A Hybrid Deep Learning Approach for Freezing of Gait Prediction in Patients with Parkinson's Disease

Hadeer El-ziaat[1], Ramadan Moawad[3]
Department of Computer Science
Future University in Egypt (FUE)
Cairo, Egypt

Nashwa El-Bendary[2]
College of Computing and Information Technology
Arab Academy for Science, Technology and Maritime Transport (AASTMT), Aswan, Egypt

***Abstract*—The main objective of this work is to enhance the prediction of the Freezing of Gait (FoG) episodes for patients with Parkinson's Disease (PD). Thus, this paper proposes a hybrid deep learning approach that considers FoG prediction as an unsupervised multiclass classification problem with 3 classes: namely, normal walking, pre-FoG, and FoG events. The proposed hybrid approach Deep Conv-LSTM is based on the use of Convolutional Neural Network layers (CNN) and Long Short-Term Memory (LSTM) units with spectrogram images generated based on angular axes features instead of the normal principle-axes features as the model input. Experimental results showed that the proposed approach achieved an average accuracy of 94.55% for FoG episodes early detection using Daphnet and Opportunity publicly available benchmark datasets. Furthermore, the proposed approach achieved an accuracy of 93.5% for FoG events prediction using the Daphnet dataset with the subject independent mode. Thus, the significance of this study is to investigate and validate the impact of using hybrid deep learning method for improving FoG episodes prediction.**

***Keywords*—Freezing of Gait (FoG); Parkinson's disease (PD); angular axes features; spectrogram; convolutional neural network (CNN); long short-term memory (LSTM)**

I. INTRODUCTION

Parkinson's Disease (PD) is a disorder that affects the patient's nerves, which are recognized by low levels in the brain's dopamine. Also, PD is the second most common symptom after Alzheimer's disease [1]. Low levels of dopamine lead the patients to the inability to control their body motion or activity. PD can affect the patients by two kinds of symptoms, motor, and non-motor symptoms, the motor symptoms, or cardinal symptoms, which are related to the movement in general, including resting tremor, stepping slowness (bradykinesia), postural instability (issues in balance), and Freezing of Gait (FoG). Non-motor or dopamine-non-responsive including cognitive weakness, sleeping behavior problems, sense of smell loss, difficulty defecation, talking and swallowing problems, and other related side effects to the human's sensory [2].

FoG is a side effect and one of the main symptoms for patients with both advanced and early PD stages. It is one of the most debilitating motor symptoms in patients with PD as it may lead to falls and a loss of independence. FoG can occur in the arms (which affects the writing ability), face (which affects the patient's vision), and leg (which results during gait initiation by a series of short steps with tremors on lower limbs, turning, or walking towards a particular goal) [2].

FoG could be seen in different forms like 1) A complete FoG cycle 2) Freezing with knee-trembling 3) Walking with very short steps. For some patients, there is a brief trembling of the feet in place followed by short steps, while others experience total immobility and are unable to move it all for a few moments.

The main objective of this study is to predict FoG episodes based on a hybrid deep learning approach using time-series episodes windowing and relay on the use of angular axes feature to help in predicting FoG episodes with their different occurrence cases as mentioned previously. The baseline model has experimented with the proposed one (Conv-LSTM), a model based on the use of a deep convolutional neural network (CNN). This model is also implemented for detecting/predicting FoG episodes. The developed models in this paper have been tested with both Daphnet and Opportunity benchmark datasets.

Accordingly, the main contributions of this paper are summarized in the following points:

- Updating the publicly available Daphnet dataset through adding a new label (3) that specifies the pre-FoG episodes prediction.
- Developing a deep learning (CNN-LSTM) hybrid approach for handling the problem of FoG episodes prediction in patients with PD.
- Investigating subject-dependent spectrograms for predicting FoG episodes.
- Testing and validating the performance of the proposed approach through implementing multiple experiments using the benchmark datasets.

The remainder of this paper is organized as follows. Section II presents a review of the state-of-the-art related work employing different deep learning algorithms for predicting/detecting FoG episodes based on different approaches. Section III presents an overview of the main structure of the proposed approach. Results and discussion are presented in Section IV. Conclusions and future work are highlighted in Section V.

II. Related Work

Relevant state-of-the-art studies addressing FoG detection and prediction are described in this section.

The authors of [4] developed an approach to predict FoG for patients with PD using a deep learning model based on the Recurrent Neural Network (RNN) and Long Short-Term Memory networks (LSTMs). The achieved results were 94.7% with 1-second prediction, 82.9% with 3 seconds prediction, and 68.1% with 5 seconds prediction.

Also, the authors in [5] aimed to detect FoG based on the use of the DL approach, the data was collected from one wrist based on linear and angular acceleration. The collected data was fed into a CNN model with 10-fold cross-validation and Leave-one-out-subject-out (LOSO) Cross-Validation. The achieved results were 83% and 86% for sensitivity and specificity, respectively when using LOSO-CV. When using the 10-fold CV the achieved results were 88% and 90% for sensitivity and specificity, respectively.

For capturing long-range dependencies in variable-length input sequences, the authors in [6] used a Deep Recurrent Neural Network (DRNN) for building recognition models that can fulfill the paper's purpose. The proposed algorithms like cascaded, unidirectional, and bidirectional architectures are based on LSTM DRNNs. By using Unidirectional DRNN for the UCI dataset, the results were 96.7%, 96.8%, 96.7%, and 0.96. By using Unidirectional DRNN for the USC-HAD dataset, the results were 97.8%, 97.4.0%, 97.4%, and 0.97. By using Bidirectional DRNN for Opportunity dataset, the results were 92.5%, 86.7%, 83.5% and 0.92. Using the Cascaded DRNN for the Daphnet dataset, the results were 94.1%, 84.7%, 78.9%, and 0.93. Using the Cascaded DRNN for the Skoda dataset, the results were 92.6%, 93.0%, 92.6%, and 0.92. All datasets were measured with accuracy, average precision, average recall, and f1 score, respectively.

The author in [7] sought to detect FoG episodes from the data signal for the subject in/dependent with a 3D accelerometer. The proposed model implemented on patients in the daphnet dataset was based on the RNN LSTM model with three accelerometer sensors. The overall results were 80% and 79% for specificity and sensitivity, respectively when using all features (statistical and frequency domain features) with sensor and subject independent. When using only the statistical features the results were 89% and 34% for specificity and sensitivity, respectively with sensor and subject independent. The achieved results when using the frequency domain features were 80% and 77% for specificity and sensitivity, respectively with sensor and subject independent. They also implement a baseline model using the Random Forest classifier for comparing the achievements of each model. The results showed that the proposed RNN LSTM model outperforms the baseline model (RF classifier).

In [8], the authors developed a system for FoG detection that is based on CNN from a 2D acceleration signal. The average results for the subject independent were 80.7%, 69.29%, and 90.6% for accuracy, precision, and specificity, respectively. Moreover, the author in [9] proposed a hybrid deep learning algorithm for detecting FoG episodes. Their first model was based on the use of convolutional layers with LSTM units. The achieved results were 87.8% accuracy, 88.1% sensitivity, 89.1% specificity, and 88.4% geometric mean. The second model was based on convolutional layers with GRU (Gated Recurrent Unit). The results achieved were 85.4% accuracy, 94.8% sensitivity, 84.7% specificity, and 89.5% geometric mean.

The goal of the authors in [10] was to detect FoG using four types of feature sets with a DNN with 4-sec windowing. A model of CNN with MLP layers was introduced. It contains two convolutional layers with a max-pooling layer for feature extraction and with three fully connected layers for classification. The achieved sensitivity was 93.1% and 75% for specificity.

Furthermore, the authors in [19] used deep learning approaches and image processing techniques to detect FOG. The presented approach was based on the use of 1D-ConvNet. The results were 88.6% for sensitivity and 78% for specificity.

The aim of the study proposed in [20] was to detect FOG episodes in PD patients with a proposed model that consists of eight layered of 1D-ConvNet using two activation functions (sigmoid and hyperbolic). Down-sampling and low pass filter are used for data preprocessing. The achieved results were 89% for accuracy, 91.9% for sensitivity, and 89.5% for, specificity, respectively.

The purpose of the approach proposed in [21] was to detect FOG episodes using CNN based on two types of features, namely time and frequency domain features with 2.5 s windowing. The features were extracted from a tri-axil accelerometer and gyroscope sensors from a smartphone located in the patient's trouser pocket. Results were 91.8%, 93.8%, and 90.1% for F1-score, sensitivity, and specificity, respectively.

The main objective of this study is to investigate the impact of using deep feature learning based on angular axes spectrogram images and different windowing mechanisms through a hybrid deep learning approach for improving FoG episodes prediction.

III. Proposed Approach

The proposed approach section is divided into four stages, starting with dataset acquisition, data preparation, going throw deep learning features, and ending with the classification, as illustrated in Fig. 1.

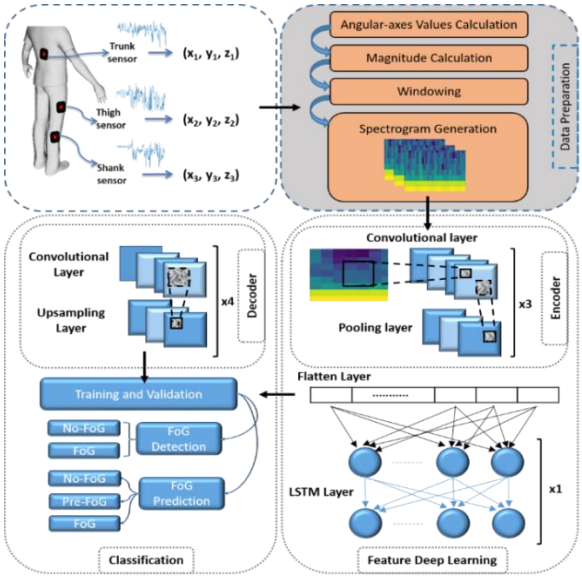

Fig. 1. Structure of the Proposed Hybrid FoG Prediction Model.

A. Dataset Acquisition

To train and validate the proposed models, two benchmark datasets are considered; namely the Daphnet and the Opportunity datasets of FoG time-series data.

1) The Daphnet dataset, which is available publicly from the UCI Machine Learning Repository [11], has been used for validating the proposed approach. The dataset was collected from a total of 10 PD volunteers patients, where 8 subjects have experienced FoG while performing several walking tasks in the lab. The Shank (ankle), the thigh (above the knee), and the trunk (lower back) of each subject were used to place a three-wearable tri-axial accelerometer for data recording [3]. From Fig. 3, which presents the same portions for ankle, knee, and trunk sensors of the walking, pre-FoG, and FoG signals it can be observed that the existence of the signal spacing when there are pre-FoG or FoG samples. The original data samples have been labeled as 0 for describing out-of-experiment events, 1 for describing no-FoG events, and 2 for describing FoG events. Sessions with a duration between 20 and 30 minutes are accomplished by the subjects in the Daphnet dataset to represent different characteristics of daily walking [11].

2) The opportunity dataset, which contains recordings that are collected from 12 subjects using 15 networked sensor systems with 72 sensors [12]. For each subject, 6 different runs were recorded 5 of them were the Activity of Daily Living (ADL) the remaining was a drill run. For this study, the activity recognition subset of the opportunity dataset was used. Only 4 subjects with ADL runs were used, which corresponds to recordings of three tri-axial accelerometer placements namely, hip, above the right knee (RKN^), and below the right knee (RKN_).

B. Data Preparation

Originally the validation Daphnet dataset doesn't contain samples with label 3 corresponding to pre-FoG events. For featuring the pre-FOG episodes with a new label as shown in Fig. 2, all samples with the same label (unified labels) for a specific window time before label 2 have been converted into label 3. Therefore, for the FoG prediction problem, a multi-class classification has been handled with the proposed approach. This method is used only for prediction implementations, but for the detection implementations, labels 1 and 2 only are used. Subjects four and ten have not experienced any Freezing of Gait episodes while performing several walking tasks. As previously mentioned in the data description section, the Daphnet dataset consists of ten PD subjects; the data of the two previous subjects and all records with zero labeling have been neglected. The data preparation phase has consisted of four steps, as follows:

1) Angular axes values calculation: The angular axes features [13], or axes of rotation, provide a combination of three rotations about different axes to represent the 3D orientation of an object. Because of the need to focus on the patient's rotation and orientation the reliance on the accelerometer features will not be efficient. Motions are dominated by rotations; therefore, it is needed to avoid the use of accelerometers and use a gyroscope sensor [14].

For the angular-axes calculation x, y, and z of the principle-axes from the three sensors have been used to calculate the angular axes features (Roll, Pitch, and Yaw). As shown in equations (1), (2), and (3) details of calculating the angular values, Roll (r), Pitch (p), and Yaw (y), about the x, y, and z axes, respectively, where Π = 3.14 is constant.

$$Roll = 180 * \arctan(y/\sqrt{x^2 + z^2}))/\pi \quad (1)$$

$$Pitch = 180 * \arctan(x/\sqrt{y^2 + z^2}))/\pi \quad (2)$$

$$Yaw = 180 * \arctan(z/\sqrt{x^2 + y^2}))/\pi \quad (3)$$

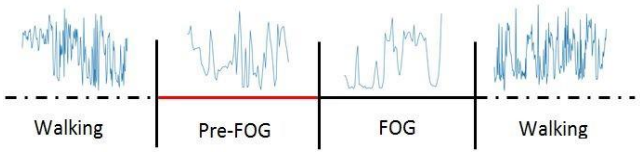

Fig. 2. An Example Signal Pattern for Accelerometer Signal Categories.

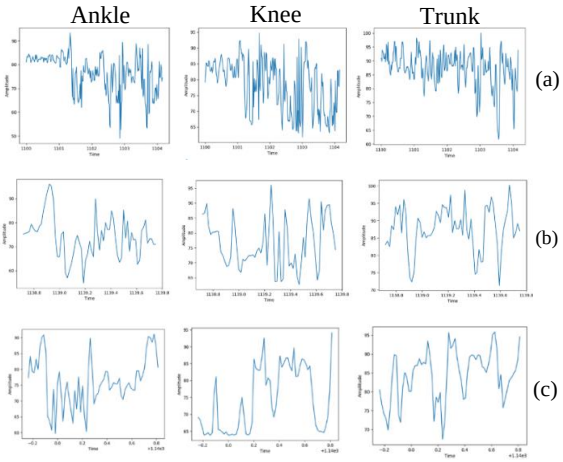

Fig. 3. Portion of Walking (a), pre-FoG (b) and FoG (c) Signals for Ankle, Knee and Trunk Sensor.

2) Magnitude calculation: After converting the principle-axes into angular axes, the magnitude of the obtained values has been calculated from each record of the three angular axes features values, according to equation (4), where *s* refers to the used ankle's (shank), knee's (lower thigh), or trunk's (lower back) sensor.

$$Magnitude = \sqrt{(r_s^2 + p_s^2 + y_s^2)} \quad (4)$$

3) Windowing: From each sensor, all the calculated magnitudes are divided into two overlapping windowing sizes, namely fixed and dynamic sizing based on the data labels. This paper presents two different schemes for selecting the best slicing technique and for testing various windowing mechanisms. Method 1 (scheme 1): The first method is adopting a 1 sec. window size, each window contains 76 samples, and each sample is 15 milliseconds [3]. Method 2 (scheme 2): On the other hand, a partially overlapping windowing method of dynamic-sized windows based on the data labels [3].

4) Spectrogram generation: Because other signals like nonstationary or non-periodic frequencies differ in time, in this step the information about the time domain and the frequency domain of real-life signals are needed, which is better computed from time-frequency analysis [15]. One of the basic visual tools for displaying the time-frequency analysis information is the spectrogram. The spectrogram is a 2D map, the vertical axis represents the frequency, and the horizontal axis represents the time of the signal. For spectrogram-based CNN, the amplitude provides a 2D array of successive segments used as an input feature for the spectrogram-based CNN. The spectrogram parameters are window size, noverlap, nfft (length of the FTT), and fs (sampling frequency). The input to the spectrogram is a signal data x and the output will be a matrix of 2D array q. The output q will have (nfft/2+1) rows if nfft is even and (nfft+1)/2 rows if nfft is odd. For columns, will be (the length of q – noverlap) / (length (window-noverlap)) [16].

Fig. 4 presents the 3D accelerometer axes spectrograms for FoG activity on the three sensors, namely the ankle, knee, and trunk. The time in milliseconds is presented on the x-axes and the frequency in Hz is presented on the y-axes, each row presents one of the axes, namely x, y, and z, respectively. The same portion of data is used for all sensors (ankle, knee, and trunk). It is visualized also from the figure areas with darker colors means that for a specific time and frequency point, the color of the image will be darker with a lower amplitude. Similarly, the color will be lighter with high magnitude.

It can be also observed that the most effective sensor placement is the trunk sensor, the readings from this sensor are better than the ankle and the knee sensors placements. It can be observed the lighter areas emitted from the trunk sensor, which means that those areas have a high frequency of FoG activity with high severity of FoG in the same area.

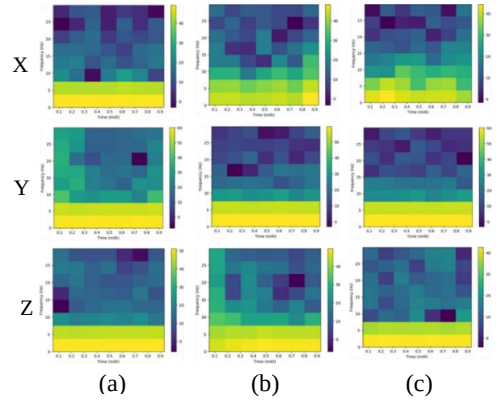

Fig. 4. Samples for each Axis (X, Y, and Z) for FoG Activity on the Ankle (a), Knee (b) and Trunk (c) Sensor.

C. Deep Learning Features

For the proposed approach (Conv-LSTM), with the use of three Convolutional layers, which receive and extract features from spectrogram images with intermediate three max-pooling layers (encoder) as shown in Fig. 1. The encoder in the network compresses or down-samples the input into a fewer number of bits. The space represented by these fewer numbers of bits is often called the latent space, at this point, the input has compressed to the maximum. A convolution layer tries to extract higher-level features by replacing data for each pixel with a value computed from the pixels. From Fig. 1 it can be observed how features are extracted from each Conv-LSTM layer, clarifying, and discussing the use of intermediate LSTM layers, which are used for the feature deep learning phase, and CNN layers that are used for classification.

It can be also observed how features are extracted from each Conv-LSTM layer after extracting features from spectrograms, by using an intermediate LSTM layer that is used to receive the output of the last convolutional layer. To do so, the flatten layer is used because convolutional layers output a shape of four dimensions whereas the LSTM layer needs a three-dimension input. Like the proposed approach, the baseline 2D CNN approach also used the encoder as a feature extractor with three convolutional layers and three max-pooling layers, but without using any LSTM layer.

CNN is a class of deep, feed-forward AI neural networks that are used in Image & video recognition/classification, Video to Text (seq. to seq.), and Image Question Answering. A CNN consists of an input and output layer as well as multiple hidden layers. Typically, a CNN's hidden layers consist of Convolutional layers, pooling layers, and fully connected layers. Typically, a CNN is used mostly for image and video recognition, classification, video to text, and image question answering applications. CNN learns useful features from data itself dispenses the use of the hand-crafted features and instead depends on learning features that the network automatically extracted. So, it is a combination of a feature extractor and a classifier. Mainly the CNN contains convolutional, pooling, fully connected layers, and soft-max [17], [18].

- Convolutional layer is responsible for extracting useful features automatically by learning and knowing the best features by selecting the highest weights of each neuron in each convolutional layer.
- Max-pooling layer is used to reduce the size of the feature map, which results in having a smaller number of parameters and computations.
- Up-sampling layer is a backward stride convolution and performed for end-to-end backpropagation learning from the pixel-wise loss
- Fully connected layer means that all the input neurons in each layer are connected to the previous layer.
- Soft-max layer acts like a classifier in which many probabilities were proposed from each class and the class with the highest probability was predicted.

LSTM layer, Fig. 5 magnifying the steps of how the LSTM network works. The first step is to decide what information will be thrown away from the cell state. This decision is made by the sigmoid layer (σ), which is called the "forget gate layer", it looks at xt (current input) and h(t-1) (output of the last LSTM unit) and outputs a number between 0 and 1 for each number in the cell state c(t-1) (memory of the last LSTM unit). 1 means "keep the previous" result and don't forget it while 0 means "don't keep the previous result" and forget it. For simplicity, if it is needed to predict the next word based on all the previous ones, the cell state might include the gender of the present subject so that the correct pronouns can be used. But when a new subject came up, the previous gender of the old subject needs to be forgotten.

In the next step, a decision as to what new information will be stored in the cell state is taken in two steps. First, a decision for which values will be updated is taken by a sigmoid layer called the "input gate layer". Next, a tanh layer creates a vector of new values ct (new updated memory). The next step is to combine the previous two outputs from each layer to create the updated cell state. For simplicity going back to the example, here adding the gender of the new subject to the cell state by replacing it with the old one that was forgotten is needed.

Finally, the ht (output) is based on the value stored in the cell state. First running a sigmoid layer that decides, which parts of the cell state will be going to be output is needed. Then, go through the tanh layer and multiply (x) the output of each layer.

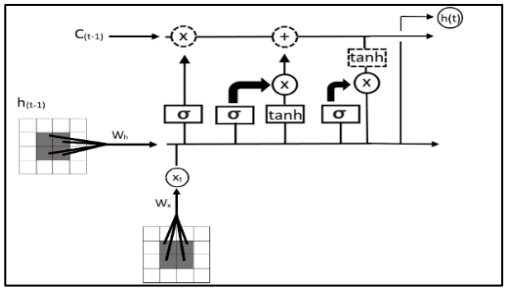

Fig. 5. Presents an Illustration of the Proposed Hybrid Model.

As the problem is based on unsupervised learning, the use of an Auto-Encoder mechanism is taking place as it has the option to not use dense layers, it can use the convolutional layers itself to learn, which is better for video, image, and series data. The encoder compresses or down-samples the input into a fewer number of bits, the space represented by these fewer bits is often called the latent space or bottleneck. At a maximum level, the input at a particular point is compressed that's why it can be also called the "maximum point of compression". The decoder is the reverse of the encoder, and it is importance lies in rebuilding the original image with the highest possible quality.

D. Classification

For the proposed model, both datasets are divided into sub-datasets, one for training and the other for validation using a k-fold cross-validation method applied for both training and validation. A hybrid deep learning structure is used as shown in Table I, the hybridization here is based on three convolutional layers with intermediate max-pooling layers, a flatten layer, three LSTM layers for feature deep learning, and four convolutional layers with intermediate up-sampling layers for classification. Convolutional layers as mentioned previously learn features automatically in each layer, so it doesn't need any hand-crafted features, and the reason is that the deep architecture, which includes multiple layers allows those layers to be stacked. So, this deep architecture can characterize the prominence of signals on different scales. In each convolutional layer, each neuron carries the maximum weight (the maximum weight means a better result) from the input layer and if a neuron was for example in the third layer it takes the output of the previous layer as its input. The convolutional layer's network has its output as a convolution fully connected neural network instead of matrix multiplication.

The structure of the baseline model (Deep 2D CNN) model was fed with spectrogram images as model input. The same as the proposed model, a baseline model consists of three convolutional layers with intermediate max-pooling layers for feature deep learning and four convolutional layers with intermediate up-sampling but without flatten and LSTM layers.

IV. Result and Discussion

This section discusses different models' results used in this paper to reach the best approach that effectively influences predicting FoG episodes. Also presents and discusses the experimental outcomes of using the proposed hybrid deep learning scheme for FoG episodes prediction. Both datasets are divided 80% for training with a random dividing and 20% for validation. The implementation consists of two phases using the Daphnet dataset:

- Working on the original dataset applying all sensors and each sensor apart (Ankle, Knee, and Trunk).
- Working on the original dataset after applying the proposed angular axes features for in/dependent subjects using all sensors and each sensor apart (Ankle, Knee, and Trunk).

The performance measurements as shown in equations (5), (6), (7), (8), and (9) are the accuracy, precision, recall, F-measure, and Specificity.

$$Accuracy = \frac{tp+tn}{tp+fn+fp+tn} \quad (5)$$

$$Precision = \frac{tp}{tp+fp} \quad (6)$$

$$Recall = \frac{tp}{tp+fn} \quad (7)$$

$$F-measure = 2\frac{precision*recall}{precision+recall} \quad (8)$$

$$\text{Specificity} = \frac{TN}{(TN+FP)} \quad (9)$$

A. Daphnet Dataset Results

In subject-independent FoG prediction, from Table II and Table III it can be observed that the results derived from sensor independent are almost the same as the results when using only data from the trunk sensor (sensor dependent). But for overall performance, using sensor independent is better than using sensor dependent. It was also observed that using a windowing with 1-second achieves a better performance than using a windowing with 4-seconds. Another enhancement is for using angular axes features over principle-axes. For the 2D CNN model, same as the hybrid Conv-LSTM model; the trunk sensor outperforms other sensors, and angular axes features achieve better performance. Using 1-second windowing records an enhancement against using 4-seconds windowing and better overall performance is achieved when using sensors independent over-dependent sensors. In addition, for the 2D CNN model, from Table IV and Table V it can be observed that using only CNN without LSTM the model achieves better results than using a hybrid structure of CNN with LSTM.

The results show that the implemented deep 2D CNN model achieves the highest accuracy against the proposed one, it can be observed in Fig. 7. It also emphasizes that the use of the angular axes features is much better than the principle-axes. For subject-independent and sensor dependent with 1-second (67 samples) windowing the two implemented models namely, 2D CNN and hybrid deep Conv-LSTM achieves an increase in the performance using angular axes against principle-axes by 2.7%, and 2.8% using ankle sensor for the two models respectively. And for the knee sensor the performance increased by 2.6%, and 2% for the two models, respectively. The enhancement in the trunk sensor is 3.3%, and 2.5% for the two models, respectively. When using sensor independent the angular axes features achieve an enhancement of 3.8%, and 2.2% for the two models, respectively, over the principle-axes.

Implementing subject independent with sensor dependent for FoG prediction with a 4-seconds (268 samples) windowing technique, the results as shown in Fig. 7 clarifies that using a bigger windowing technique negatively affects the performance. When using 1-second windowing against 4-seconds windowing with angular axes features from Fig. 6 and Fig. 7, the performance increases by 9% and 4.3% for the ankle sensor with 2D CNN, hybrid deep Conv-LSTM, respectively. On the knee sensor, the enhancement is 6.5% using the 2D CNN model and 4.4% using the hybrid Conv-LSTM model. The enhancement using trunk sensor is 4.8% and 3.7 for the two models, respectively. Finally, using sensor independent achieves increasing in the performance by 5.6% and 3.8% for the two models, respectively.

Applying the 4-seconds windowing technique using angular axes features outperforms using principle-axes as shown in Fig. 7. The enhancement on the ankle sensor is 2.3% and 5.6% for 2D CNN, and hybrid deep Conv-LSTM respectively. On the knee sensor, the enhancement increased by 3.1% and 3.6% for the two models, respectively. The performance using the trunk sensor increases by 2.6% and 2.8%. When applying sensor independent with angular axes features the performance against the use of principle-axes increased by 2.8% and 3.1% for the two models, respectively.

The implementation of Subject-independent FoG detection for deep learning models with sensor dependent from Table VI emphasized that the trunk sensor outperforms other sensors, and angular axes features achieve better performance against principle-axes. Also, using 1-second windowing achieves an enhancement over using the 4-second windowing. It has been observed from Table VI and Table VII that a better overall performance is achieved when using sensors independent over-dependent sensors. When comparing results of prediction with detection, it can be observed that FoG prediction outperforms FoG detection for both 2D CNN and hybrid deep Conv-LSTM models. In addition, from Table VIII and Table IX the results ensure that the 2D CNN model structure is better than the hybrid Conv-LSTM model structure.

Detecting FoG episodes for subject-independent and sensor dependent with 1-second (67 samples) windowing as shown in Fig. 8 when applying angular axes features outperforms using principle-axes. The enhancement on the ankle sensor is 2.6% and 3% for the 2D CNN and hybrid deep Conv-LSTM models, respectively. Furthermore, the enhancement on the knee sensor is 0.5% and 2.5% for the two models, respectively. The performance increased also when using the trunk sensor, the enhancement is 1% and 2.9%. An enhancement using angular axes features with sensor independent by 1.6% and 1.3%. Applying the 4-seconds windowing technique using angular axes features outperforms using principle-axes as shown in Fig. 9. The enhancement on the ankle sensor is 3.7% and 4.9% for 2D CNN, and hybrid deep Conv-LSTM, respectively. On the knee sensor, the enhancement increased by 3.3% and 3.5% for the two models, respectively. The performance using the trunk sensor increased by 4% and 3.2%. An enhancement is achieved when applying sensor independent with angular axes features by 2.9% and 3.4% for the two models, respectively against the use of principle-axes.

An enhancement can be observed in Fig. 8 and 9 when using angular axes features with 1-second, which equals 67 samples against 4-seconds (268 samples) windowing. The enhancement on the ankle sensor increased by 4.6% using the 2D CNN model and 4.3% using the hybrid deep Conv-LSTM model. On the knee sensor, the enhancement is 3.5% using the 2D CNN model and 4.9% using the hybrid deep Conv-LSTM

model. The enhancement on the trunk sensor achieves 5% using the 2D CNN model and 3.2% using the hybrid deep Conv- LSTM model. The implementation of the sensor independent has also increased the performance by 4.7% using the 2D CNN model and 4.3% using the hybrid deep Conv-LSTM model.

TABLE I. HYBRID CONV-LSTM MODEL STRUCTURE

Layer	Kernel size	Output map size	Activation	Optimizer
Conv2D	3*3	28*28*16	tanh	adadelta
Max-pooling2D	2*2	14*14*16		
Conv2D	3*3	14*14*8		
Max-pooling2D	2*2	7*7*8		
Conv2D	3*3	7*7*8		
Max-pooling2D	2*2	4*4*8		
Flatten	-	128		
LSTM	-	1*16		
Conv2D	3*3	4*4*8		
Up-sampling2D	2*2	8*8*8		
Conv2D	3*3	8*8*8		
Up-sampling2D	2*2	16*16*8		
Conv2D	3*3	14*14*16		
Up-sampling2D	2*2	28*28*16		
Conv2D	3*3	28*28*3	Sigmoid	

TABLE II. HYBRID DEEP CONV-LSTM 1 SEC. AND 4 SEC. WINDOWING SENSOR DEPENDENT

1-second windowing				4-second windowing	
Sensor	**Subject**	***Accuracy***			
		Angular axes	***Principle axes***	***Angular axes***	***Principle axes***
Ankle	1	89.5%	86.7%	84.4%	78.1%
	2	91.7%	86.6%	79.1%	77.9%
	3	91.3%	90.5%	80.4%	80.4%
	5	91.9%	86.8%	82.4%	75.9%
	6	90.6%	85.7%	87.6%	78.6%
	7	90.8%	84.8%	81.3%	75.1%
	8	**92.3%**	78.9%	**91.1%**	**83.2%**
	9	92.8%	**88.8%**	74.6%	75.1%
Knee	1	90.1%	87.1%	78.9%	80.1%
	2	91.8%	86.6%	87.1%	77.5%
	3	91.1%	89.8%	85.3%	78.2%
	5	90.6%	86.7%	87.3%	79.4%
	6	91.9%	87.1%	85.2%	84.5%
	7	89.2%	89.6%	82.4%	77.7%
	8	**92.3%**	**91.3%**	**90.8%**	**87.4%**
	9	90.5%	89.7%	81.4%	83.1%
Trunk	1	91.7%	88.2%	81.5%	84.7%
	2	93.5%	88.1%	89.6%	81.4%
	3	89.1%	92.1%	82.2%	83.2%
	5	93.1%	88.8%	87.1%	84.6%
	6	94.6%	90.6%	92.5%	85.3%
	7	94.1%	90.9%	92.1%	85.1%
	8	**94.7%**	**92.6%**	**91.7%**	**89.8%**
	9	92.6%	91.3%	90.9%	82.9%

TABLE III. HYBRID DEEP CONV-LSTM 1 SEC. AND 4 SEC. WINDOWING USING SENSOR INDEPENDENT

1-second windowing				1-second windowing	
Sensor	**Subject**	***Accuracy***			
		Angular axes	***Principle axes***	***Angular axes***	***Principle axes***
All sensors	1	90.9%	89.1%	88.1%	84.1%
	2	93.5%	89.1%	89.8%	83.9%
	3	93.5%	93.1%	85.9%	82.9%
	5	91.3%	91.8%	86.7%	83.9%
	6	94.1%	92.2%	89.1%	87.9%
	7	92.2%	92.2%	84.9%	81.5%
	8	**95.2%**	**92.3%**	**93.1%**	**89.2%**
	9	94.1%	91.5%	89.2%	80.2%

TABLE IV. 2D CNN 1 SEC. AND 4 SEC. WINDOWING USING EACH SENSOR DEPENDENT

1-second windowing				4-second windowing	
Sensor	**Subject**	***Accuracy***			
		Angular axes	***Principle axes***	***Angular axes***	***Principle axes***
Ankle	1	91.3%	86.1%	79.3%	79.2%
	2	91.1%	89.3%	78.5%	74.7%
	3	90.5%	88.5%	79.4%	71.6%
	5	93.1%	90.2%	78.6%	76.4%
	6	86.6%	89.1%	79.8%	70.9%
	7	92.1%	88.6%	86.1%	77.3%
	8	91.5%	**90.5%**	**83.3%**	**75.4%**
	9	**91.6%**	90.1%	70.4%	63.8%
Knee	1	88.6%	87.5%	**84.1%**	57.5%
	2	92.1%	83.9%	60.1%	74.3%
	3	91.8%	90.6%	82.1%	75.1%
	5	82.7%	87.3%	80.8%	78.7%
	6	89.1%	84.2%	76.8%	50.2%
	7	90.4%	88.6%	81.3%	51.2%
	8	**92.8%**	**90.5%**	82.1%	**77.9%**
	9	91.8%	89.4%	74.6%	77.9%
Trunk	1	90.8%	84.3%	83.4%	76.1%
	2	89.1%	90.9%	81.9%	69.8%
	3	92.8%	89.5%	82.1%	78.4%
	5	92.4%	83.7%	80.3%	74.5%
	6	91.9%	86.5%	80.9%	70.9%
	7	93.3%	90.9%	81.3%	86.1%
	8	92.5%	86.5%	**87.1%**	**87.2%**
	9	**96.1%**	**93.1%**	79.9%	84.3%

TABLE V. 2D CNN 1 SEC. AND 4 SEC. WINDOWING USING SENSOR INDEPENDENT

1-second windowing				4-second windowing	
Sensor	**Subject**	***Accuracy***			
		Angular axes	***Principle axes***	***Angular axes***	***Principle axes***
All sensors	1	92.6%	92.1%	87.7%	81.7%
	2	**95.1%**	92.5%	**89.8%**	80.9%
	3	93.5%	92.9%	80.1%	79.2%
	5	92.1%	93.1%	75.7%	79.8%
	6	93.6%	93.1%	79.8%	80.2%
	7	89.2%	**93.6%**	81.6%	82.1%
	8	94.6%	91.8%	87.8%	**82.6%**
	9	94.5%	92.4%	81.1%	82.4%

TABLE VI. HYBRID DEEP CONV-LSTM 1 SEC. AND 4 SEC. WINDOWING USING SENSOR DEPENDENT

1-second windowing				4-second windowing	
Sensor	**Subject**	***Accuracy***			
		Angular axes	***Principle axes***	***Angular axes***	***Principle axes***
Ankle	1	89.4%	85.3%	80.8%	74.1%
	2	90.3%	85.6%	85.1%	79.4%
	3	89.1%	84.5%	85.9%	81.2%
	5	91.9%	89.1%	85.5%	77.2%
	6	91.9%	85.7%	85.2%	77.2%
	7	88.8%	84.4%	79.1%	76.2%
	8	**94.3%**	**92.6%**	**88.2%**	**84.8%**
	9	91.7%	90.5%	84.1%	78.4%
Knee	1	89.5%	86.6%	81.4%	75.4%
	2	92.1%	86.1%	84.1%	75.5%
	3	92.1%	87.9%	83.3%	83.3%
	5	90.2%	89.1%	84.8%	80.7%
	6	92.6%	87.6%	77.6%	85.6%
	7	91.2%	89.5%	84.7%	78.4%
	8	**91.4%**	**90.3%**	**88.1%**	**83.1%**
	9	91.3%	89.3%	83.4%	79.3%
Trunk	1	91.2%	88.3%	82.8%	83.6%
	2	93.9%	88.7%	90.1%	86.7%
	3	94.2%	91.9%	86.4%	86.7%
	5	92.2%	90.1%	85.1%	82.7%
	6	93.1%	92.2%	87.9%	89.7%
	7	93.6%	86.4%	86.7%	87.2%
	8	**94.9%**	**92.9%**	**92.9%**	**91.3%**
	9	94.2%	91.4%	85.8%	84.5%

TABLE VII. HYBRID DEEP CONV-LSTM 1 SEC. AND 4 SEC. WINDOWING USING SENSOR INDEPENDENT

1-second windowing				4-second windowing	
Sensor	**Subject**	***Accuracy***			
		Angular axes	***Principle axes***	***Angular axes***	***Principle axes***
All sensors	1	92.1%	86.5%	86.2%	79.9%
	2	92.8%	86.9%	91.1%	83.1%
	3	93.1%	90.1%	88.8%	82.7%
	5	93.2%	90.5%	88.4%	83.6%
	6	93.9%	90.5%	89.2%	85.1%
	7	93.2%	89.9%	84.2%	82.3%
	8	**95.2%**	**91.9%**	**90.2%**	**90.5%**
	9	94.1%	91.8%	90.1%	85.7%

TABLE VIII. CNN WITH 1 SEC. AND 4 SEC. WINDOWING USING SENSOR DEPENDENT

1-second windowing				4-second windowing	
Sensor	**Subject**	***Accuracy***			
		Angular axes	***Principle axes***	***Angular axes***	***Principle axes***
Ankle	1	89.2%	85.6%	76.9%	73.2%
	2	83.4%	80.3%	73.6%	70.5%
	3	92.8%	89.6%	73.1%	72.4%
	5	91.6%	87.8%	81.5%	79.1%
	6	89.6%	83.4%	57.6%	50.1%
	7	91.5%	87.5%	83.1%	72.8%
	8	**92.7%**	**89.7%**	81.1%	74.8%
	9	86.7%	83.8%	**82.1%**	**75.2%**
Knee	1	88.3%	83.5%	79.8%	72.1%
	2	87.1%	84.9%	82.1%	75.8%
	3	91.1%	88.7%	82.4%	77.1%
	5	91.5%	89.1%	80.8%	76.3%
	6	89.1%	85.1%	80.2%	70.5%
	7	89.2%	85.4%	80.1%	77.8%
	8	**91.1%**	**86.3%**	**83.5%**	**80.4%**
	9	91.6%	88.5%	83.4%	80.2%
Trunk	1	89.4%	87.1%	86.1%	75.1%
	2	87.8%	86.4%	78.7%	72.9%
	3	90.2%	87.7%	87.6%	65.4%
	5	92.8%	88.6%	89.5%	67.5%
	6	93.6%	91.4%	84.5%	66.1%
	7	92.6%	88.1%	78.1%	72.3%
	8	**95.6%**	**91.6%**	78.5%	75.4%
	9	93.5%	91.1%	**78.6%**	**76.2%**

TABLE IX. CNN 1 SEC. AND 4 SEC. WINDOWING USING SENSOR INDEPENDENT

1-second windowing				4-second windowing	
Sensor	**Subject**	***Accuracy***			
		Angular axes	***Principle axes***	***Angular axes***	***Principle axes***
All sensors	1	92.1%	88.2%	88.4%	79.8%
	2	94.9%	90.5%	83.4%	80.5%
	3	94.8%	90.2%	65.8%	83.6%
	5	92.1%	85.6%	87.1%	82.8%
	6	**94.9%**	90.8%	**88.6%**	80.1%
	7	85.9%	80.9%	88.2%	78.9%
	8	93.5%	**91.1%**	87.2%	**87.1%**
	9	93.3%	88.9%	85.8%	83.7%

B. *Opportunity Dataset Results*

Results for sensor dependent, Table X shows that using angular axes features outperforms using principle-axes, as the Conv-LSTM model achieved an enhancement of 4.3%, 2.6%, and 2.4% for upper knee, hip, and lower knee sensors, respectively. Sensor-dependent results for 2D CNN, Table XI ensure also that the use of angular axes features results are better than principal axes. The results increased by 3.1%, 1.6%, and 1.2% for upper knee, hip, and lower knee sensors, respectively. For the independent sensor using angular axes features outperform the results from principle-axes with an enhancement by 3.5% and 4.1% for Conv-LSTM and 2D CNN models, respectively as shown in Table XII. From Table X and Table XI, it can be observed that the hip sensor outperforms other sensors and the performance of the 2D CNN model achieves results better than the Conv-LSTM model. Table XIII presents different deep learning models and features for the related work compared with the proposed models. Two different models are proposed for deep learning implementation, the first one is a combination of a convolutional neural network layer for 2D spectrograms images and LSTM recurrent units (Conv-LSTM). The second model used CNN also with 2D spectrogram images. Both models are implemented under the use of angular axes features instead of principle-axes. The best F-measure achieved by the proposed related work algorithms using the same Daphnet dataset was 80% on the other hand, the Conv-LSTM model achieves 95.1% F-measure and the CNN model achieves 98.1% F-measure. The enhancement here is 15.1% and 18.1% for Conv-LSTM and CNN models, respectively.

TABLE X. HYBRID DEEP CONV-LSTM USING SENSOR DEPENDENT

Sensor	**Accuracy**	
	Angular axes	***Principle axes***
Upper Knee	95.4%	91.1%
Hip	95.3%	92.7%
Lower Knee	95.5%	93.1%

TABLE XI. 2D CNN USING SENSOR DEPENDENT

Sensor	**Accuracy**	
	Angular axes	***Principle axes***
Upper Knee	96.3%	93.2%
Hip	96.2%	94.6%
Lower Knee	94.7%	93.5%

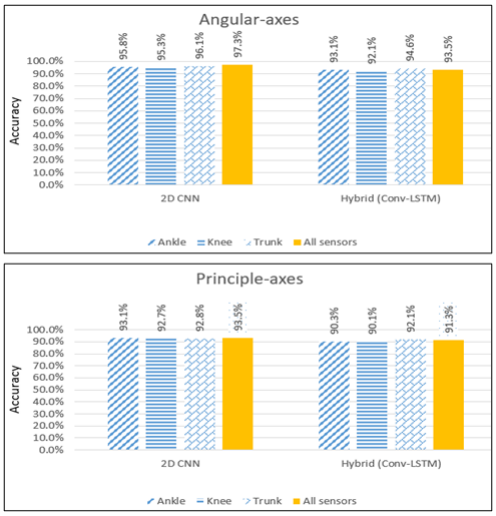

Fig. 6. Accuracy for FoG Prediction using Angular and Original Axes with 1-Second Windowing (67 Samples).

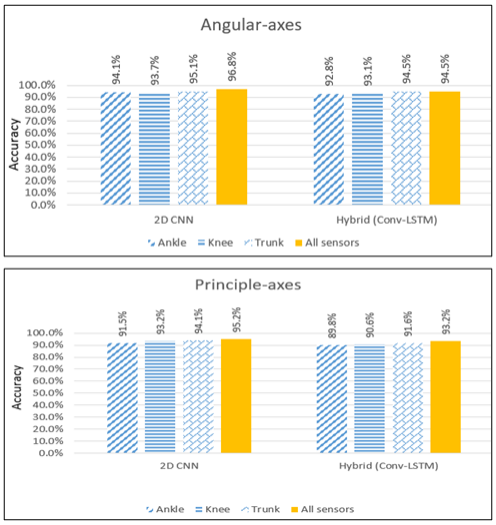

Fig. 7. Accuracy for FoG Prediction using Angular and Original Axes with 4-Seconds Windowing (268 Samples).

TABLE XII. HYBRID DEEP CONV-LSTM AND 2D CNN USING SENSOR INDEPENDENT

Model	**Accuracy**		**Pre.**	**Recall**	**F-measure**
	Angular axes	***Principle axes***	***Angular axes***		
Conv-LSTM	94.6%	91.1%	94.5%	93.1%	93.8%
CNN	97.6%	93.3%	96.5%	95.5%	95.1%

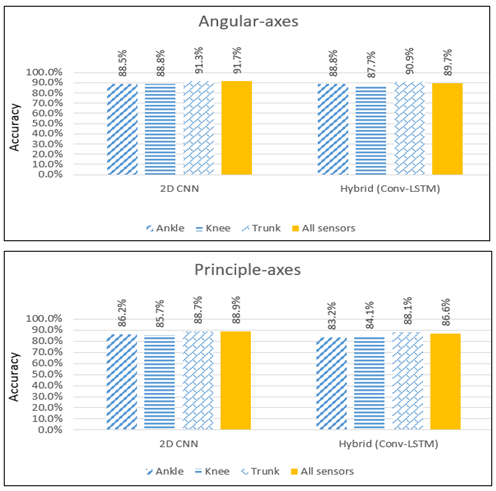

Fig. 8. Accuracy for FoG Detection using Angular and Original Axes with 1-Second Windowing (67 Samples).

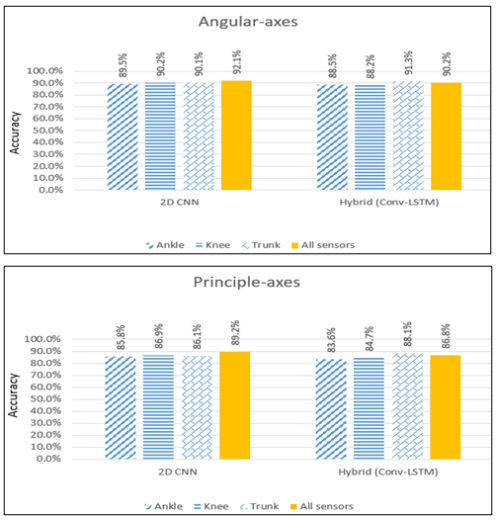

Fig. 9. Accuracy for FoG Detection using Angular and Original Axes with 4-Seconds Windowing (268 Samples).

TABLE XIII. DEEP LEARNING PERFORMANCE MEASUREMENTS COMPARED TO DIFFERENT RELATED WORK MEASUREMENTS

Ref.	Classifier	Accuracy	Precision	Recall	F-measure	Specificity
Murad & Pyun [6] 2017	unidirectional, bidirectional, and cascaded architectures based on long short-term memory (LSTM) DRNNs	94.1%	84.7%	78.9%	93%	-
Masiala [7] 2017	RNN with LSTM units with statistical features	-	-	79%	-	80%
Camps [9] 2017	Hybrid model based on convolutional layers with LSTM units	87.8%	-	88.1%	-	89.1%
Camps et al. [19] 2017	Four and five layers of 1D-ConvNet	-	–	88.6%	-	78%
Xia et al. [8] 2018	CNN	80.7%	69.29%	-	-	90.6%
Segundo et al. [10] 2018	CNN + MLP	–	–	93.1%	-	75%
Camps et al. [20] 2018	Eight 1D-ConvNet layers	89%	-	91.9%	-	89.5%
Kim et al. [21] 2018	CNN	-	-	93.8%	91.8%	90.1%
Yuan & Chakraborty [4] 2020	RNN with LSTMs and three different schemes.	94.7%	-	-	-	-
Bikias et al [5] 2021	CNN with two schemes (10-fold CV and LOSO-CV)	-	83% LOSO CV and 86% 10-CV	-	-	88% LOSO CV and 90% 10-CV
The proposed model (prediction)	**Conv-LSTM**	**93.5%**	**95.1%**	**93.5%**	**94.3%**	**95.1%**
The proposed model (detection)		**94.5%**	**94.6%**	**92.1**	**93.3%**	**94.1%**
Baseline model (prediction)	**2D CNN**	**97.3%**	**96.1%**	**97.1%**	**96.6%**	**96.5%**
Baseline model (detection)		**96.8%**	**96.3%**	**95.4%**	**95.8%**	**96.1%**

V. CONCLUSION AND FUTURE WORK

In this paper, a hybrid model of 2D convolutional layers with LSTM layers (Conv-LSTM) was proposed with the use of spectrograms as an input for the model and another baseline model was 2D CNN. The developed models are implemented using angular axes features and principle-axes sensor data readings. In this paper, two windowing methodologies are tested and applied for the Daphnet dataset. The two methodologies are a window of 67 samples, which is equivalent to 1-second, and a window of 268 samples, which

is equivalent to 4 seconds. The paper aimed to predict the FoG episodes, the most approach that achieves this aim was the use of the CNN approach for both Daphnet and Opportunity datasets as shown in Fig. 6 and Table XII, followed by adopting the hybrid Conv-LSTM approach. Several observed enhancements are achieved. First, the enhancement was achieved when using angular axes features over principle-axes. Second, using 1-second windowing against 4 seconds achieves better performance. Deep learning algorithms for predicting pre-FoG episodes, results from Table II and Table III for the Conv-LSTM model, Table IV and Table V for the CNN model ensure on using angular axes features outperforms the use of principle-axes. Deep learning results for detecting FoG episodes from Tables VI and VIII for the Conv-LSTM model and Tables VIII and IX for the CNN model also clarify that the use of the angular axes features is better than the principle-axes. In addition, the prediction results for both Conv-LSTM and 2D CNN models are better than the detection results.

Deep learning algorithms are implemented with the use of an early stopping approach to avoid the model's overfitting. All training sets are implemented for 15 epochs and 15 folds in each epoch for both datasets. It also emphasizes that the maximum number of epochs the model reached was 14 epochs, the minimum number was 2 epochs and the average number of epochs lasting in each fold was 6 epochs. The best result achieved from deep learning models was 97.6% and 93.5% for 2D CNN and hybrid Conv-LSTM respectively for the subject and sensor independent. It can be also observed that patient 8 was the subject with almost the best accuracy in subject dependent, as the patient recorded a score of 4 out of 5 on the Hoehn and Yahr (H&Y) scale, which is considered the highest score among the other 10 patients in the Daphnet dataset.

For future work, various challenges could be considered in the domain of predicting the FoG episodes via pre-fog behavior detection as well as predicting FoG severity. Working on new feature fusion sets. Build our dataset to experiment with different algorithms and features and to be compared with other datasets.

REFERENCES

[1] S. García, E. S. Castro, I. Expósito, T. De Deus, C. Tuñas, A. Aneiros, M. L. Fernández, D. N. Arias, and M. B. Torres, “Comorbid conditions associated with Parkinson's disease: a longitudinal and comparative study with Alzheimer disease and control subjects,” Journal of the neurological sciences, vol. 373, pp. 210-215, 2017.

[2] A. H. V. Schapira, K. R. Chaudhuri, and P. Jenner, “Non-motor features of Parkinson disease,” Nature Reviews Neuroscience, vol. 18, no. 7, pp. 435-450, 2017.

[3] H. Elziaat, N. El-Bendary, and R. Moawad, “Multi-Feature Fusion and Machine Learning: A Model for Early Detection of Freezing of Gait Events in Patients With Parkinson's Disease,” Handbook of Research on Automated Feature Engineering and Advanced Applications in Data Science, edited by Mrutyunjaya Panda and Harekrishna Misra, IGI Global, pp. 95-118, 2021.

[4] A. M. Yuan and S. Chakraborty, “A Study of Deep Learning for Predicting Freeze of Gait in Patients with Parkinson’s Disease,” IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1324-1331, 2020.

[5] T. Bikias, D. Iakovakis, S. Hadjidimitriou, V. Charisis, and L. J. Hadjileontiadis, “DeepFoG: An IMU-Based Detection of Freezing of Gait Episodes in Parkinson’s Disease Patients via Deep Learning,” Frontiers in robotics and AI, vol. 8, 2021.

[6] A. Murad and J. Y. Pyun, “Deep Recurrent Neural Networks for Human Activity Recognition,” Sensors, vol. 17, no. 11, pp. 2556, November 2017.

[7] S. Masiala, “Detection of Freezing of Gait in Patients with Parkinson’s Disease Using Deep Recurrent Neural Networks,” PhD diss., Tilburg University, 2017.

[8] Y. Xia, J. Zhang, Q. Ye, N. Cheng, Y. Lu, and D. Zhang, “Evaluation of deep convolutional neural networks for detection of freezing of gait in Parkinson’s disease patients,” Biomedical Signal Processing and Control, vol. 46, pp. 221-230, 2018.

[9] J. Camps, “Applying deep-learning techniques to detect freezing of gait episodes in Parkinson's disease patients,” Master's thesis, Universitat Politècnica de Catalunya, 2017.

[10] R. S. Segundo, H. N. Hellín, R. Torres, J. Hodgins, and F. D. L. Torre, “Increasing Robustness in the Detection of Freezing of Gait in Parkinson’s Disease,” Electronics, vol. 8, no. 2, pp. 119, 2019.

[11] M. Ba¨chlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. T ro¨ster, “Wearable assistant for Parkinson’s disease patients with the freezing of gait symptom,” IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 2, pp. 436-446, 2009.

[12] R. Chavarriaga, H. Sagha, A. Calatroni, S. Digumarti, G. Tröster, J. D. R. Millán, and D. Roggen, “The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition”, Pattern Recognition Letters, vol. 34, no. 15, pp. 2033-2042, 2013.

[13] H. A. Hashim, “Special orthogonal group SO (3), euler angles, angle-axis, rodriguez vector and unit-quaternion: Overview, mapping and challenges,” arXiv preprint arXiv:1909.06669, 2019.

[14] I. A. Faisal, T. W. Purboyo, and A. S. Raharjo Ansori, “A Review of accelerometer sensor and gyroscope sensor in IMU sensors on motion capture,” J. Eng. Appl. Sci, vol. 15, no. 3, pp. 826-829, 2019.

[15] M. Njirjak, E. Otović, D. Jozinović, J. Lerga, G. Mauša, A. Michelini, and I. Štajduhar, “The Choice of Time–Frequency Representations of Non-Stationary Signals Affects Machine Learning Model Accuracy: A Case Study on Earthquake Detection from LEN-DB Data,” Mathematics, vol. 10, no. 6, pp. 965, 2022.

[16] L.Wang, C.Wang, and Y. A. Chen, “Fast three-dimensional display method for time-frequency spectrogram used in embedded fault diagnosis devices,” Applied Sciences, vol. 8, no. 10, pp. 1930, 2018.

[17] M. Panwar, S. R. Dyuthi, K. C. Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, and G. R. Naik, “Cnn based approach for activity recognition using a wrist-worn accelerometer,” In Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE, pp. 2438-2441, 2017.

[18] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino, “How deep learning extracts and learns leaf features for plant classification,” Pattern Recognition, vol. 71, pp. 1-13, 2017.

[19] J. Camps, A. Samà, M. Martín, D. Rodríguez-Martín, C. Pérez-López, S. Alcaine, B. Mestre, A. Prats, M. C. Crespo, J. Cabestany, À. Bayés, and A. Català, “Deep Learning for Detecting Freezing of Gait Episodes in Parkinson’s Disease Based on Accelerometers,” In International Work-Conference on Artificial Neural Networks, pp. 344-355, 2017.

[20] J. Camps, A. Sama, M. Martin, D. Rodriguez-Martin, C. Perez-Lopez, J. M. M. Arostegui, J. Cabestany, A. Català, S. Alcaine, B. Mestre, A. Prats, M. C. Crespo-Maraver, T. J. Counihan, P. Browne, L. R. Quinlan, G. Ó Laighin, D. Sweeney, H. Lewy, G. Vainstein, A. Costa, and R. Annicchiarico, “Deep learning for freezing of gait detection in Parkinson’s disease patients in their homes using a waist-worn inertial measurement unit,” Knowledge-Based Systems, vol. 139, pp. 119-131, 2018.

[21] H. B. Kim, H. J. Lee, W. W. Lee, S. K. Kim, H. S. Jeon, H. Y. Park, C. W. Shin, W. J. Yi, B. Jeon, and K. S. Park, “Validation of Freezing-of-Gait Monitoring Using Smartphone,” Telemedicine and e-Health, vol. 24, no. 11, pp. 899-907, 2018.