Machine learning based approaches for cough detection from acceleration signal

Ines Belhaj Messaoud[2], Elyes Ben Cheikh[1], Assaad Chiboub[1], Karim Loulou[3], Youssef Ouakrim[3], Sofia Ben Jebara[1], Neila Mezghani[3]

[1]Université de Carthage, École Supérieure des communications de Tunis, Tunis, Tunisia
[2]Université Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis,Tunis, Tunisia
[3]Laboratoire de recherche en Imagerie et orthopedie (LIO), CRCHUM, TELUQ university, Montreal, Canada
[1]Laboratoire de recherche COSIM, Sup'Com, Tunis, Tunisia
e-mail: inesbelhajj@gmail.com, elyes.bencheikh@supcom.tn, assaad.chiboub@supcom.tn, karim.loulou@outlook.com, oua.youssef@gmail.com, sofia.benjebara@supcom.tn, neila.mezghani@teluq.ca

Abstract –

The main goal of this research is to develop a a machine learning based method in order to detect cough from acceleration signals. In this study, two different methods are proposed: a conventional one that uses Xgboost as a classifier and a deep learning which uses CNN-1D as an architecture. We found that these models were able to distinguish between acceleration signals caused by coughing and acceleration signals caused by other activities such as clearing throat, talking, laughing and movements in different directions with high accuracy. This study affirms that cough monitoring based on accelerometer measurements generated by the Hexoskin device is possible, making it a new user-independent tool of cough detection.

***Keywords* – Cough detection, connected textile sensors, acceleration signals, supervised classification, machine learning, deep learning.**

I. Introduction

Cough can be described as a sudden and often repetitive expulsion of air with a strong expiratory effort. It is an important indicator of several health problems such as upper and lower respiratory tract infections (common cold, influenza, bronchitis bronchiolitis), asthma, chronic obstructive pulmonary, pulmonary edema, pneumonia, tuberculosis,... and recently COVID-19 [1].

Cough detection can be useful in a variety of purposes. Some potential applications include monitoring respiratory health (alert to infection presence of a chronic respiratory condition for instance), early detection of respiratory infections (useful in situations where it is not possible or desirable to perform more invasive diagnostic tests), and remote monitoring (using smart watches or fitness trackers, allowing individuals to monitor their own respiratory health on a continuous basis), surveillance in public settings (identifying individuals who may be infected with a respiratory illness and alert them to the need for further testing or isolation), improving public health (detection and tracking the spread of respiratory infections)...

There are several ways to detect coughs when using wearable devices and the Internet of Things [2]. Coughs can be detected using either audio signals or physiological signals (such as heart activity, blood pressure, airflow, or respiratory inhalation and expiration, movement signals,...). In this work, we are interested with acceleration signals that detect the sudden and forceful movement associated with a cough. In fact, monitoring cough based on signals from an accelerometer placed on a human body is less intrusive and it was proven to be an efficient way of detection [3]. It was also proven that the accelerometer could be used with other physiological sensors such as ECG and respiration sensors [4].

Automatic cough detection can be carried using machine learning. It involves training a machine learning model on a dataset which will be able to predict whether a measured information contains cough or not. Either conventional machine learning or deep learning can be used. Conventional machine leaning algorithms (such as decision trees, random forests, and support vector machines) can learn patterns in the data and make predictions based on those patterns. Deep learning, on the other hand, involves the use of neural networks composed of multiple layers of interconnected nodes which can learn complex patterns in the data by adjusting the connections between nodes. In this research, a machine learning based method using Extreme Gradient Boosting (XGBoost) as well as a deep learner Convolutional Neural Network (CNN) based model are proposed to detect the symptom of cough by using only accelerometer raw data. This measurement tool is attached to a human body and the acceleration is 3 dimensional (X, Y and Z axis) is acquired to constitute data of interest.

Dealing with literature review, one of the recent related works proposed deep learning methods to perform cough detection based on measurements from an accelerometer attached to the patient's bed [5] . This study approach offers cough monitoring for patients in their own bed without having to move or attach an accelerometer in their own body. The used classifiers are based on CNN, LSTM residual neural network (Resnet50). In another related work [6], an automatic cough detection system was considered to differentiate between acceleration signals associated with coughs and those associated with swallows, tongue movements and other activities such as speech. The raw acceleration data was represented in term of time-frequency meta features, a binary genetic algorithm was used for feature selection and a support vector machine (SVM) classifier was used.

In this work, we deal with a deep learner architecture based on Convolutional neural networks (CNNs), along with the XGBoost machine learning model, to explore the detection of cough solely based on acceleration signals. These signals were obtained in a non invasive way using the Hexoskin device. Our specific objective was to know whether the body movements associated with coughing exhibit distinctive characteristics compared to other activities like throat clearing, sniffling, speaking, laughing, etc...

The paper is organized as follows. Section II deals with data collection procedure and preparation to make it ready for classification. Section III gives an overview of selected and adjusted deep learning based architectures and machine learning model. Section IV adresses the classification results methodology and results. Finally section V draws some conclusions.

II. Data preparation

A. Data collection

This project is submitted to the research ethics committees of University of Montreal Hospital Center (CHUM), École de technologie supérieure (ÉTS) and TELUQ University. The fundamental ethical principles as well as the guidelines of the councils are respected during the project.

In this work, a total of 36 participants, men and women aged from18 to 65, were recruited internally, at the Laboratory of Simulation and Modeling of Motion (S2M) of the University of Montreal or in the offices of Carré Technologies, Inc. which is the provider of equipments acquisition [1]. Participants were in good health and had no chronic pain or known respiratory problems. Pregnant women were not to be recruited for this study.

The day of data acquisition, each participant weared the Hexoskin device in order to allow acquisition of some physiological signals [2]. Among them, we mainly cite the , 3-axis acceleration signals which are the signals of interest in this study. The participants were asked to perform the following 11 consecutive activities in different positions : sitting position with 30-degree inclination, sitting position with 90-degree inclination and lying position. Each session for one particular position lasts between 10 and15 minutes. A period of 2-5 minutes of rest between each position was also considered. The 11 consecutive activities are :

- 2-3 minutes normal respiration without coughing
- 5 normal volume coughs
- 5 double coughs (2 consecutive coughs repeated 5 times)
- 5 throat clears
- 5 low volume coughs
- Laugh during 2-5 seconds
- 5 normal volume coughs
- speaking loudly 3 words (from the Harvard corpus)
- 5 loud coughs
- 5 cycles of deep breathing (inhalation through the nose / exhalation through the mouth)

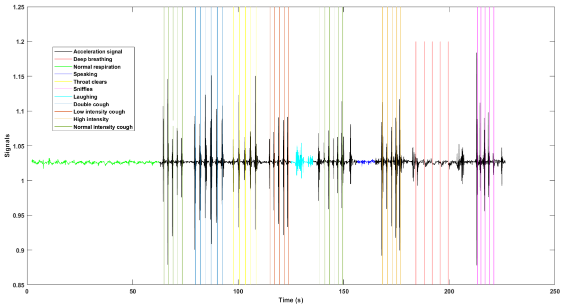

Fig. 1. Illustration of the acceleration signal during the experimental protocol activities.

- 5 sniffles

The 3-axis acceleration signals are acquired and sampled at $f_s = 64\,\text{Hz}$. Computing the modulus of the triple components is a simple solution to reduce the data dimension by 3 and also to make data orientation independent. Let A_x, A_y and A_z the accelerations along respectively the x, y and z axes and G the gravity constant. The norm A of the acceleration vector is defined as : $A = \sqrt{A_x^2 + A_y^2 + A_z^2}/G$.

Fig. 1 illustrates an acceleration norm signal example of one participant. The different human body activities are plotted in different colors either in the acceleration signal or using vertical lines indicating the activity instance of occurrence. The objective is to highlight the differences in the acceleration signal according to the activity. For example, the signal looks like an noise when no physical activity occurs. During the coughing exercise, the acceleration signal appears as a puff with an ascending session followed by a descending one. A peak arises representing the maximal velocity. Some other peaks appear also during other activities such as laughing but the shape of the activity differs from that of cough. On the other hand, sniffling and throat clearing look like coughing and contrarily, speaking does not have effect of acceleration.

To conclude, this particular representation highlights the importance to rigorously segregate the cough from the other activities in the acceleration signal by taking into account the morphology of each activity acceleration.

B. Data labeling

During acquisition, the original data was not completely labeled, which means that cough frames are not precisely identified. The only available information is a flag indicating the presence of the cough (vertical lines in Fi.g 1). This flag was raised manually by the user at the instant when he intented to cough. To label the acceleration data, the audio data was used as a reference and segmented based on the presence or absence of silence in the signal. The minimum duration of silence was set to 350 ms and the upper limit of the silence level in full scale decibels (dBFS) was set to -55 dBFS. From these segments, the active parts of the audio were determined, which are the parts located between two silent segments.

To determine the type of activity associated with each segment, a portion of the dataset was manually labeled for supervised learning. The coughing and throat clearing activities were grouped into the same "Cough" class due to their similar auditory properties. The manually labeled portion

[1] Hexoskin
[2] Hexoskin

of the dataset represented 25% of the total dataset, which is equivalent to about 2445 instances used for training. This annotated section of the database makes it possible to use supervised learning to classify the activities. The process involves extracting various descriptors from segmented audio files such as MFCC coefficient, RMS vector, spectral bandwidth, zero crossing rate, and spectral contrast. Next, the data is expanded, and the best features are selected to apply the XGBoost model on the remaining data. The results are then manually verified to avoid any errors caused by the model.

C. Data preprocessing

The normalized raw data underwent denoising using discrete wavelet transform with interval-dependent thresholding. To remove noise from detail coefficients, Daubechis wavelet db10 at three levels was utilized. [7] Subsequently, the denoised signal was divided into short time windows, resulting in frames that are useful for cough detection. Each frame lasts 1.5 seconds with an overlap of 25%. During this processing step, each acceleration in the three axes is considered. The information that needs to be fed into the classifier input has a two-dimensional shape, with columns relating to the three directions of accelerations, and rows representing the samples of each acceleration frame.

D. Data balancing

Before dividing the data into segments, we initially had 18% of the samples belonging to the "Cough" class, while the remaining 82% belonged to the "Non-Cough" class. It is essential to maintain this proportional distribution even after segment creation. Following the segmentation process, which involved splitting the acceleration signal into frames and generating 3D data samples, we obtained a dataset consisting of 50,008 samples. Remarkably, 19.5% of these samples corresponded to cough frames. This imbalanced dataset highlights the need for oversampling techniques to enrich the cough data. In this research, the Synthetic Minority Over-sampling Technique (SMOTE) was employed to address the class imbalance [8]. By oversampling the small cough class, the dataset length increased to 39306, and the class imbalance was fixed to 3.1. This method helps to provide a balanced and more representative dataset for training machine learning models, thereby improving their accuracy and generalization performance.

III. Classification models

A. Traditional machine learning method

Traditional machine learning methods have been widely applied in cough detection systems. These methods typically involve several stages as shown in figure 2 top diagram, starting with feature extraction from cough segments. Following this step, feature selection techniques are applied to capture only the most pertinent information. For this task, The XGBoost model was selected due to its ability to handle large datasets with a high number of features effectively. It is trained using labeled cough data, with features as input and corresponding cough or non-cough labels as output. Hyperparameter tuning is then employed to optimize the model's performance.

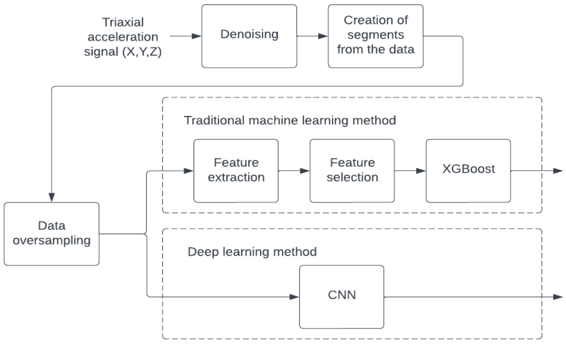
Triaxial acceleration signal (X,Y,Z)
Denoising
Creation of segments from the data
Traditional machine learning method
Feature extraction
Feature selection
XGBoost
Data oversampling
Deep learning method
CNN

Fig. 2. Block diagram of the two approaches

1) Feature extraction and selection:

1) Feature extraction and selection: The system generates a feature vector for each frame in order to help the machine learning module in learning the underlying characteristics from the raw signals. This feature vector is a combination of three vectors, each consisting of 51 features extracted from the acceleration data in the three axes (X, Y, and Z). These features can be categorized into two groups: those extracted from the time domain and those from the frequency domain after applying a Fast Fourier Transfrom (FFT) to the time signal. The frequency domain features include statistical features like mean, standard deviation, average absolute deviation, minimum and maximum values, difference between maximum and minimum values, median, median absolute deviation, interquartile range, count of negative values, count of positive values, number of values above mean, number of peaks, skewness, kurtosis, and energy. [9] The set of features estimated from time domain signals include similar features to those from the frequency domain, plus Hjorth features (Activity, mobility and complexity) as in [10] and Auto-regression coefficients with Burg order equal to three [11]. Furthermore, the system extracts the maximum and minimum index and their difference from both the time and frequency domain signals.

To enhance the performance of a classifier, it is crucial to select the most relevant and informative features that are closely related to the activity being recognized. This involves

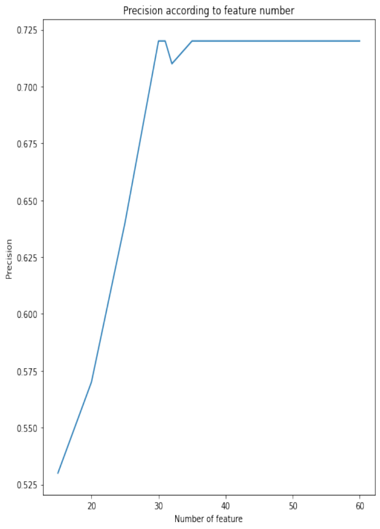

Fig. 3. Precision according to feature number

discarding any redundant or irrelevant features before feeding the feature set into the classifier. By doing so, the classifier can focus on the essential features and improve its overall performance. By selecting the top 30 most important features, Figure demonstrates that the model achieved a maximum precision of 72% before undergoing any tuning.

2) XGBoost model: The XGBoost algorithm is widely recognized as one of the most powerful and efficient machine learning models. By combining many weak classifiers into a single, robust classifier, XGBoost is able to achieve high levels of accuracy and predictive power. This ensemble algorithm is based on decision trees and uses a gradient refinement framework [12]. The output of a tree in gradient boosting can be expressed as follows:

$$f(x) = W_q(x_i)$$

where x is the input vector and W_q is the score of the corresponding leaf q. The output of an ensemble of K trees will be :

$$y_i = \sum_{k=1}^{K} f_k(x_i)$$

At step t, the algorithm in gradient boosting aims to minimize the following objective function J:

$$J(t) = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \sum_{i=1}^{t} \Omega(f_i)$$

where the first term contains the train loss function L (e.g.mean squared error) between the real class y and predicted output $\hat{y}$ for the n samples and the second term is the regularization term, which controls the complexity of the model and helps to avoid overfitting. In XGBoost, the complexity is defined as :

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

In this equation, γ and λ are hyperparameters that control the complexity of the model. γ is the minimum loss reduction required to make a further partition on a leaf node, and λ is the regularization term on the weights of leaf nodes. T represents the total number of leaf nodes in the tree, w_j represents the weight of leaf node j and the term $\frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$ penalizes the weights to avoid overfitting. The complexity term is added to the objective function to balance the trade-off between minimizing the training loss and controlling the complexity of the model, promoting simpler and more generalized solutions.

3) Hyperparameter tuning: In order to optimize the performance of the model, we tuned various hyperparameters. The XGboost best performance were obtained with the following hyperparameters : number of estimators =100, Maximum depth of a tree = 18, Minimum loss reduction required to make a further partition on a leaf node of the tree = 3, L1 regularization term on weights = 75, Minimum sum of instance weight needed in a child=3, L2 regularization term on weights=0.6, the subsample ratio of columns when constructing each tree=0.7

B. Deep learning Approach

For this task, deep learning models utilizing Convolutional Neural Networks (CNNs) were employed. The selection of a CNN-based architecture was motivated by its ability to process acceleration data with consideration for its spatial structure.

1) CNN Architecture: In this study, we applied a convolutional neural network (CNN) architecture for the classification of pre-processed and segmented data. The network architecture, as illustrated in Figure 4, comprised three stages: an input stage, a feature extraction stage with five convolutional layer blocks, and an output provision classification stage. The input stage received pre-processed and segmented data, denoted as X, containing the training dataset with a size of (N × H × D). Here, N represents the size of the dataset, H represents the time series sample number (also known as frame size), which we set to 96, equivalent to one and a half seconds of data. Additionally, D represents the number of sensor axes (also referred to as acceleration dimensions), which we set to 3.

The feature extraction stage, consisting of five convolutional layer blocks, played a crucial role in identifying and extracting relevant features. Each block utilized convolutional layers with varying filter sizes ranging from 32 to 512 (32, 64, 128, 256, 512), with each layer having more filters than the previous one. Batch normalization and max pooling with a pool size of 2 were applied after each convolutional layer to reduce computational requirements and identify crucial features. The max pooling technique allowed the network to identify important features and compress layer sizes through subsampling, thereby helping to prevent overfitting and reduce computational requirements. Following the convolutional layers, a flatten layer was employed to transform the output into a 1-dimensional array. The architecture further included one fully connected layers with 288 nodes, respectively. Both layers utilized the rectified linear unit (ReLU) activation function, kernel regularization with a value of 0.01, and a dropout rate of 0.5. These layers played a role in capturing higher-level representations and enabling more complex feature interactions.

For the final classification stage, the output layer employed the sigmoid activation function for binary classification. The model was compiled with binary cross-entropy loss and utilized the Adam optimizer, which helped optimize the network's weights and biases to minimize the loss function and improve classification performance.

2) Hyperparameters optimisation: Deep learning models contain a considerable number of hyperparameters and their setting is very complex. Part of the configuration of the neural network is to decide how many hidden layers of nodes will be used between the input and output layers of the network.The number of nodes used for each layer must also be determined. These configuration variables are set manually and are part of the hyperparameter tuning of an artificial neural network. It is also possible to draw inspiration from the choice of hyperparameters from related topics that have dealt with raw data like the ones treated in this subject.

For the CNN-1D model, the best performance were obtained with the following hyperparameters: batch size=32, number of epochs=30, number of convolutional filters=32, kernel size=3, learning rate= 0.001, dropout rate= 0.5 and dense layer sizes are 1024 and and 288 respectively.

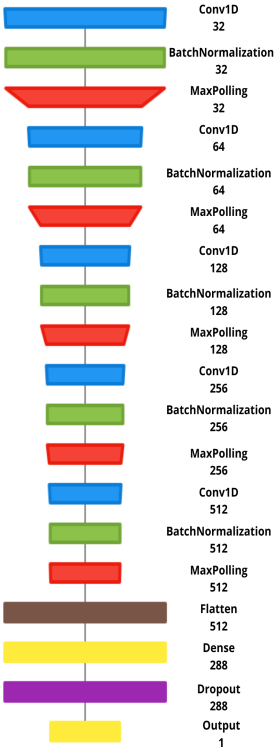

Fig. 4. CNN network illustration.

IV. Classification results

A. Evaluation metrics

In this supervised classification work, some metrics were used to evaluate the performance of the classification model, including:

- Accuracy: it is the ratio of correct predictions of cough segments to the total number of predictions made.
- Precision: This metric measures the proportion of true cough frames predictions among all positive predictions made by the model. It is calculated as the number of true cough frames prediction divided by the total number of positive predictions made.
- Recall: This metric measures the proportion of cough frames predictions that were actually correct. It is calculated as the number of true cough frames predictions divided by the number of cough frames in the dataset.
- F1 score: This metric is the harmonic mean of precision and recall, and is calculated as the product of precision and recall divided by their sum. It is a good metric to use when we want to balance precision and recall.

B. Performances per subject

Due to the relatively small dataset size when usind deep learning based machine learning models, the Leave-One-Person-Out Cross Validation (LOOCV) is the type of cross validation technique used to evaluate the performance of the implemented models. In this work, a particular LOOCV is chosen : the dataset is divided into $K = 36$ folds since 36 persons participated to the experience and each fold contains exclusively data of one participant. The model is trained on all the folds except for one, which is held out as the test set. This process is repeated for each fold (equivalent to each participant) in the dataset, so that every participant is used as the test set once. This particular choice of LOOCV configuration guarantees that the model is not suited to certain participants.

Figures 5, 6 and show, for each model, the accuracy (right side) values for every participant, whose identifier (chosen as a number) is given in the left side. One can notice the good performances obtained with XGBoost for all participants (accuracy range is $[0.81;0.96]$). The CNN model demonstrates excellent performance across major participants, (accuracy range is $[0.84;0.97]$). This conclusion confirms the CNN optimisation architecture and confirms its use.

C. Overall performances

The final model performance is calculated as the average performance across all the folds. The performance of the classifiers can be evaluated by examining the metrics presented in the table I. In terms of accuracy, the CNN achieves the highest score of 91%, closely followed by XGBoost with 90%. Regarding precision, all three classifiers demonstrate relatively similar performance, with XGBoost and CNN achieving 77%. In terms of recall, the CNN performs the best, achieving an impressive score of 82%, while XGBoost achieve 74%. Similarly, the CNN also achieves the highest F1 score of 79%, followed by XGBoost with 74%. Overall, the CNN classifier demonstrates the strongest performance across multiple metrics, exhibiting higher accuracy, recall, and F1 score compared to XGBoost. However, it is worth noting that each classifier has its strengths and weaknesses, and the choice of the most suitable model ultimately depends on the specific requirements and objectives of the task at hand.

	XGBoost	CNN
Accuracy	90%	91%
Precision	77%	77%
Recall	74%	82%
F1 Score	74%	79%

TABLE I

Performance metrics of the different classifiers.

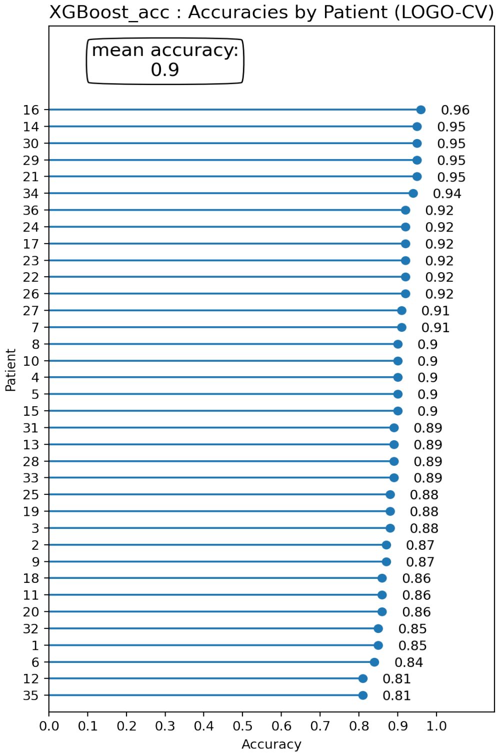

Fig. 5. XGBoost model accuracy results for each participant.

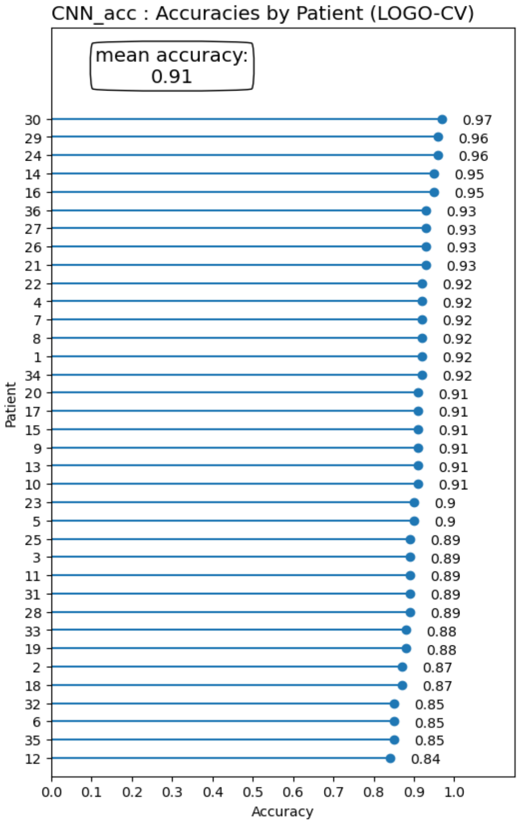

Fig. 6. CNN model accuracy results for each participant.

V. Conclusion and Future work

In this research, a machine learning-based method was developed for the detection of cough using acceleration signals. Two approaches were proposed: a conventional method using XGBoost as a classifier and a deep learning method using CNN-1D as an architecture. Both models achieved high accuracy in distinguishing coughing signals from other activities like clearing throat, talking, laughing, and movements in different directions. The study results demonstrate the feasibility of cough monitoring using the Hexoskin device's accelerometer measurements. The use of wearable connected textile sensors presents new possibilities for user-independent cough detection. This technology holds potential for applications such as respiratory health monitoring, early detection of respiratory infections, remote monitoring, and surveillance in public settings.

Future work in this field can focus on several aspects. Firstly, exploring Signal Preprocessing techniques could enhance cough detection accuracy by improving performance and precise identification of cough signals. Additionally, expanding the dataset to include a more diverse population, including individuals with chronic respiratory conditions or specific respiratory diseases, would enhance the generalizability and robustness of the developed models. Furthermore, integrating accelerometer measurements with other physiological sensors like ECG and respiration sensors can provide a comprehensive analysis of cough-related signals, potentially improving accuracy and reliability. Additionally, deploying the developed cough detection system in real-world scenarios and integrating it with existing healthcare infrastructure warrants exploration. This involves validation studies in clinical settings and evaluating the system's performance in detecting cough in various real-life situations.

In conclusion, this research contributes to the advancement of cough detection using machine learning and accelerometer signals. The developed models exhibit promise in accurately identifying coughing events and have the potential to be implemented as user-independent tools for cough detection, benefiting various healthcare applications and public health initiatives. Continued research and development in this area will lead to improved respiratory health monitoring and management.

REFERENCES

[1] S. Sharma, M. F. Hashmi, M. S. Alhajjaj, "Cough", *Treasure Island, StatPearls Publishing*, Updated 2022 Aug 18, URL: `https://www.ncbi.nlm.nih.gov/books/NBK493221/`.

[2] T. Drugman, J. Urbain, N. Bauwens, R. Chessini, C. Valderrama, P. Lebecque, T. Dutoit, "Objective Study of Sensor Relevance for Automatic Cough Detection", *Journal of Biomedical and Health Informatics*, vol. 17, pp. 699–707, May 2013.

[3] H. Mohammadi, A. A. Samadani, C. Steele, T. Chau, "Automatic discrimination between cough and non-cough accelerometry signal artefacts", *Biomedical*

Signal Processing and Control, vol. 52, pp. 394–402, 2019.

[4] J. Y. M. Chan, S. A. Tunnell, J. A. L. Jacobs, "Systems, methods and kits for measuring cough and respiratory rate using an accelerometer", *US Patent App 13/783,257*, 2014.

[5] M. Pahar, I. Miranda, A. Diacon, T. Niesler, "Deep neural network based cough detection using bed-mounted accelerometer measurements", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2021.

[6] I. D. S. Miranda, A. H. Diacon, T. R. Niesler, "A Comparative Study of Features for Acoustic Cough Detection Using Deep Architectures", *in 2019* 41_{st} *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2601–2605, 2019, doi: 10.1109/EMBC.2019.8856412.

[7] D. L. Donoho, I. M. Johnstone, "Ideal Spatial Adaptation by Wavelet Shrinkage.", *Biometrika*, vol. 81, no. 3, pp. 425–55, 1994.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique", *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[9] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, "A Comparison of Feature Extraction Methodsfor the Classification of Dynamic ActivitiesFrom Accelerometer Data", *IEEE transaction on biomedical engineering*, vol. 56, no. 3, pp. 871–879, mar 2009.

[10] H.-L. Le, D.-N. Nguyen, T.-H. Nguyen, H.-N. Nguyen, "A Novel Feature Set Extraction Based on Accelerometer Sensor Data for Improving the Fall Detection System", *Electronics*, 2022.

[11] Z. Jiadong, S.-S. Rubén, P. J. M., "Feature extraction for robust physical activity recognition", *Human-centric Computing and Information Sciences*, 2017.

[12] D. Nielsen, *Tree Boosting With XGBoost :Why Does XGBoost Win "Every" Machine Learning Competition?*, Master's thesis, Norwegian University of Science and Technology, 2016.