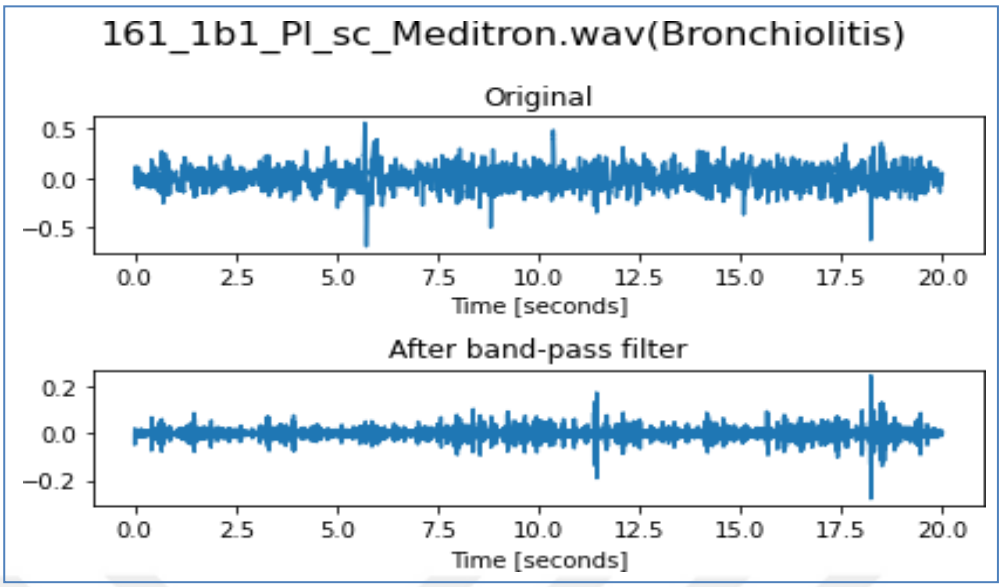

Figure 3. 6. Bronchiolitis in original and after band pass-filter applied

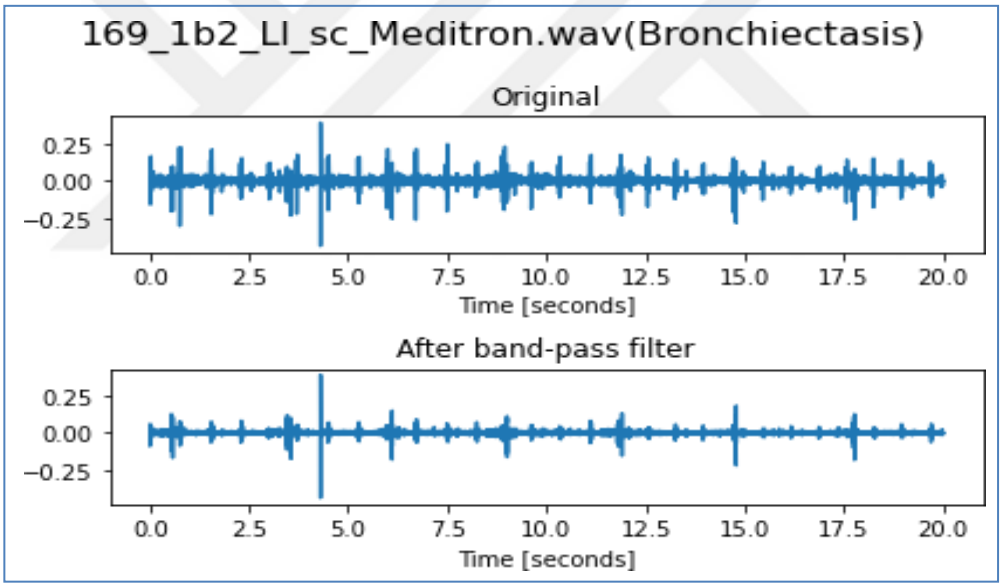

Figure 3. 7. Bronchiectasis in original and after band pass-filter applied

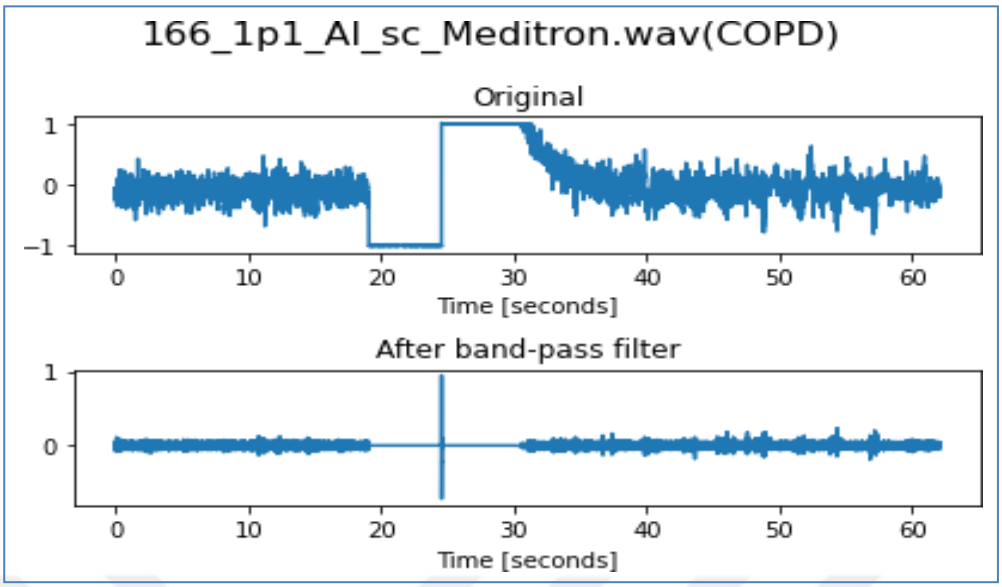

Figure 3. 8. COPD in original and after band pass-filter applied

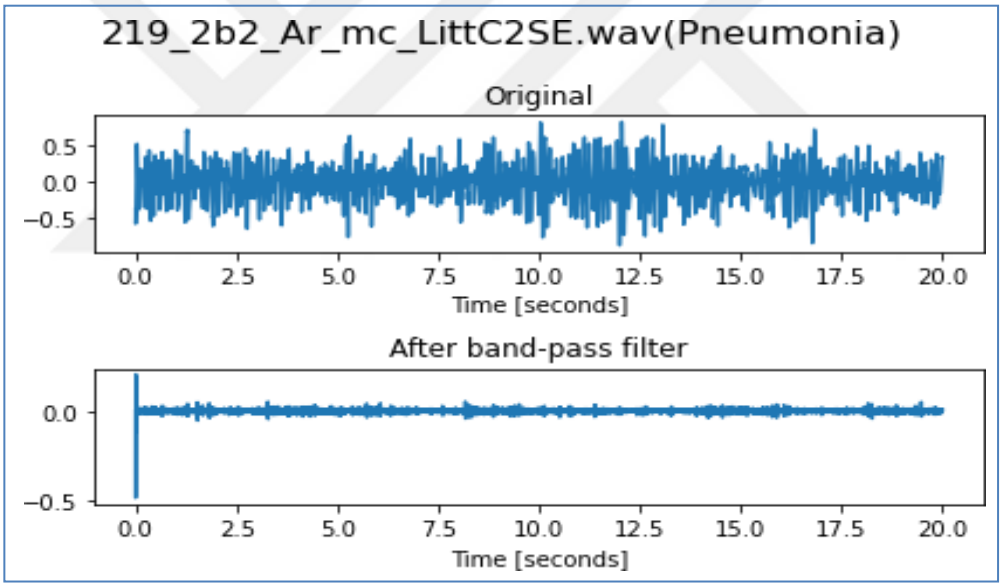

Figure 3. 9. Pneumonia in original and after band pass-filter applied

3.3. Feature extraction

In general, our study introduces a new model for classifying lung sound signals using a CNN mixing with LSTM. Additionally, the study proposed a novel preprocessing method that addresses the unique characteristics of the ICBHI data. By utilizing the cosine similarity-based multilevel discrete wavelet transform decomposition (CS-MDWTD) approach to cycles of preprocessed respiratory sound

signal data, we can determine the frequency content of the data and extract features from it. We extract the mean (average) feature type and select the most similar detail coefficient vector, ensuring all vectors are of the same size. Finally, the vectors are fed into a CNN-LSTM hybrid model to obtain the classification output.

3.3.1. Cosine similarity

A metric-measure that is used to compare two vectors or sequences is called cosine similarity (CS). The cosine similarity measure is computed as the cosine of the angle between two vectors. This value is obtained by taking the dot product of the vectors and dividing it by the product of their lengths. CS can be defined in the interval of [-1, 1]. CS is widely used in text mining (Jose, 2001) and text clustering (Victor et al., 2014)(Al-Anzi and AbuZeina, 2017) and so, used in other various fields, including signal processing and speech analysis. In these studies, texts are usually represented as vectors after applying different transformations sometimes, indeed. Since CS measures the similarity of vectors, we employed it in our study. The concept of CS is rooted in linear algebra and geometry. Specifically, this concept is related to the calculation of the dot product between two vectors and the angle between them. The dot product is obtained by adding the product of the corresponding components of the vectors. For two vectors A and B of the same dimension, the dot product can be defined as in Equation 3.1:

$$A \cdot B = A_1B_1 + A_2B_2 + \ldots + A_nB_n \tag{3.1}$$

Assuming A is composed of components A_1, A_2, ..., A_n, and B is composed of components B_1, B2, ..., B_n. To calculate the similarity between two signals, or to compute the angle between two vectors by using the dot product, we can use the cosine similarity formula as in Equation 3.2:

$$\text{similarity} = \cos(\theta) = (A \cdot B) / (||A|| \, ||B||) \tag{3.2}$$

Equation 3.2 shows the formula for cosine similarity, which is used to calculate the similarity between two signals or the angle between two vectors. It involves the

dot product of the (A.B) vectors divided by the product of their (||A|| ||B||) magnitudes. The cosine similarity value ranges between -1 and 1, with 1 indicating identical signals and -1 indicating completely dissimilar signals. The vector magnitude can be calculated by taking the square root of the square's summation of its individual components. The dot product of two vectors is calculated by multiplying the corresponding components of the vectors and summing them up, as shown in equation 3.3.

$$A \,.\, B \; = \Sigma \, (A_i \; * \; B_i) \tag{3.3}$$

Where Ai and Bi are the elements of the vectors A and B respectively, and the summation is over the length of the vectors.

And so, the magnitude of a vectors ||A||, and ||B|| can be computed as follows, respectively:

$$||A|| \; = \; \sqrt{(\Sigma \; (A_i)^2)} \tag{3.4}$$

Where Ai is an element of the vector A and the summation is over the length of the vector.

$$||B|| \; = \; \sqrt{(\Sigma \; (B_i)^2)} \tag{3.5}$$

Where Bi is an element of the vector B and the summation is over the length of the vector.

Hence, the division of the dot product of two vectors by the product of their magnitudes yields the cosine value, which represents the angle between the vectors. Now, the CS formula is in Equation 3.6.

$$\text{Similarity}=\cos(\Theta) = (A.\, B) \,/\, (||A|| * ||B||) = CS = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.6}$$

Where A and B are distinct vectors represented signals to be compared, i is the index of elements, and n is the number of elements (Park et al., 2020).

Normally, in this study, numerical data is obtained. So, there is no need for any transformation to measure CS. After applying CS, it can be detected that if the vectors indicate whether same direction or not. In essence, cosine similarity calculates the cosine of the angle formed by two vectors by dividing their dot product by the product of their magnitudes. The resulting cosine value indicates the similarity between the two vectors, with the angle between them being a significant factor. A cosine value of -1 implies that the vectors are opposing (orthogonal or dissimilar). Otherwise, when it gets closer to 1, it means that a higher similarity is observed (Al-Otaibi et al., 2022). When comparing detail coefficient vectors, the vector which is most similar to the other two vectors is chosen, because since it is similar it is a lower chance to lose details. It is probable that this vector has the characteristics of other two vectors. It is possible to have a robust method by combining CS with DWT. Discrete Wavelet Transform (DWT) is a method of signal processing that that enables the decomposition of a signal into different frequency bands. This technique can be useful in identifying and filtering out noise present in the signal. We can compare signals in a robust manner, even in the presence of noise. In addition, by comparing the CS of signals at different scales, we can gain a more comprehensive understanding of the signals. That is, we can compare signals that have similar time-frequency content, even if they have different amplitude and phase characteristics.

Cosine similarity offers several advantages. Firstly, it is a fast and efficient method for measuring similarity, making it an excellent technique for processing large datasets such as the ICBHI dataset. Secondly, it is robust to small variations in the data, which makes it a useful technique for processing lung sound signals that may contain variations due to noise or other artefacts. Thirdly, cosine similarity can be used for multi-class classification, which is essential for classifying the ICBHI dataset into different categories such as healthy, unhealthy, or pathologic. Fourthly, cosine similarity can determine the similarity between signals obtained from a patient and a reference signal. A high cosine similarity value signifies that the two signals are similar, while a low value indicates that they are dissimilar. Thus, cosine similarity is a useful tool for classifying lung sounds into different categories based on the similarity between the signals.

3.3.2. Multilevel discrete wavelet transform decomposition (MDWTD)

The conventional wavelet transform (WT) involves applying the function of wavelet with translation and dilation to generate small waveforms (Camps-valls et al., 2013). In DWT, these wavelets are sampled discretely. DWT offers a multiresolution wavelet analysis, which allows the signals to be represented and analyzed at multiple resolutions. This enables features that are not detected at one level to be detectable at another level, making it a valuable tool for signal analysis (Plaza et al., 2009).

During the process of DWT, signals are subjected to both high-pass and low-pass filters (Ni and Ma, 2015). The output of high-pass filters is detail coefficients (high-resolution) whereas the output of low-pass filters is called approximation coefficients (low-resolution). The MDWTD method generates a sequence of signals by iteratively deriving fresh approximation and details coefficients from the output of low-pass filters and high-pass filters used to the previous approximation coefficients. In this study, Daubechies 1 (db1) is employed for MDWTD and the number of levels is set to 3. A demonstration of MDWTD is assumed in Figure 3.10. below.

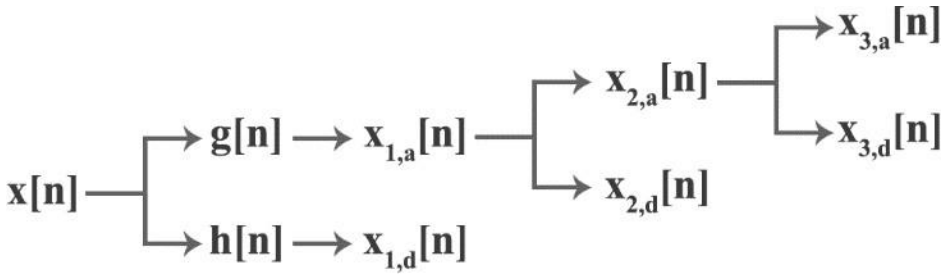

Figure 3. 10. Illustration of recursive MDWTD

Therefore, the process of signal decomposition in DWT results in the splitting of the input signal x[n] into two sub-bands, which are represented by the low pass filter signal g[n] and the high pass filter signal h[n].

3.3.3. The proposed cosine similarity-based multilevel discrete wavelet transforms decomposition (CS-MDWTD)

Multilevel decomposition is applied for 3 levels. A detailed coefficient vector is obtained for each level using the classical MDWT. The challenge in diagnosing signals is to determine which vector is most representative of the data. To address this, a new method has been proposed that selects the most suitable detail coefficient vector. This method involves two steps: using MDWT to extract the coefficient details and using CS to select the best detail coefficient vector. After obtaining three different detail coefficient vectors employing MDWT, the vectors are trimmed to the same size, which is the length of the smallest vector. And, the Equation 3.7 is applied to use CS:

$$CE_{best} = MAX(CS_A(CE_1)), (CS_A(CE_2)), CS_A(CE_3)) \quad (3.7)$$

Where CE_{best} is a vector of detail coefficients that is the most suitable., and CS_A is the average CS (which is assumed in Equation (3.6)) of each vector with the other two vectors, which can be computed by using Equation 3.7.

$$CS_A = \frac{CS(CE_x,CE_y)+CS(CE_x,CE_z)}{2} \quad (3.8)$$

Where CE_x is the related detail coefficients vector, and CE_y and CE_z are two other detail coefficients vectors.

In the end, the proposed method of feature extraction is used to crop the feature vectors to the same size, which is a vector of smallest size. This is important for classification purposes. Each feature vector consumes a length of 1x18, which is a fixed size set by the method. Figure 3.11 illustrates the block diagram of the suggested method from the step of input to the classification output to become input to the model.

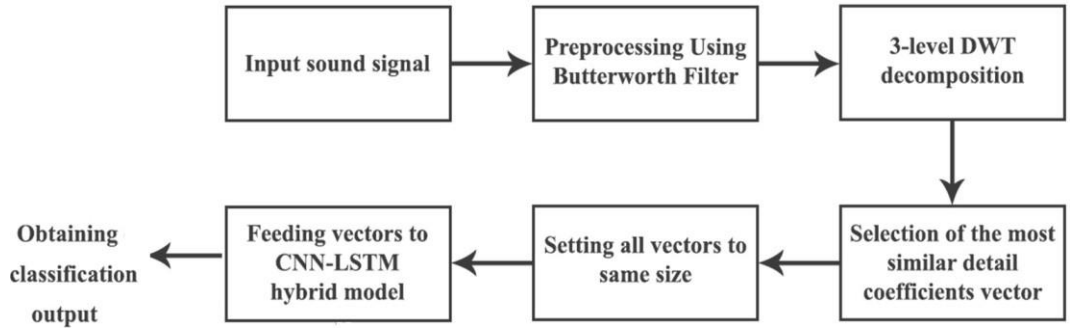

Figure 3. 11. Block diagram of the proposed method, from input to the classification

Typically, research methodology places great emphasis on dataset preprocessing, since it heavily influences the generalization performance of DL algorithms (Rahaman and Rasel, 2021). Subsequently, the dataset undergoes feature selection and extraction, and other processing steps, before being fed into a convolutional neural network (CNN) model. CNNs are also used in audio processing and are a type of feedforward neural network. They have artificial neurons that respond to neighboring units within their coverage area. Initially, CNNs were developed for image processing but have since been used for natural language as well. The most critical components of a CNN are the convolution layer and pool layer (Li et al., 2020).

To avoid errors caused by inconsistent and noisy data, several preprocessing steps are taken (García et al., 2015). One of these steps involves using a band-pass filter to remove noise from lung sound signals. To select the most significant features in the dataset, feature selection is typically performed either manually or with the help of algorithms that remove redundant features without affecting the data pattern. In this research, the method of CS-MDWTD with the librosa library is utilized for feature selection and extraction. These involves selecting the most similar detail coefficient vectors, resizing them to the same length, and feeding them to a CNN-LSTM hybrid model for classification output (Tsai et al., 2009).

3.4. Classification

In this section, the used deep learning algorithms are covered and the classification technique is explained.

3.4.1. Convolutional neural networks (CNNs)

CNN is a kind of neural network (NN) and it is a neural network architecture used extensively in medical imaging (Ohata et al., 2021), image classification (Hassan et al., 2021) and image segmentation (Mecheter et al., 2022). Convolutional layers are responsible for applying filters to the input, therefore obtaining feature maps (Zhang et al., 1990). Since CNNs down sample the input data fed to them, they are not invariant to translation (Mouton et al., 2020).

CNNs are a type of multilayer perceptron, hence, they are typically fully connected. Full connectivity might lead to overfitting, in which with the data training the model is only familiar so it is not very successful at classifying different data. CNN fix this problem by applying different techniques (i.e., dropout). Compared to other algorithms, CNNs require relatively less pre-processing. During training, CNNs learn to build the filters (kernels) themselves, whereas in traditional algorithms, these filters are manually designed. This automated feature extraction is a significant advantage. CNN also has components, specialized activation functions and an optimizer. These components will be explained in this section. In the study, the CNN employed a convolutional layer, utilized a binary cross-entropy loss function, a categorical cross-entropy loss function, and employed a SoftMax activation function.

Convolutional layers are specialized for processing images, they can manipulate 1D, 2D or 3D data. The 1D convolutional layer is utilized in this work to process coefficient vectors. Equation (3.9) may be utilized to do a 1D convolution process.

$$y[n] = \sum_{k=-\infty}^{\infty} c[k].\, h[n - k] \tag{3.9}$$

In this process, an input vector or signal is represented by c[k], a filter is represented by h[n], and the output is represented by y[n], which generates feature maps. The filter is applied by sliding it through each window to obtain feature maps. To fit the features within the range of [0, 1] and create a probability distribution, an activation function is required. The SoftMax formula is given by Equation 3.10.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.10}$$

The SoftMax function is used as an activation function to convert the output of a convolutional layer into a distributed of probability. The formula for SoftMax is expressed in equation 3.10, where z is the input vector, e is the exponential function, K is the number of classes, and i and j are the indices for the i-th element in the input vector and the j-th class, respectively. The denominator term in SoftMax is used for normalization. For minimizing the error of classification, the loss function is used, and two loss functions are used in this study: the binary cross-entropy loss function (equation 3.11) and the sparse categorical cross-entropy loss function (equation 3.12).

$$BCE = -\frac{1}{n}\sum_{i=1}^{n} (Y_i . \log \hat{Y}_i + (1 - Y_i) . \log(1 - \hat{Y}_i)) \tag{3.11}$$

$$CCE = -\sum_{i=1}^{n} Y_i . \log \hat{Y}_i \tag{3.12}$$

Where the number of classes is n, i is the index, Y is the ground truth label, $\hat{Y}$ is the predicted probability, and BCE and CCE are results.

3.4.2. Long short-term memory

LSTM (Hochreiter and Schmidhuber, 1997) is a kind of artificial neural network (ANN) architecture that is commonly used in artificial intelligence and deep learning. In contrast to conventional feedforward neural networks, (LSTM) networks incorporate feedback connections that allow them to effectively handle sequential data, including sound signals and time series. LSTM possesses both short-term memory and long-term memory as a kind of (RNN). The biases and weights networks are updated

after each training iteration, mirroring the physiological changes that occur in synaptic connections that store long-term memories. Furthermore, the network's activation patterns are dynamically updated at each time step, resembling the brain's ability to retain short-term memories through the modulation of moment-to-moment variations in electric firing patterns (Elman, 1990).

LSTM is unique among other architectures because it uses "LSTM units" that have three gates: an output gate, forget gate, and an input gate (Gers et al., 2000) The input gate, output gate, and forget gate are denoted by equations 3.13, 3.14, and 3.15, respectively.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{3.13}$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{3.14}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{3.15}$$

Where the gate of input is i_t, the gate of forget is f_t, o_t is the output gate, w is the weights for the related gate, x is the input at timestamp t, b is the biases for the related gate and h_{t-1} is the output of the timestamp t-1 (previous) LSTM unit and σ is the sigmoid function given in Equation 3.16.

$$\sigma = \frac{1}{1+e^{-x}} \tag{3.16}$$

The LSTM unit operates by using cell states. The cell state is given by Equation 3.17, while the candidate cell state is given by Equation 3.18. The final output is given by Equation 3.19:

$$d_t = tanh(w_c[h_{t-1}, x_t] + b_c) \tag{3.17}$$

$$c_t = f_t * c_{t-1} + i_t * d_t \tag{3.18}$$

$$h_t = o_t * \tanh(c^t) \tag{3.19}$$

Where at timestamp t the cell state is c_t, the candidate cell state is d_t at timestamp t and other symbols are the same as explained before. Equation 3.20 Denoted the formula of tanh function. An illustration of the flow through a complete LSTM unit is given in Figure 3.12.

$$tanh = \frac{(e^{x}-e^{-x})}{(e^{x}+e^{-x})} \quad (3.20)$$

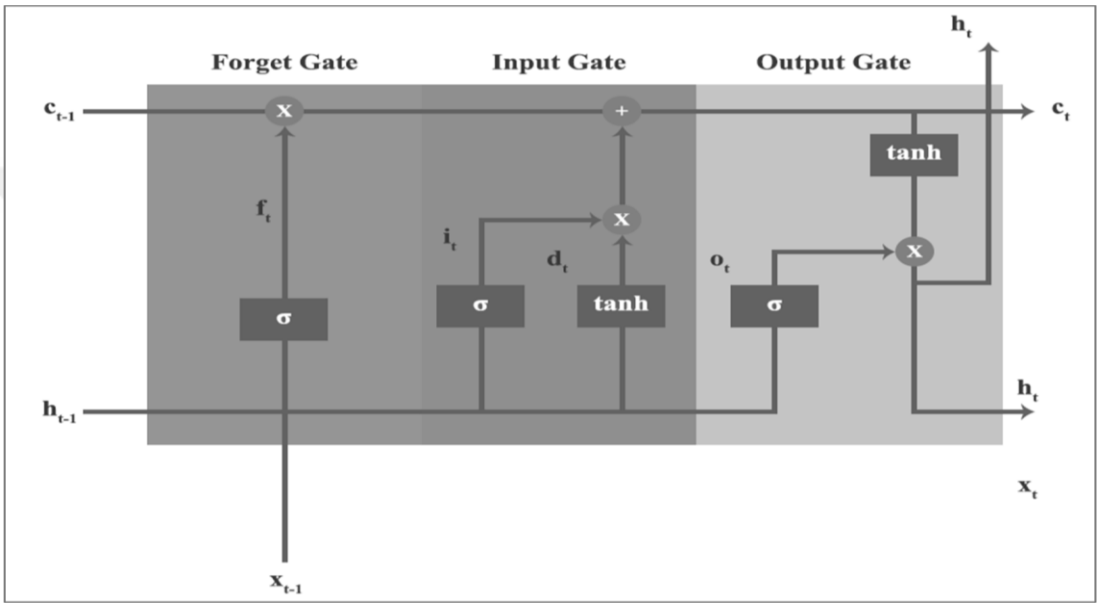

Figure 3. 12. Illustration of the flow through a complete LSTM unit (LSTM Network internal architecture)

The LSTM cell is composed of various inputs and outputs, as shown in Figure 3.12. These elements encompass the input signal at each time step (X_t), the current memory cell output (C_t), the output of the previous memory cell (C_{t-1}), the current hidden unit output (H_t), and the output produced by the previous hidden unit (H_{t-1}). The forget gate regulates and controls the impact of past information on the current memory cell, while the input gate governs the amount of input from the current moment that is retained in the cell of memory. The control gate manages the process of updating the memory cell contents from C_{t-1} to Ct based on the actions of accounting for the output of f and i. Additionally, the output gate determines the degree to which the current moment's internal state influences the external state. The update process for these four gates is expressed mathematically in Equations 3.21 to 3.25, where the

symbols $\otimes$ and $\oplus$ indicate vector multiplication and addition, respectively, and σ denotes the activation function.

$$ft = \sigma(Wxfxt + Whfht - 1 + bf) \quad (3.21)$$

$$it == \sigma(Wxixt + Whiht - 1 + bi) \quad (3.22)$$

$$ot = \sigma(Wxoxt + Whoht - 1 + bo) \quad (3.23)$$

$$ct = ft \times ct - 1 + it \times \tanh(Wxcxt + Whcht - 1 + bc) \quad (3.24)$$

$$ht = ot \times \tanh(ct) \quad (3.25)$$

In each LSTM update process, the corresponding weight matrices Wxf, Wxi, Wxo, Wxc, Whf, Whi, Who, and Whc, as well as bias vectors bf, bi, bo, and bc are updated. These weights and biases are used to calculate the input gate i_t, the forget gate ft, the cell state ct, the output gate Ot, and the final output of final ht at timestamp t , these parameters are updated during each iteration of the LSTM algorithm. (Deng et al., 2022).

3.4.3. The used hybrid deep learning architecture

In the literature, researchers use CNN-LSTM hybrid models to combine the strengths of two architectures to create a more strong model for acquiring temporal-spatial information (Bao et al., 2021);(Dai et al., 2021). Table 3.1. provides a concise overview of the model.

Table 3.1: The model Architecture

Type of layer	The shape of output
Input Layer	(18, 1)
1D Convolutional Layer	(16, 128)
1D Max poling	(8,128)
Dropout	(8,128)
Layer of LSTM	256
Dense Layer	No. Classes

Table 3.1. Demonstrated the CNN-LSTM model structure of class classification tasks, that contains input layers with reshape layers, conv1D, Max poling1D, Dropout, LSTM, dense layer, and showed output shape of the execution code operations parameters in trainable and nontrainable. The full details about the model structure are shown in the Figure 3.13., the hybrid models architecture and Figure 3.14., the Flowchart of the CNN-LSTM hybrid model below, respectively:

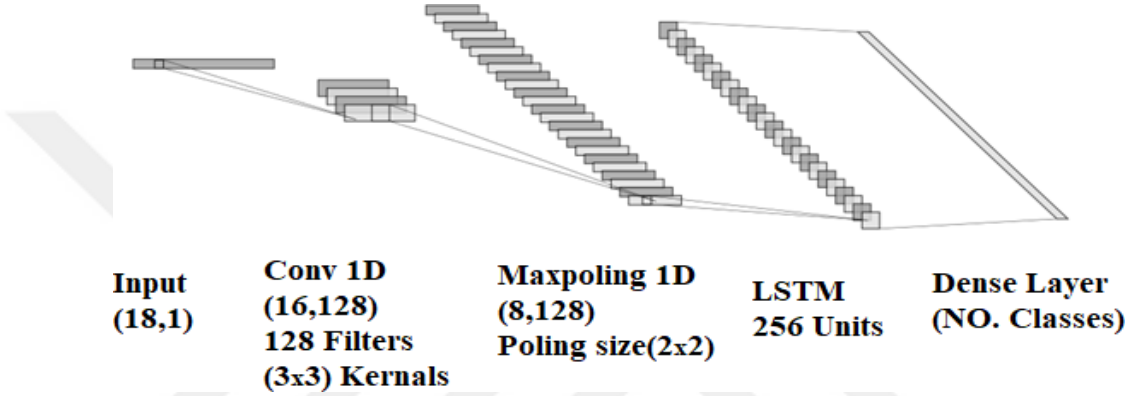

Figure 3. 13. The fully CNN-LSTM hybrid models architectur

In general, the CNN-LSTM hybrid model utilized in this study takes the output vectors of the method and utilizes them as inputs for classification purposes. The model is made up of four main layers: the input layer, the convolutional layer, the LSTM layer, and the output layer. The input layer handles the input data and has a size of 1 × 18, which is the size of each feature vector. The convolutional layer is responsible for filtering out noise from the input data and extracting valuable features from it, with 128 filters and a 3 × 3 kernel size. SoftMax activation function is applied to these filters, as explained in the preceding subsection. The LSTM layer, consisting of 256 units, is used to capture short-term and long-term dependencies in the data. For binary classification tasks (healthy vs. unhealthy), the output layer uses the sigmoid activation function and has one neuron, while for multi-class classification tasks (i.e., chronic vs. non-chronic vs. healthy), the SoftMax activation function is used, Furthermore, the neurons number is equivalent to the classes number in this layer.

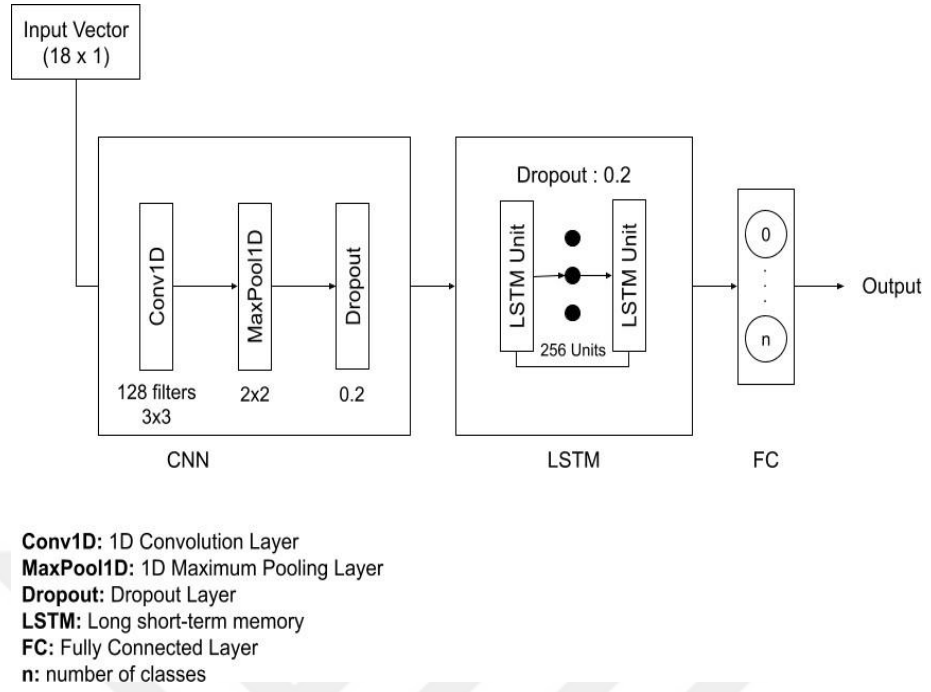

Figure 3.14 Flowchart of the CNN-LSTM hybrid model

After the selections of the most similar detail coefficients vector and setting all vectors to the same size and feeding vectors by CS-MDWTD method to CNN-LSTM hybrid model, this model shows the finalist steps of work of classification as shown in Figure 3.14 above, it takes features vectors from the output of method and it is an input of model in classification tasks during training. however, the steps of our model's work start from the feature vectors block that it is (input of model), CNN feature extraction block (Conv1, Max polling of size 2*2, Dropout of 0.2), sequence learning block (LSTM, LSTM with Dropout 0.2), the fully Connected layer, and the output layer. So, the dropout layer is a common technique utilized in models deep learning to address the problem of overfitting. It randomly drops out some neurons during training to prevent them from becoming too specialized to the training data. In this study, a dropout layer was added between the CNN of extraction feature and the LSTM of sequence learning layers to prevent overfitting. The output of the sequence learning block was then directed to a dropout layer, and subsequently to a fully connected layer, to generate the final output.

4. RESULTS and DISCUSSIONS

4.1 Experimental results

Firstly, this paragraph describes the classification tasks used in the study, including a table with the classes The study involves two distinct classification tasks: a 2-class task that distinguishes between (healthy and unhealth), and a 3-class task that classifies data as (healthy, chronic, or non-chronic). The classes are also presented in a table for reference. The number of samples for the 3-class classification is given in Table 4.1., and for the 2-class classification is given in Table 4.2., below. Before dividing the raw dataset into subsets, it consisted of 920 samples in total. Since the data is imbalanced in distribution, the waves are split into chunks which lasts between 1.3 seconds and 2.5 seconds, for better classification. CS-MDWTD features are extracted from these chunks. Feature vectors are of size 1x18.

Table 4.1 The distribution of samples among classes for the 3-class classification task

3-class		Total number of samples
Healthy	35	920
Chronic	810	
Non-chronic	75	
3-class after splitting		
Healthy	256	768
Chronic	256	
Non-chronic	256	

Table 4.2. The Sample Distribution in Classes for a two-Class categorization

2-class		The number of samples in total.
Healthy	35	920
Unhealthy	885	
2- class after splitting		
Healthy	200	400
Unhealthy	200	

Secondly, in evaluating model performance, two commonly used metrics are accuracy and loss function, each serving a different purpose. Accuracy measures how often a model correctly predicts the target variable, expressed by dividing the correct

number predictions with the predictions total number. This metric is typically used in classification problems where the goal is to correctly classify data into different categories. In contrast, a loss function calculates the difference between the predicted output and the actual output, serving to optimize the model by minimizing this error during training. The most common loss function used is mean squared error, though other types may be appropriate depending on the data and problem being addressed.

So, the accuracy and loss function are both metrics in machine learning that are quantified using mathematical equations. The equation (4.1) used for calculating accuracy is:

$$\text{Accuracy} = (\text{no. correct predictions}) / (\text{total no. predictions}) \quad (4.1)$$

And so, in this particular case, accuracy can be calculated as the percentage of correctly classified lung sound signals in the test dataset, as in equation 4.2 below:

$$\text{Accuracy} = (\text{no. correctly classified signals}) / (\text{total no. signals in the test dataset}) \quad (4.2)$$

Regarding the loss function, the choice of a specific loss function depends on the particularities of the problem and the type of data being used. For instance, one potential loss function for classifying lung sound signals is the categorical cross-entropy loss function, measures the disparity between the predicted probability distribution and the true probability distribution of the classes. This loss function can be mathematically represented as:

$$\text{Loss} = -(1/\text{N}) * \Sigma\,[\text{y} * \log(\text{y\_pred}) + (1 - \text{y}) * \log(1 - \text{y\_pred})] \quad (4.3)$$

Where N is the number of samples, y is the actual probability distribution of the classes (i.e., a one-hot encoded vector), and y_pred is the predicted probability distribution of the classes.

Now, to performance the model's evaluation, the accuracy rate is used as a metric. The true positive (TP) represents the predicted positive classes correctly, whereas the true negative (TN) showed the predicted of negative classes correctly. On

the other hand, false positives (FP) and false negatives (FN) are the opposite of the correct predictions. The accuracy rate can be computed utilizing Equation 4.4, which is provided below.

$$ACC = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{4.4}$$

Each classification task's accuracy is measured using the proposed CNN-LSTM hybrid architecture. For 2-class classification, an accuracy rate reached up to 80% is obtained using 80/20 split but for 2-fold-cross validation an accuracy rate is 75%. For 3-class classification, an accuracy rate of 50% is obtained using 80/20 split but for 2-fold-cross validation an accuracy rate reached 55%.

In this study, a hybrid model of 1D CNNs and LSTMs was chosen for the purpose of training and classification. The CNN is capable of extracting spatial features from the signal, whereas the LSTM captures features by focusing on variations in the time domain. Combining both networks often leads to improved model performance in predicting signals depend on both spatial and time-domain features. The model's performance was assessed both collectively as a combined network and individually as a CNN. Figure 4.1 illustrates the proposed method for training and evaluating the model, using 2-fold Cross Validation.

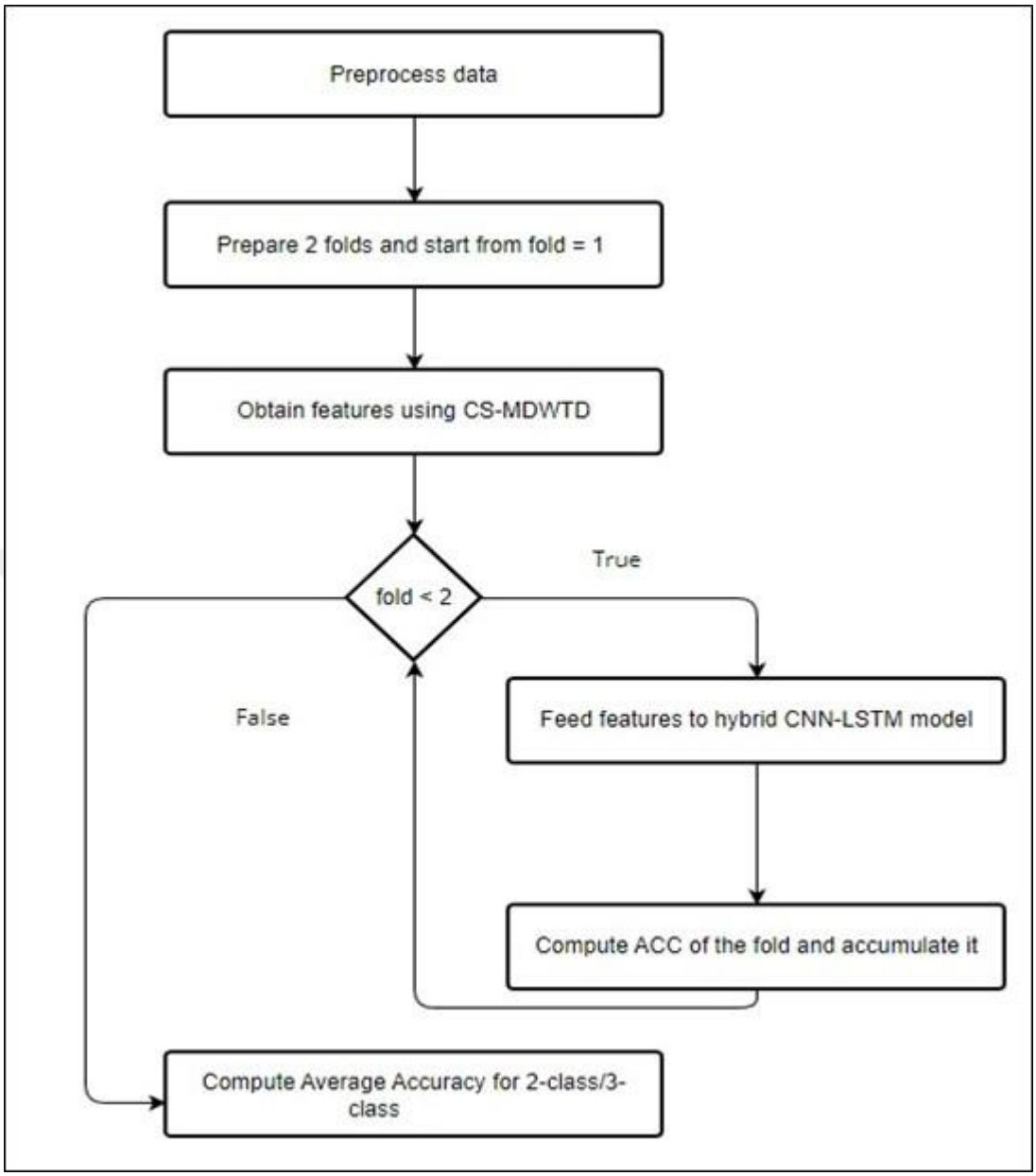

Figure 4.1. the bock diagram of the suggested method using 2-fold Cross Validation

The proposed method and model flowchart is depicted in the block diagram which incorporates pseudocode. The first step is the partitioning of data into two folds. Two-fold cross-validation is then implemented, and the training procedure is conducted for all fold groups. In each iteration, the model is trained and validated for one-fold, and then tested for two-fold. Subsequently, accuracy is computed. After the training is completed for all folds, the accuracy is obtained.

Each classification task's accuracy is measured using the proposed CNN-LSTM hybrid architecture under CS-MDWTD method. Therefore, Figures 4.2 to 4.7, respectively, show the accuracy and loss graphs for the 2-class and 3-class classification tasks with 80/20 split and k-fold-cross validation.

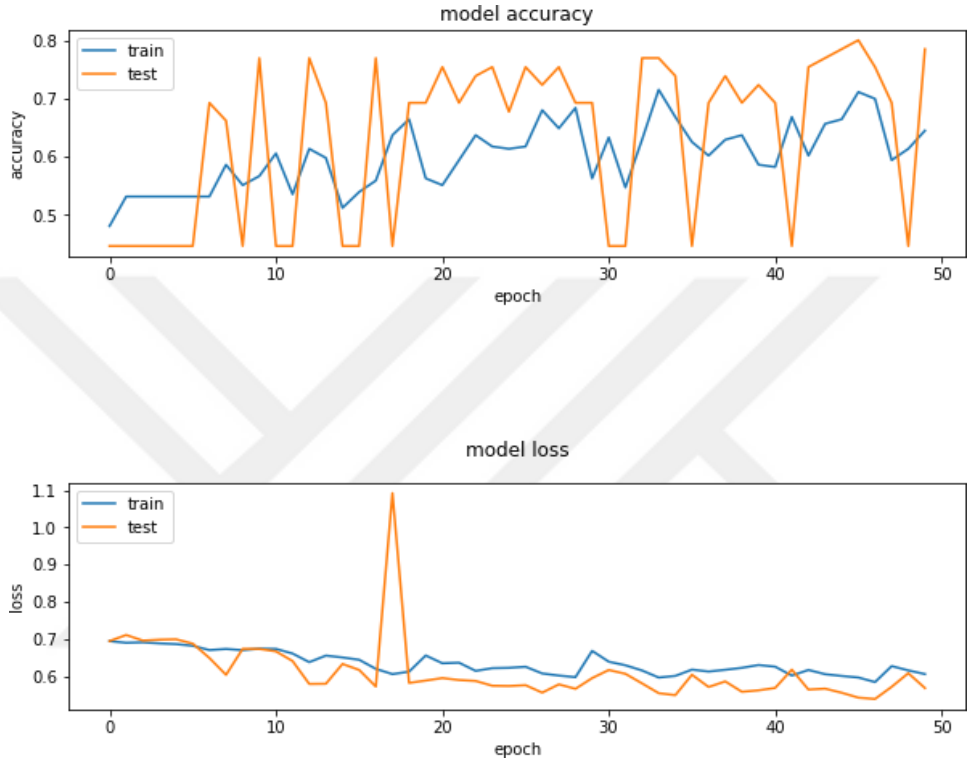

Figure 4.2. The accuracy and loss graph for 2-class classification using an 80/20 split

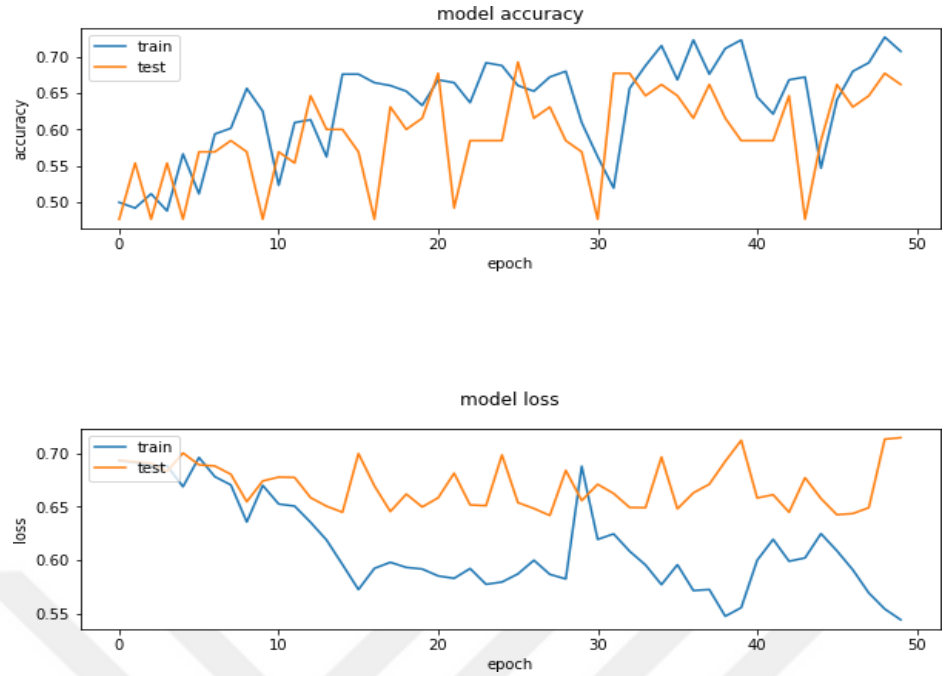

Figure 4.3: The accuracy and loss graph for 2-class classification using a 1-fold-cross validation split

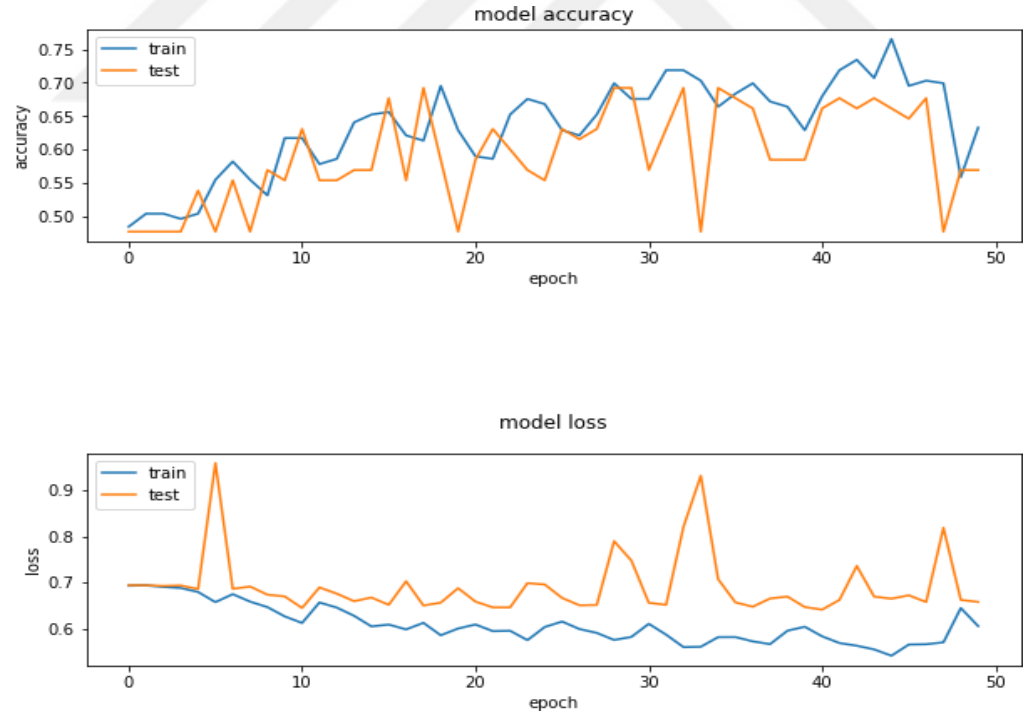

Figure 4.4 The accuracy and loss graph for two-class classification with a 2-fold-cross validation split

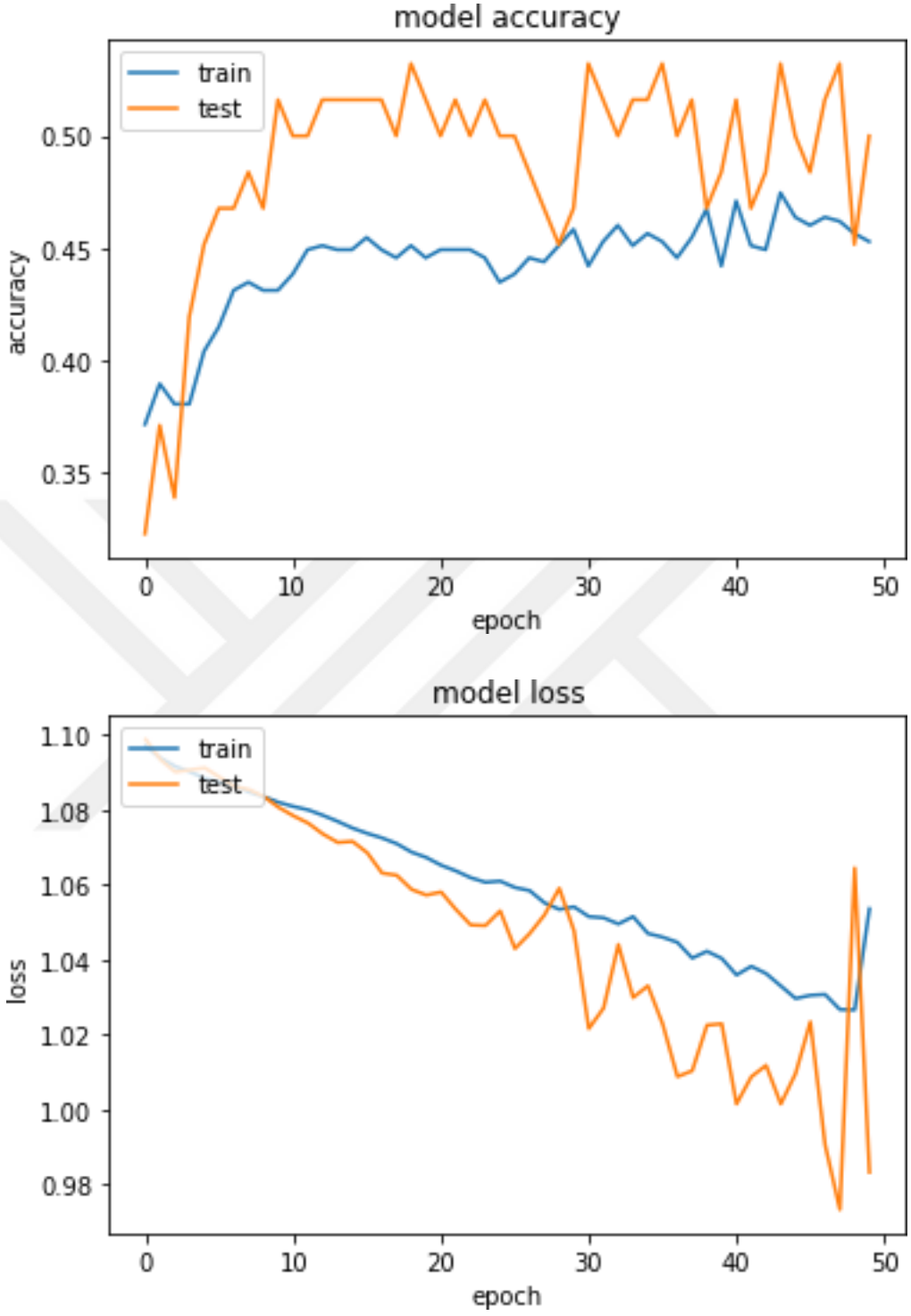

Figure 4.5 The accuracy and loss curve for three-class classification utilizing an 80/20 split

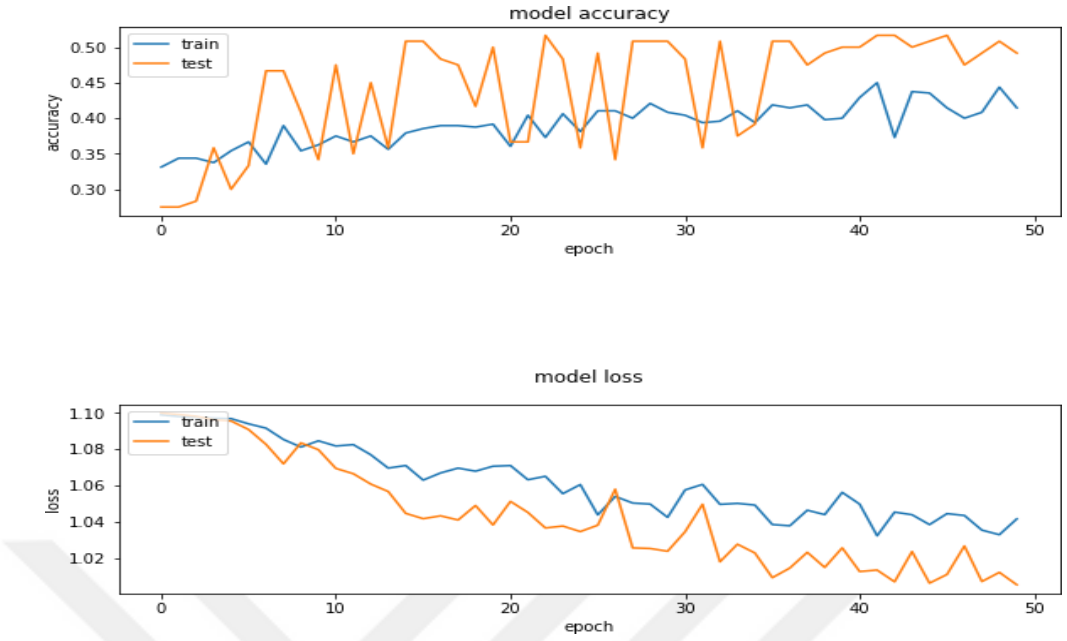

Figure 4.6 The accuracy and loss graph for 3-class classification using a 1-fold-cross validation split

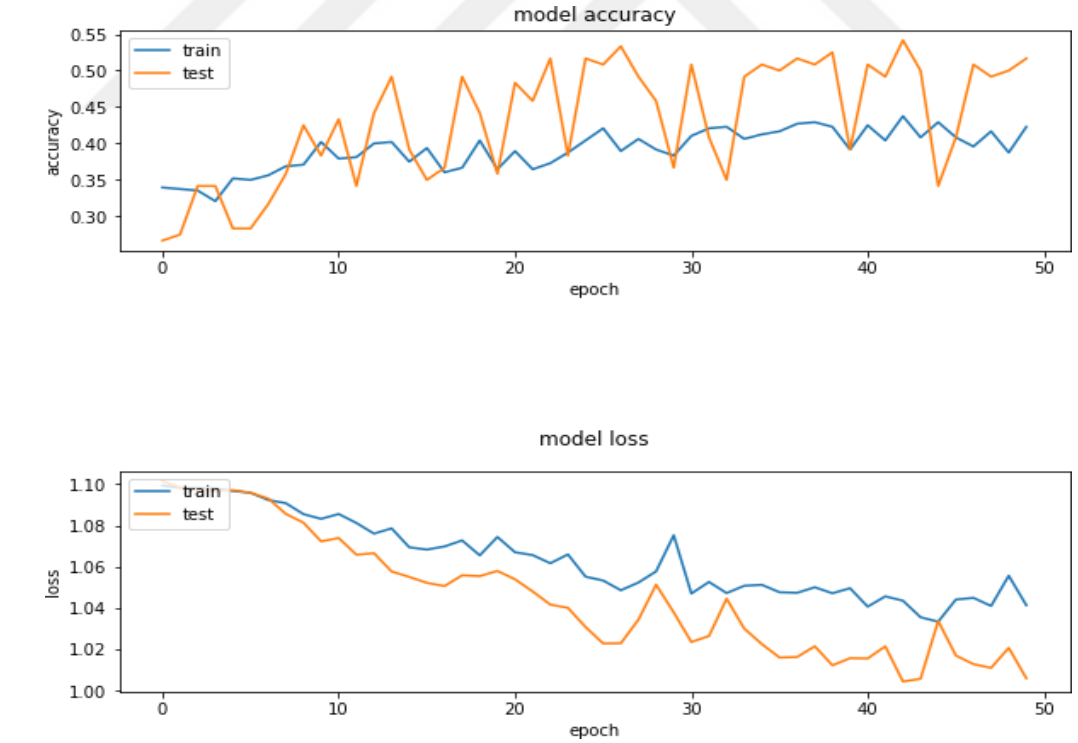

Figure 4.7 The accuracy and loss graph for 3 -class categorize utilizing a 2-fold-cross validation split

Finally, there are several metrics in this research used to analyze the classification confusion matrix and ROC curve for performance assessment of the developed model. After each fold, the confusion matrix was produced, and all assessment metrics were calculated from the collective confusion matrix generated by the 2-fold cross-validation method for the training and classification process. Therefore, the usual assessment measures for instance the accuracy, sensitivity, specificity, precision, and F1-score were calculated according to the values of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN). These metrics derived from the Confusion matrix. However, in our study, we opted to use four assessment measures: accuracy, precision, recall, and F1 score. Furthermore, the ROC curve was utilized as a visual tool to demonstrate the connection between the (FPR) and (TPR) across different threshold values. It is created by plotting the TPR on the y-axis against the FPR on the x-axis for different threshold values. The area under the ROC curve (AUC-ROC) is another common performance metric utilized to assess the binary performance classifiers. Its varieties between 0 and 1, representing the better performance with a higher value. In this context, the confusion matrix serves as a table that presents a concise overview of the model's performance, capturing the counts of false positive, true positive, true negative, and false negative classifications made by the model. This matrix is typically presented in a format that includes predicted positive and negative values, as well as actual positive and negative values. Specifically, it is represented as follows:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Positive
Actual Negative	False Negative	True Negative

The utilization of the confusion matrix is valuable in the classification of lung sound signals as it enables the calculation of multiple performance metrics such as precision, recall, and F1 score. In lung sound signal classification, the confusion matrix and ROC curve can be computed using the true labels and predicted labels of the model. The equations to calculate the metrics are as follows: First, the calculation formula of Precision:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (4.5)$$

Recall can be computed as follow:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (4.6)$$

F1 score can be calculated as:

$$F1\,Score = 2 * ((\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})) \quad (4.7)$$

The True Positive Rate (TPR) and the False Positive Rate (FPR) can be calculated in the Equation 4.8 and Equation 4.9, respectively:

$$\text{TPR} = \text{TP} / (\text{TP} + \text{FN}) \quad (4.8)$$

$$\text{FPR} = \text{FP} / (\text{FP} + \text{TN}) \quad (4.9)$$

The Sensitivity is TPR/Recall, and Specificity is TNR are in two equations below:

$$\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN}) \quad (4.10)$$

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP}) \quad (4.11)$$

The False Negative Rate (FNR) and the False Positive Rate (FPR) are in the two equations below:

$$\text{FNR} = \text{FN} / (\text{TP} + \text{FN}) \quad (4.12)$$

$$\text{FPR} = \text{FP} / (\text{TN} + \text{FP}) = 1 - \text{Specificity} \quad (4.13)$$

So, true positive (TP) denotes the count the samples positive that the classifier accurately identified, while False Positive (FP) represents the number of incorrectly classified positive samples. True Negative (TN) indicates the number of correctly classified negative samples, and False negative (FN) represents the count of negative samples that were mistakenly classified by the classifier. The graph's vertical axis (y-axis) corresponds to the (TPR), whereas the horizontal axis represents the false positive rate (FPR) in the plotted diagram. The AUC-ROC, a performance metric for binary classifiers, quantifies the quality of the classifier, with greater values indicating better performance.

In practice, the computation of the confusion matrix and ROC curve can be done using libraries such as sci-kit-learn in Python. Other metrics also is measured using the proposed CNN-LSTM hybrid architecture under the CS-MDWTD method for analyzing the classification confusion matrix and ROC curve to evaluate the performance of the developed model. Figures 4.8 to 4.13 present the confusion matrix and ROC curve for 2-class and 3-class classification, utilizing an 80/20 split and k-fold cross-validation. Similarly, Tables 4.3 to 4.8 provide corresponding results for the evaluation of performance metrics.

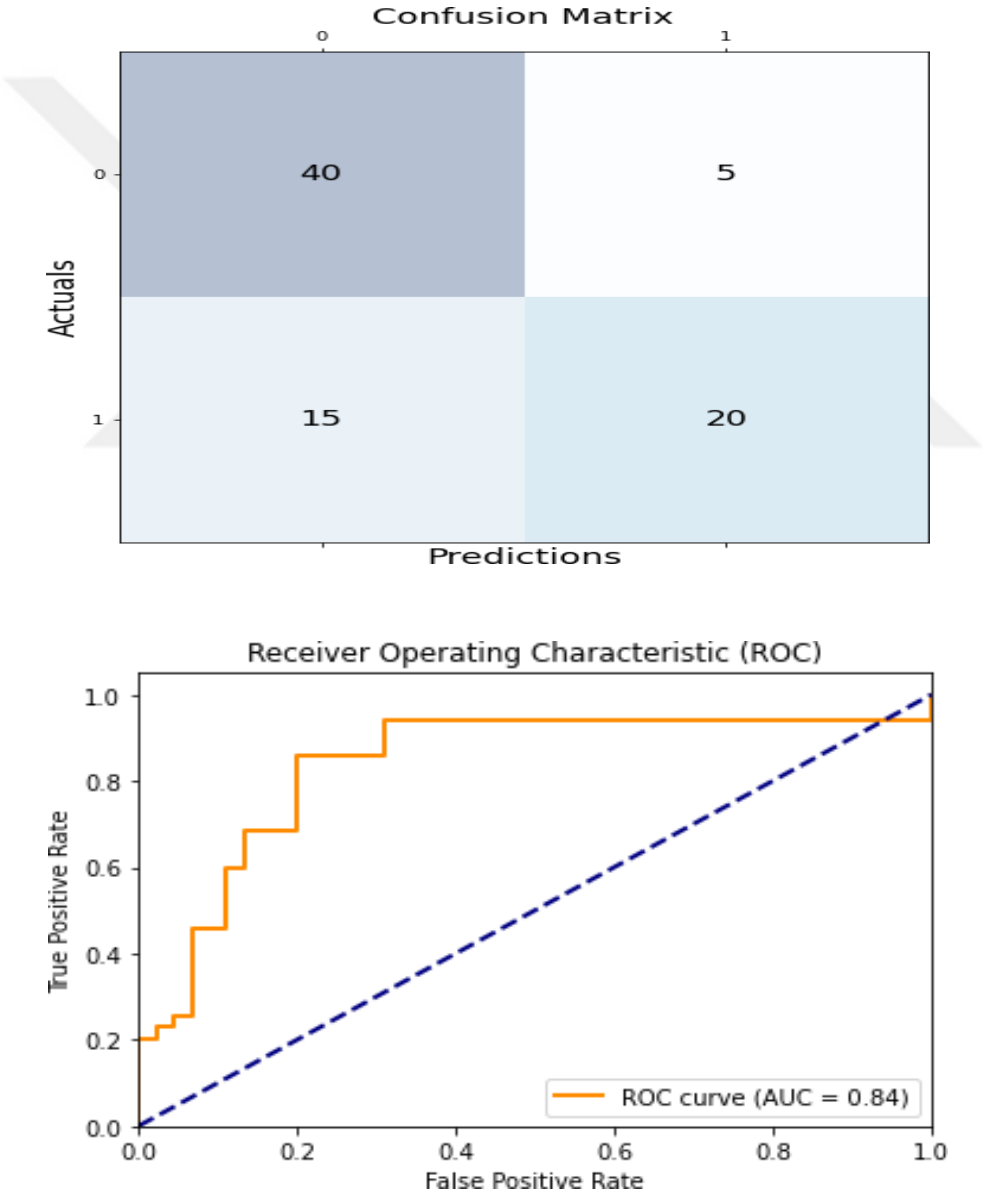

Figure 4.8. The Confusion Matrix and ROC curve for two-class categorization using an 80/20 split

Table 4.3. The other Metrics of 2- class classification an 80/20 split

	precision	recall	f1-score	support
0.0	0.73	0.89	0.80	45
1.0	0.80	0.57	0.67	35
accuracy			0.75	80
macro avg	0.76	0.73	0.73	80
weighted avg	0.76	0.75	0.74	80

A model that doesn't produce any false positives has a precision score of 1.0 and a recall score of 1.0. In our 2-class classification model, the precision score is 0.80, meaning that 80% of the time, it predicts correctly. Meanwhile, the recall score is 0.57, indicating that it can identify 57% of all 2-class classification correctly. Generally, improving precision may reduce recall, and vice versa. To consider both precision and recall, various metrics such as F1 score have been developed. In our confusion matrix, we have 40 TP, 20 TN, 5 FP, and 15 FN. The ROC curve is a widely used evaluation metric in binary classification, depicting the association between the true positive rate (TPR) and the false positive rate (FPR) across a range of threshold values. It serves as a valuable tool for assessing the performance of binary classification models. The AUC, or area under the ROC curve, is a compact metric that captures the overall discriminatory power of the model. It ranges from 0 to 1, where a value of 0.0 signifies a model that makes solely incorrect predictions, while a value of 1.0 indicates a model that makes solely correct predictions. The AUC serves as a concise measure of the model's ability to distinguish between classes and assess its overall performance. Any transformation of the predictions that preserves the relative ranking has no effect on the AUC. The performance of a categorization model at every classification threshold is essentially shown by the ROC curve.

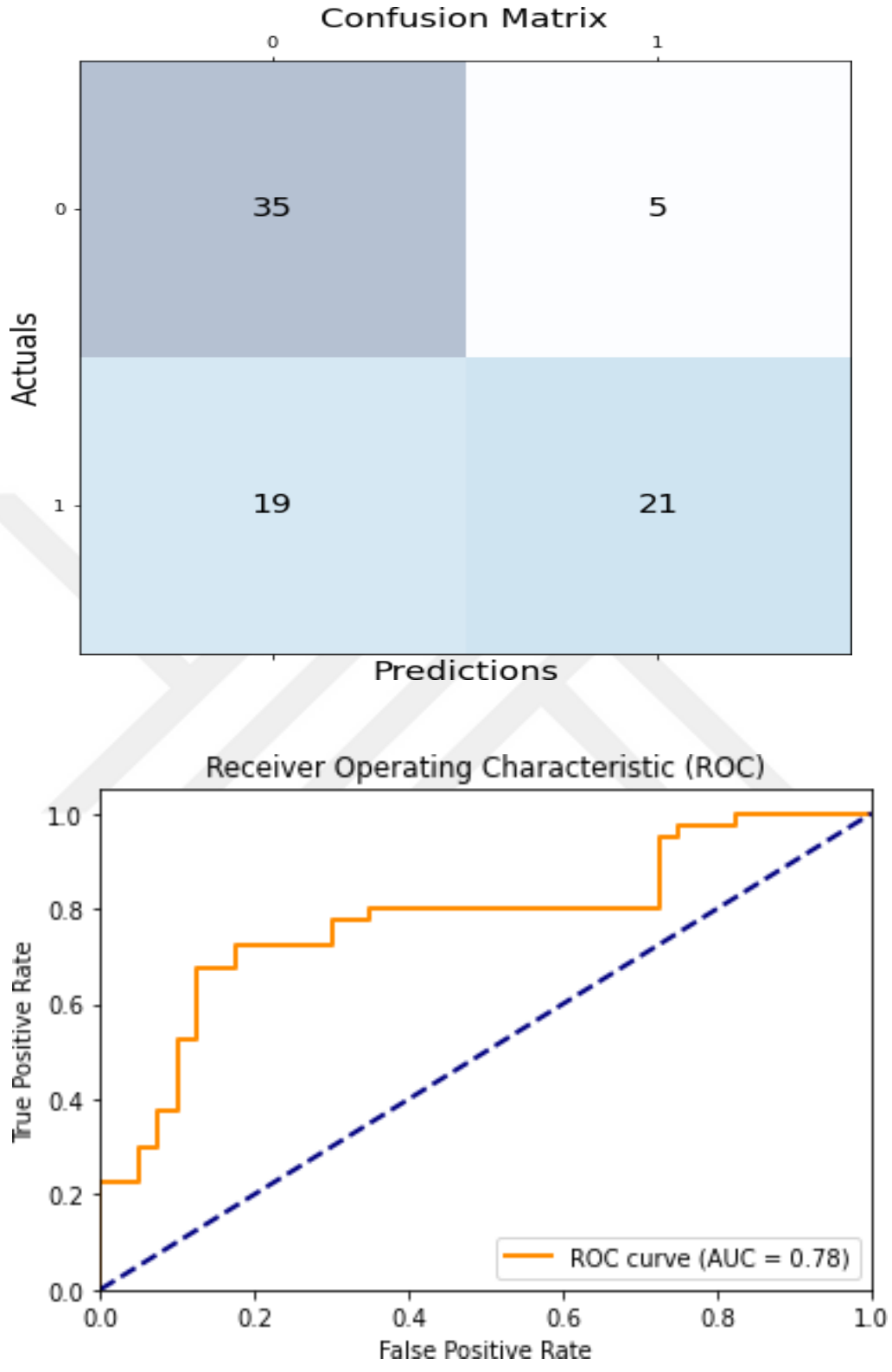

Figure 4.9. The Confusion Matrix and ROC curve function for 2-class classification utilizing a 2-fold-cross validation split

Table 4.4. The other Metrics of 2- class classification using a 1-fold-cross validation split.

	precision	recall	f1-score	support
0.0	0.65	0.88	0.74	40
1.0	0.81	0.53	0.64	40
accuracy			0.70	80
macro avg	0.73	0.70	0.69	80
weighted avg	0.73	0.70	0.69	80

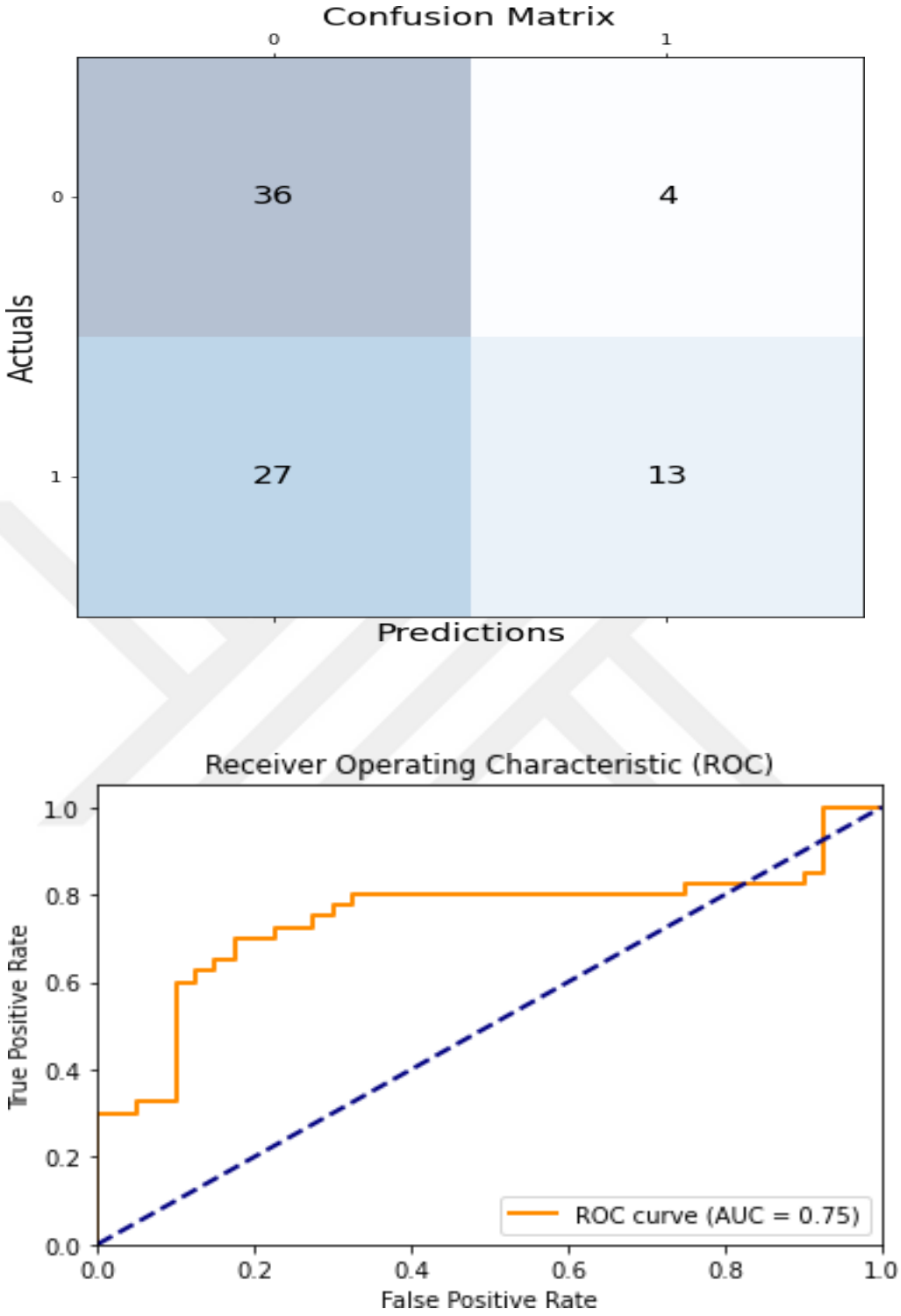

Figure 4.10. The Confusion Matrix and ROC curve function for two-class categories utilize a 2-fold-cross validation split

Table 4.5. The additional metrics for two-class categories employing a split for cross-validation of two folds

	precision	recall	f1-score	support
0.0	0.57	0.90	0.70	40
1.0	0.76	0.33	0.46	40
accuracy			0.61	80
macro avg	0.67	0.61	0.58	80
weighted avg	0.67	0.61	0.58	80

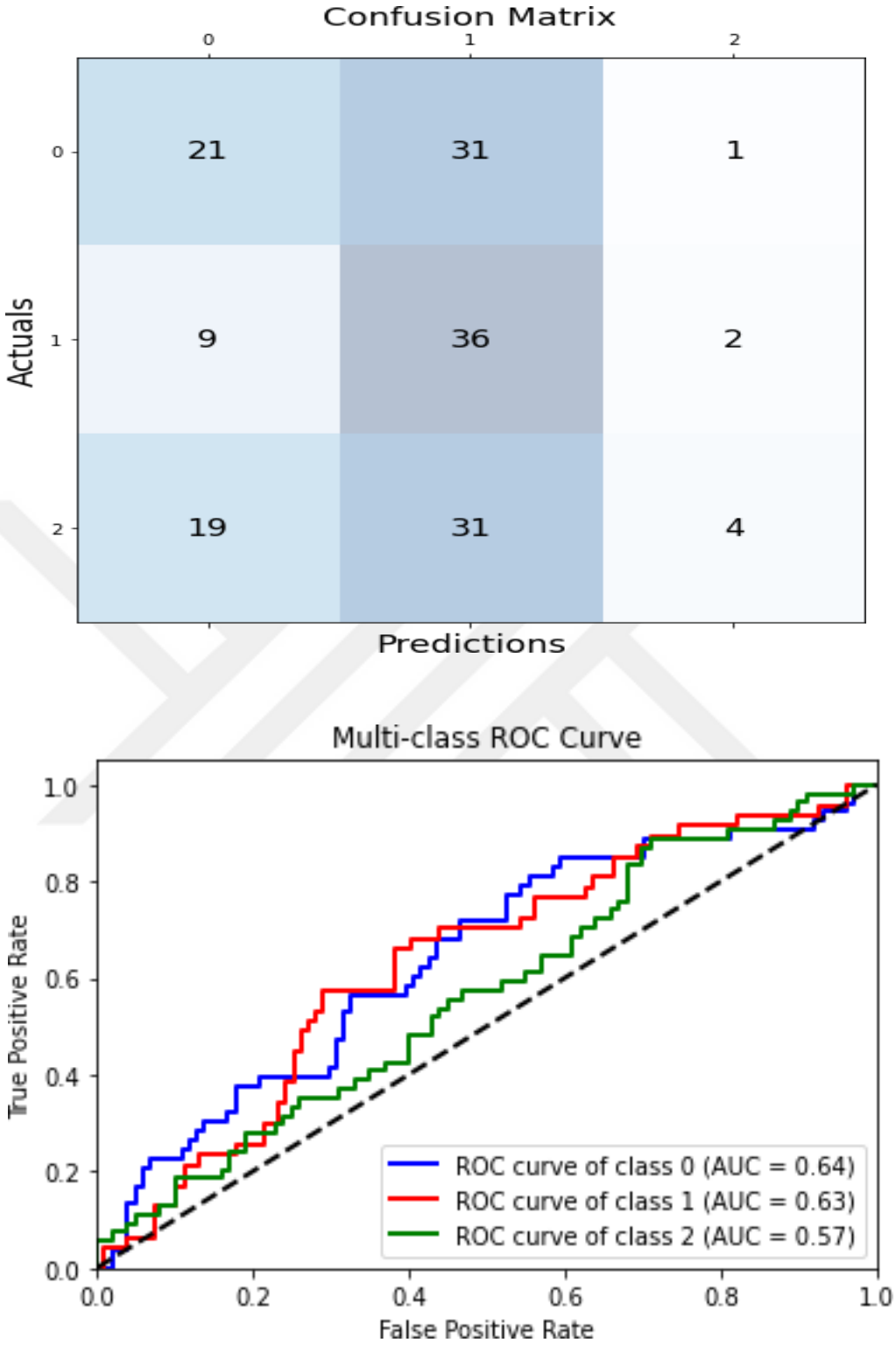

Figure 4.11. The Confusion Matrix and ROC curve function for three-class categorization utilizing an 80/20 split

Table 4.6. The additional metrics for three-class categorization utilizing an 80/20 split.

	precision	recall	f1-score	support
0.0	0.43	0.40	0.41	53
1.0	0.37	0.77	0.50	47
2.0	0.57	0.07	0.13	54
accuracy			0.40	154
macro avg	0.46	0.41	0.35	154
weighted avg	0.46	0.40	0.34	154

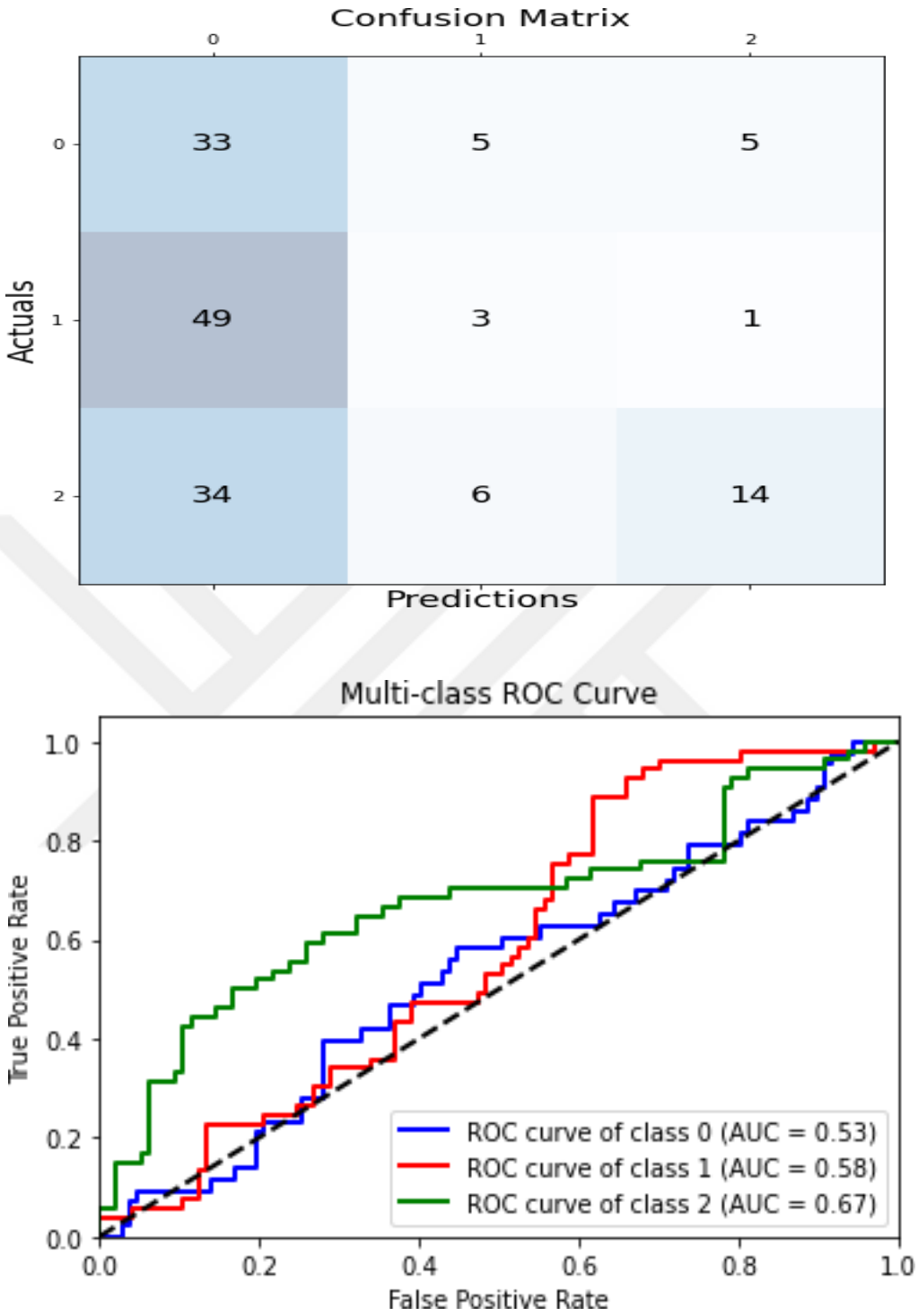

Figure 4.12. The Confusion Matrix and ROC curve function for 3-class classification using a 1-fold-cross validation split

Table 4.7. The other Metrics of 3- class classification utilizing a one-fold-cross validation split

	precision	recall	f1-score	support
0.0	0.28	0.77	0.42	43
1.0	0.21	0.06	0.09	53
2.0	0.70	0.26	0.38	54
accuracy			0.33	150
macro avg	0.40	0.36	0.29	150
weighted avg	0.41	0.33	0.29	150

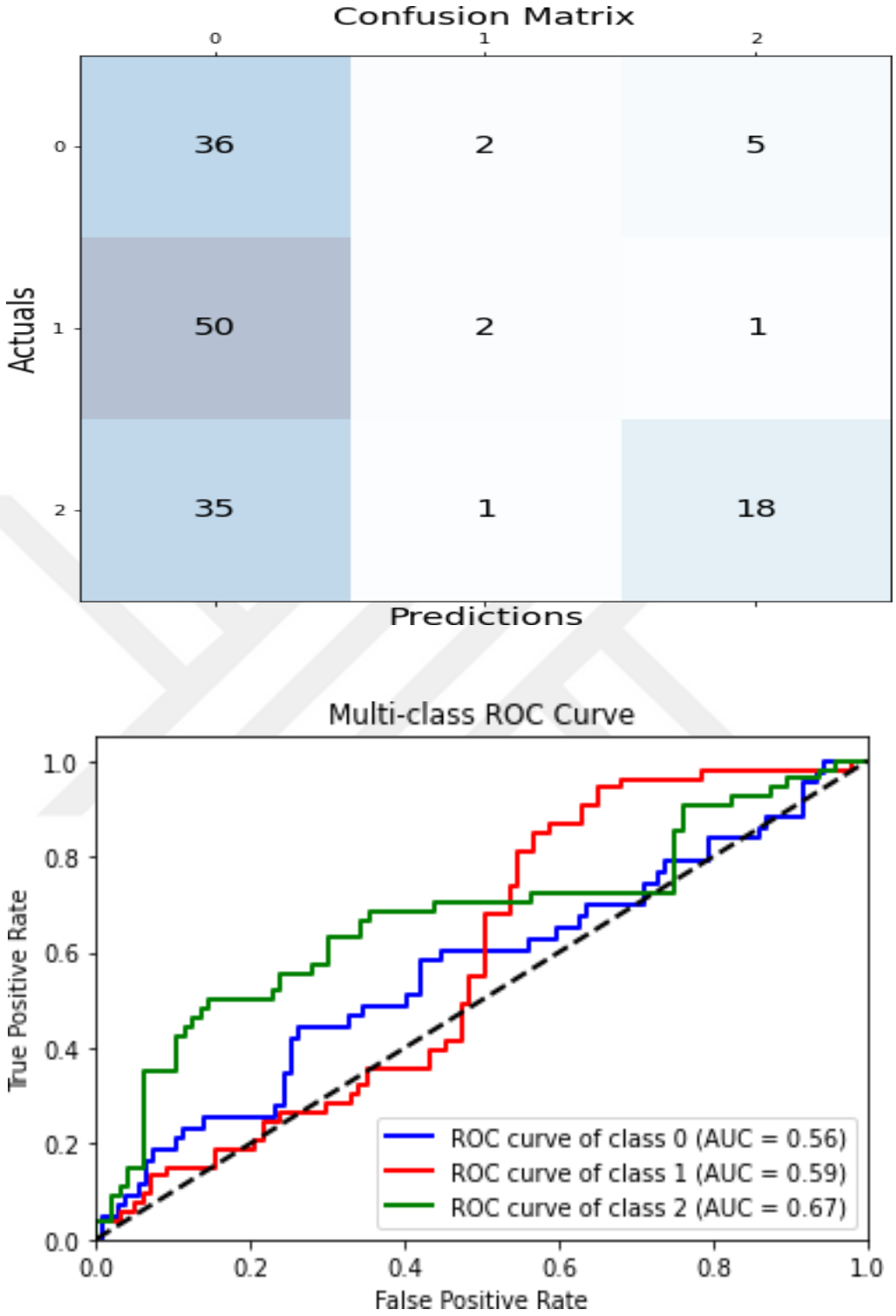

Figure 4.13 The Confusion Matrix and ROC curve function for 3-class categorization utilizing a 2-fold-cross validation split

Table 4.8. The other Metrics of 3- class categorization utilizing a 2-fold-cross validation split

	precision	recall	f1-score	support
0.0	0.30	0.84	0.44	43
1.0	0.40	0.04	0.07	53
2.0	0.75	0.33	0.46	54
accuracy			0.37	150
macro avg	0.48	0.40	0.32	150
weighted avg	0.50	0.37	0.32	150

The comparison of our study is made by referring to other studies in Table 4.5. For instance, in the study made by (Perna, 2019), CNN is used. They reached an accuracy rate of 83.00% for 2-class classification and %82.00 for 3-class classification. In the study made by (Kok et al., 2019), an accuracy rate of 87.50% is achieved using MFCC features. Our study reached an accuracy rate 80% for 2-class classification and an accuracy rate of 55% for 3-class.

Table 4.9. The Comparison of my research with other Studies

Different study and method	No.class	The Class type (target)	(ACC)
(Perna, 2019) CNN	2	The unhealthy with healthy	83.00%
	3	healthy, chronic and nonchronic diseases	82.00%
(Kok et al., 2019), MFCC Features-Based Classification	2	(healthy and diseased)	87.50%
(Serbes et al., 2018), STFT, WT, SVM	4	Wheeze, Crackle, wheeze plus crackle and Normal	49.86%
(N. Jakovljević and T. Lončar-Turukalo, 2018) hidden Markov models with Gaussian mixture models, MFCC	3	normal, wheeze and crackle	39.56%
(Demir et al., 2020), used a combination of deep features and (CNN) with parallel-poling technique and classifier method of (LDA-RSE).	4	Normal, Wheezes, crackles, wheezes+ crackles.	71.15%
(Jung et al., 2021), a DS-CNN model was used with two features combined (STFT and MFCC)	4	The discontinuous, Normal, Continuous, and Unknown	85.74%
(Nguyen and Pernkopf, 2020), The snapshot Ensemble of the CNN.	4	(Normal, crackles, wheezes, and both crackles and wheezes)	78.4%
	2	Abnormal (crackles, wheezes, and both crackles with wheezes) and Normal	83.7%

Table 4.9 (continue)

(Petmezas et al., 2022), CNN-LSTM with function selection (FS), and (LOOCV)	4	Normal, Wheezes, Crackles, and Both wheezes and crackles.	(76.39%), And (74.57%)
(Aykanat et al., 2017), CNN and SVM	3	Crackle(rale), rhonchus, and normal sound classification	76% and 75%
(ARI et al., 2022) WST, ELM-W-AE method, SVM and (EBT) classification respectively.	4	normal, wheezing, wheezing, wheezing, and wheezing	%72.69 %60.61
(Romero Gómez and Orjuela-Cañón, 2021) 13 MFCCs coefficients STF and LTF parameters, with SVM, ANN and RF model respectively	2		0.721 0.775 0.782
	4		0.721 0.687 0.687
(Chambres et al., 2018) Decision tree, Lower-level feature			46.62
My Method	2	(Healthy vis unhealthy)	80%
	3	(Healthy, chronic, and non-chronic)	55%

4.2 Discussions

In this study, our proposed method achieved an accuracy rate reached 80% for 2-class classification and an accuracy rate of 55% for 3-class. So, our study more suitable in case 2 class classification because of good accuracy rate and novel method. Besides that, it outperformed most of the other studies in the literature in terms of accuracy, so, our study has better performance than others existing methods in comparison for lung sound signal classification. The study showed that denoising using bandpass filter helped to reduce the noise in the lung sound signals, which improved the accuracy of classification. The CS-MDWTD method decomposed the signals into different frequency sub-bands, which provided more informative features for the task of categorization. The study showed that the CNN-LSTM hybrid model effectively learned the features extracted by the CS-MDWTD method and was able to classify the lung sounds accurately. The hybrid model was able to capture both the spatial and temporal features of the signals, which improved the accuracy of classification. Automated categorization of lung audio signals plays a crucial role in the early identification of respiratory conditions, which can enhance patient care and quality of life. The proposed method can also be used in remote patient monitoring, where patients can use wearable devices to record their lung sound signals and transmit them to healthcare professionals for diagnosis and treatment. So, our approach is capable of diagnosis, severity categorization and disease classification. It can classify long-term audio signals belonging to the lungs, although, most of the studies in the literature evaluated short-term audio signals (frames). So, a preprocessing step is not needed, but the memory space is important. The drawback of our method is the amount of memory it needs to work, but it can be omitted for most of the computers in this area. Over all, the study shows that the suggested method is effective in classifying lung sound signals and has potential clinical implications. There are some researchers name as mentioned in the literature review got a bit higher accuracy rate than our method but our research method also has good accuracy especially in case 2- class classification to compare with others research method.

5. CONCLUSIONS and RECOMMENDATIONS

5.1. Conclusion

The hybrid model named CNN-LSTM is utilized in this research to assess the suggested CS-MDWTD approach on the ICBHI dataset. The study exposed that denoising using a bandpass filter helped reduce the noise in the lung sound signals, improving the classification accuracy. And the CS-MDWTD method decomposed the signals into different frequency sub-bands, which provided more informative features for the classification task. Also, the CNN-LSTM hybrid model effectively learned the features and was able to classify the lung sounds accurate. So, we showed that CS can be used together with MDWTD in order to increase performance. We proposed a distinct method by comparing detail coefficients according to similarity. The vector including the most similar detail coefficient has opted in this method. Hence, we could capture multi-level similar feature sets.

The experimental results demonstrate that the technique that has been suggested can conduct diagnosis and classification; hence, it may be used as a support tool in medical clinics. Based on this, a combination of method with model, which provides the highest accuracy of lung, which is 80% for the case 2- class classification of splitting as proposed. But with different accuracy in K-Fold Cross Validation. Overall, the proposed approach has the potential to be used in clinical practice for the automated diagnosis of lung diseases based on lung sound signals. However, further research is needed to validate the proposed method on a larger dataset, with more divers set of lung sounds, with a larger sample size and compare it with other existing methods, to assess its effectiveness and performance in comparison to other methods. So, our study more suitable in case 2 class classification because of high accuracy rate and used novel method. It is advised to use deep feature analysis with DWT for future study. In the case of classification in particular, hybrid approaches can produce higher accuracy rates.

5.2. Recommendations

In the future, we want to evaluate the effectiveness of the suggested strategy on fresh data from different databases and play around with various features or potential feature combinations. And we suggest being used our proposed method to further optimized by exploring different denoising techniques, feature extraction methods, and classification algorithms. The other recommendations are the evaluation of the methods and investigation of their real-world applications that would be valuable in understanding their full potential and how they can be effectively unified into clinical practice and then comparing the obtained results with other approaches, which would provide a complete understanding of the strengths and limitations of the proposed method. Also, in future studies, authors could use transfer learning, and further researchers can use many machine learning metrics using the new methods and new models or the same as our model for evaluations, more scientific challenges and comparisons with the other research results.

REFERENCES

AL-ANZI, F. S. and ABUZEINA, D. 2017. Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. *Journal of King Saud University - Computer and Information Sciences*, *29*(2), 189–195. https://doi.org/10.1016/j.jksuci.2016.04.001

AL-OTAIBI, S., ALTWOIJRY, N., ALQAHTANI, A., ALDHEEM, L., ALQHATANI, M., ALSURAIBY, N., ALSAIF, S. and ALBARRAK, S. 2022. Cosine similarity-based algorithm for social networking recommendation. *International Journal of Electrical and Computer Engineering*, *12*(2), 1881–1892. https://doi.org/10.11591/ijece.v12i2.pp1881-1892

ALTAN, G., YAYIK, A. and KUTLU, Y. 2021. Deep Learning with ConvNet Predicts Imagery Tasks Through EEG. *Neural Processing Letters*, *53*(4), 2917–2932. https://doi.org/10.1007/s11063-021-10533-7

ARI, B., ALÇİN, Ö. F. and ŞENGÜR, A. 2022. A Lung Sound Classification System Based on Data Augmenting Using ELM-Wavelet-AE. Turkish Journal of Science and Technology, 17(1), 79–88. https://doi.org/10.55525/tjst.1063039

AYKANAT, M., KILIÇ, Ö., KURT, B. and SARYAL, S. 2017. Classification of lung sounds using convolutional neural networks. *Eurasip Journal on Image and Video Processing*, *2017*(1). https://doi.org/10.1186/s13640-017-0213-2

BAO, T., ZAIDI, S. A. R., XIE, S., YANG, P. and ZHANG, Z. Q. 2021. A CNN-LSTM Hybrid Model for Wrist Kinematics Estimation Using Surface Electromyography. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–9. https://doi.org/10.1109/TIM.2020.3036654

BENGIO, Y. 2009. Learning deep architectures for AI. In *Foundations and Trends in Machine Learning* (Vol. 2, Issue 1). https://doi.org/10.1561/2200000006

CAMPS-VALLS, G., TUIA, D. and BRUZZONE, L. 2013. Advances in hyperspectral image classification. IEEE Signal Processing Magazine, January, 45–54.

CHAMBRES, G., HANNA, P. and DESAINTE-CATHERINE, M. 2018. Automatic detection of patient with respiratory diseases using lung sound analysis. Proceedings - International Workshop on Content-Based Multimedia Indexing, 2018-Septe. https://doi.org/10.1109/CBMI.2018.8516489

CHANG, G. C. and CHENG, Y. P.2008. Investigation of noise effect on lung sound recognition. *Proceedings of the 7th International Conference on Machine Learning and Cybernetics, ICMLC*, *3*(July), 1298–1301. https://doi.org/10.1109/ICMLC.2008.4620605

CHANG, G. C. and LAI, Y. F.2010. Performance evaluation and enhancement of lung sound recognition system in two real noisy environments. *Computer Methods and Programs in Biomedicine*, *97*(2), 141–150. https://doi.org/10.1016/j.cmpb.2009.06.002

CHAOVALIT, P., GANGOPADHYAY, A., KARABATIS, G. and CHEN, Z.2011. Discrete wavelet transform-based time series analysis and mining. *ACM Computing Surveys*, *43*(2). https://doi.org/10.1145/1883612.1883613

CHOLLET, F.2017. Deep Learning with Python. In Manning Publications Co. ACCESSED 2022-Nov.-08.

DAI, H., HUANG, G., WANG, J., ZENG, H. and ZHOU, F.2021. Prediction of air pollutant concentration based on one-dimensional multi-scale cnn-lstm considering spatial-temporal characteristics: A case study of Xi'an, China.

Atmosphere, *12*(12). https://doi.org/10.3390/atmos12121626

DEMIR, F., ISMAEL, A. M. and SENGUR, A. 2020. Classification of Lung Sounds with CNN Model Using Parallel Pooling Structure. *IEEE Access*, *8*, 105376–105383. https://doi.org/10.1109/ACCESS.2020.3000111

DEMIR, F., SENGUR, A. and BAJAJ, V.2020. Convolutional neural networks based efficient approach for classification of lung diseases. *Health Information Science and Systems*, *8*(1). https://doi.org/10.1007/s13755-019-0091-3

DENG, F., CHEN, Z., LIU, Y., YANG, S., HAO, R. and LYU, L.2022. A Novel Combination Neural Network Based on ConvLSTM-Transformer for Bearing Remaining Useful Life Prediction. *Machines*, *10*(12). https://doi.org/10.3390/machines10121226

ELMAN, J. L. 1990. Finding structure in time. *Cognitive Science*, *14*(2), 179–211. https://doi.org/10.1016/0364-0213(90)90002-E

FORTUNE, N. A. 2018. A Short Guide to Using Python For Data Analysis In Experimental Physics. In *Authorea Preprints.* Python Scipy Butterworth Filter - Python Guides (Accessed:18 august 2022)

FRAIWAN, L., HASSANIN, O., FRAIWAN, M., KHASSAWNEH, B., IBNIAN, A. M. and ALKHODARI, M.2021. Automatic identification of respiratory diseases from stethoscopic lung sound signals using ensemble classifiers. *Biocybernetics and Biomedical Engineering*, *41*(1), 1–14. https://doi.org/10.1016/j.bbe.2020.11.003

GAIROLA, S., TOM, F., KWATRA, N. and JAIN, M.2021. RespireNet: A Deep Neural Network for Accurately Detecting Abnormal Lung Sounds in Limited Data Setting. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 527–530. https://doi.org/10.1109/EMBC46164.2021.9630091

GARCÍA, S., LUENGO, J. and HERRERA, F.2015. Data Preprocessing in Data Mining. In *Intelligent Systems Reference Library* (Vol. 72).

GERS, F. A., SCHMIDHUBER, J. and CUMMINS, F.2000. Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

HASSAN, S. M., MAJI, A. K., JASIŃSKI, M., LEONOWICZ, Z. and JASIŃSKA, E. 2021. Identification of plant-leaf diseas[1] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of plant-leaf diseases using cnn and transfer-learning approach," Electron., vol. 10, no. 12, 2021, doi: 10.3390/electronics101213. *Electronics (Switzerland)*, *10*(12).

HOCHREITER, S. and SCHMIDHUBER, J.1997. Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

JAYALAKSHMY, S. and SUDHA, G. F.2021. Conditional GAN based augmentation for predictive modeling of respiratory signals. *Elsevier Ltd*, *October*, 11. https://doi.org/https://doi.org/10.1016/j.compbiomed.2021.104930

JOSE, S.2001. *Concept Decompositions for Large Sparse Text Data.pdf*. 143–175.

JUNG, S. Y., LIAO, C. H., WU, Y. S., YUAN, S. M. and SUN, C. T.2021. Efficiently classifying lung sounds through depthwise separable cnn models with fused stft and mfcc features. *Diagnostics*, *11*(4). https://doi.org/10.3390/DIAGNOSTICS11040732

KAMBLE, H. S. and PHADKE, G. S. 2016. *Respiratory Sound Analysis for the Diagnosis of Respiratory Abnormalities*. 113–123.

KHAN, S. I. and PACHORI, R. B.2021. Automated classification of lung sound signals based on empirical mode decomposition. *Expert Systems with Applications*, *184*(June), 115456. https://doi.org/10.1016/j.eswa.2021.115456

KIM, Y., HYON, Y. K., JUNG, S. S., LEE, S., YOO, G., CHUNG, C. and HA, T. 2021. Respiratory sound classification for crackles, wheezes, and rhonchi in the clinical field using deep learning. *Scientific Reports*, *11*(1), 1–11. https://doi.org/10.1038/s41598-021-96724-7

KOK, X. H., ANAS IMTIAZ, S. and RODRIGUEZ-VILLEGAS, E. 2019. A Novel Method for Automatic Identification of Respiratory Disease from Acoustic Recordings. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2589–2592. https://doi.org/10.1109/EMBC.2019.8857154

LECUN, Y., BENGIO, Y. and HINTON, G. 2015. Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

LI, C., DU, H. and ZHU, B. 2020. *Classification of Lung Sounds Using Classification of lung sounds using CNN- Attention*.

LYNCH, S. 2018. Image Processing with Python. *Dynamical Systems with Applications Using Python*, 471–489. https://doi.org/10.1007/978-3-319-78145-7_18. Python Scipy Butterworth Filter - Python Guides (Accessed:18 august 2022)

MECHETER, I., ABBOD, M., ZAIDI, H. and AMIRA, A. 2022. Brain MR images segmentation using 3D CNN with features recalibration mechanism for segmented CT generation. *Neurocomputing*, *491*, 232–243. https://doi.org/10.1016/j.neucom.2022.03.039

MOUTON, C., MYBURGH, J. C. and DAVEL, M. H. 2020. Stride and Translation Invariance in CNNs. *Communications in Computer and Information Science*, *1342*, 267–281. https://doi.org/10.1007/978-3-030-66151-9_17

NGUYEN, T. and PERNKOPF, F. 2020. Lung Sound Classification Using Snapshot Ensemble of Convolutional Neural Networks. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, *2020-July*, 760–763. https://doi.org/10.1109/EMBC44109.2020.9176076

NGUYEN, T. and PERNKOPF, F. 2022. Lung Sound Classification Using Co-Tuning and Stochastic Normalization. *IEEE Transactions on Biomedical Engineering*, *69*(9), 2872–2882. https://doi.org/10.1109/TBME.2022.3156293

NI, D. and MA, H. 2015. Hyperspectral Image Classification via Sparse Code Histogram. IEEE Geoscience and Remote Sensing Letters, 12(9), 1843–1847. https://doi.org/10.1109/LGRS.2015.2430871

N. JAKOVLJEVIĆ and T. LONČAR-TURUKALO. 2018. Hidden Markov Model Based Respiratory Sound Classification (Vol. 66, Issue November). https://doi.org/https://doi.org/10.1007/978-981-10-7419-6_7

OHATA, E. F., BEZERRA, G. M., CHAGAS, J. V. S. DAS, LIRA NETO, A. V., ALBUQUERQUE, A. B., ALBUQUERQUE, V. H. C. D. and REBOUCAS FILHO, P. P. 2021. Automatic detection of COVID-19 infection using chest X-ray images through transfer learning. *IEEE/CAA Journal of Automatica Sinica*, *8*(1), 239–248. https://doi.org/10.1109/JAS.2020.1003393

PARK, K., HONG, J. S. and KIM, W. 2020. A Methodology Combining Cosine Similarity with Classifier for Text Classification. *Applied Artificial Intelligence*,

34(5), 396–411. https://doi.org/10.1080/08839514.2020.1723868

PATEL, R. 2013. Design Technique of Bandpass FIR filter using Various Window Function. *IOSR Journal of Electronics and Communication Engineering*, *6*(6), 52–57. https://doi.org/10.9790/2834-0665257

PEDRYCZ, W. and CHEN, S. 2020. *Deep Learning Algorithms And Applications by Witold Pedrycz, Shyi-Ming Chen (z-lib.org).pdf*. Accessed: 08-09-2022

PERNA, D. 2019. Convolutional Neural Networks Learning from Respiratory data. *Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018*, 2109–2113. https://doi.org/10.1109/BIBM.2018.8621273

PETMEZAS, G., CHEIMARIOTIS, G. A., STEFANOPOULOS, L., ROCHA, B., PAIVA, R. P., KATSAGGELOS, A. K. and MAGLAVERAS, N. 2022. Automated Lung Sound Classification Using a Hybrid CNN-LSTM Network and Focal Loss Function. *Sensors*, *22*(3). https://doi.org/10.3390/s22031232

PHAM, L., PHAN, H., SCHINDLER, A., KING, R., MERTINS, A. and MCLOUGHLIN, I. 2021. Inception-Based Network and Multi-Spectrogram Ensemble Applied to Predict Respiratory Anomalies and Lung Diseases. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 253–256. https://doi.org/10.1109/EMBC46164.2021.9629857

PLAZA, A., BENEDIKTSSON, J. A., BOARDMAN, J. W., BRAZILE, J., BRUZZONE, L., CAMPS-VALLS, G., CHANUSSOT, J., FAUVEL, M., GAMBA, P., GUALTIERI, A., MARCONCINI, M., TILTON, J. C. and TRIANNI, G. 2009. Recent advances in techniques for hyperspectral image processing. *Remote Sensing of Environment*, *113*(SUPPL. 1), S110–S122. https://doi.org/10.1016/j.rse.2007.07.028

PRA, Y. DE and FONTANA, F. 2020. *DEVELOPMENT OF REAL-TIME AUDIO APPLICATIONS USING PYTHON DEVELOPMENT OF REAL-TIME AUDIO APPLICATIONS USING*. *March*. Python Scipy Butterworth Filter - Python Guides (Accessed:18 august 2022)

QAYYUM, A., MERIAUDEAU, F. and CHAN, G. C. Y. 2019. Classification of atrial fibrillation with pretrained convolutional neural network models. *2018 IEEE EMBS Conference on Biomedical Engineering and Sciences, IECBES 2018 - Proceedings*, *February 2019*, 594–599. https://doi.org/10.1109/IECBES.2018.8626624

RAHAMAN, K. J. and RASEL, M. 2021. *An Analytical Evaluation of a Deep Learning Model to Detect Network Intrusion An Analytical Evaluation of a Deep Learning Model to Detect Network* (Issue July). Springer International Publishing. https://doi.org/10.1007/978-3-030-80253-0

REICHERT, S., GASS, R., BRANDT, C. and ANDRÈS, E. 2008. Analysis of Respiratory Sounds: State of the Art. Clinical Medicine. Circulatory, Respiratory and Pulmonary Medicine, 2, CCRPM.S530. https://doi.org/10.4137/ccrpm.s530

ROMERO GÓMEZ, A. F. and ORJUELA-CAÑÓN, A. D. 2021. Respiratory Sounds Classification employing a Multi-label Approach. 2021 IEEE Colombian Conference on Applications of Computational Intelligence, ColCACI 2021 - Proceedings, July. https://doi.org/10.1109/ColCACI52978.2021.9469042

SERBES, G., ULUKAYA, S. and KAHYA, Y. P. 2018. An automated lung sound

preprocessing and classification system based onspectral analysis methods. *IFMBE Proceedings*, *66*(April), 45–49. https://doi.org/10.1007/978-981-10-7419-6_8

SHUVO, S. B., ALI, S. N., SWAPNIL, S. I., HASAN, T. and BHUIYAN, M. I. H. 2021. A Lightweight CNN Model for Detecting Respiratory Diseases from Lung Auscultation Sounds Using EMD-CWT-Based Hybrid Scalogram. *IEEE Journal of Biomedical and Health Informatics*, *25*(7), 2595–2603. https://doi.org/10.1109/JBHI.2020.3048006

TANG, Y. X., TANG, Y. B., PENG, Y., YAN, K., BAGHERI, M., REDD, B. A., BRANDON, C. J., LU, Z., HAN, M., XIAO, J. and SUMMERS, R. M. 2020. Automated abnormality classification of chest radiographs using deep convolutional neural networks. *Npj Digital Medicine*, *3*(1). https://doi.org/10.1038/s41746-020-0273-z

TSAI, C. F., HSU, Y. F., LIN, C. Y. and LIN, W. Y. 2009. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, *36*(10), 11994–12000. https://doi.org/10.1016/j.eswa.2009.05.029

VICTOR, G. S., ANTONIA, P. and SPYROS, S. 2014. CSMR: A scalable algorithm for text clustering with cosine similarity and MapReduce. *IFIP Advances in Information and Communication Technology*, *437*, 211–220. https://doi.org/10.1007/978-3-662-44722-2_23

WORLD HEALTH ORGANIZATION. 2016. WORLD HEALTH STATISTICS - MONITORING HEALTH FOR THE SDGs. *World Health Organization*, 1.121.

ZHANG, W., ITOH, K., TANIDA, J. and ICHIOKA, Y. 1990. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied Optics*, *29*(32), 4790. https://doi.org/10.1364/ao.29.00479