Leveraging Machine Learning Techniques to Predict Freezing of Gait

Wohsumoe Soyazhe Khan, Raj Senthilkumar, Emily Anne Steiner, Mahad Zaryab
SYDE 522 – Group 22, Option 2
Department of Systems Design Engineering
University of Waterloo, Canada
{wskhan, r2senthi, easteine, mzaryab}@uwaterloo.ca

***Abstract*—Freezing of Gait (FoG) is a common phenomenon that occurs in patients diagnosed with Parkinson's Disease (PD). This phenomenon results in episodes in which patients are unable to move forward despite having the intention to do so. This paper explores the prediction of the onset of FoG through the use of on-body accelerometers from the Daphnet dataset, using a series of five machine-learning techniques. These techniques are (1) Support Vector Machine (SVM); (2) Random Forest; (3) AdaBoost; (4) Convolutional Neural Network (CNN); and (5) ProtoNN. The raw accelerometer data was pre-processed into windows of varying lengths and 90 features were extracted before being fed into the classifiers. To test the effectiveness of each classifier in predicting the onset of FoG, a series of tests were run with smaller feature subsets. In evaluating the effectiveness of the classifiers, the primary metrics that were examined were the F1-score of the model on the test dataset and the model's size. Overall, the SVM classifier performed the best with an F1-score of 99.04% while ProtoNN had the smallest model size of 2.9 kB.**

***Index Terms*—Freezing of Gait, Daphnet, Support Vector Machine, Random Forest, AdaBoost, CNN, ProtoNN**

I. Introduction

Freezing of Gait (FoG) is a common clinical phenomenon observed in patients with advanced Parkinson's disease (PD) that significantly impairs mobility, often leading to falls and reduced quality of life [1]. FoG is characterized by episodes in which patients intend to walk but cannot move their feet [1].

FoG is an episodic disruption during the normal flow of walking, with episodes generally lasting a few seconds, but severe episodes can exceed 30 seconds [1]. The timing and duration of FoG episodes are highly variable and personalized, making observation and classification difficult. Furthermore, the complexity of the disease has prevented a universally accepted definition of FoG.

Based on clinical findings, FoG can be classified into three different patterns: (1) trembling in place of the knees; (2) short, forward shuffling steps; and (3) no movement of the limbs or trunk [1]. Understanding these patterns can aid in diagnosing and treating FoG, which can greatly improve the quality of life for patients with PD.

During normal walking, several physical symptoms precede and occur during an FoG episode, including a decrease in stride length, reduced joint ranges in the hip, knee, and ankle, abnormal stepping patterns, and trembling leg movements at a high frequency [1]. These symptoms can be detected using sensors placed on the trunk and extremities of Parkinson's disease patients. The collected data can then be used to predict future FoG episodes.

In a clinical setting, there are several methods for predicting the onset of FoG. One method uses Inertial Measurement Unit (IMU) sensors or accelerometers affixed to patients while they perform prescribed tasks, such as walking on a treadmill or navigating a given path. Another method uses pressure sensors placed under the patient's feet to collect pressure maps during standing and walking. A third method uses motion capture systems such as Vicon to track extremities and gait using tracking dots placed on the patient [2].

Early prediction of FoG is crucial as it is one of the most common and disabling symptoms of Parkinson's disease. Early detection allows for early intervention, which can greatly improve the quality of life for patients. One way to counteract FoG is by applying various stimuli, such as tactile vibrations or auditory rhythmic cues, to unfreeze patients when a freezing episode is detected, thereby enabling them to regain control [3]. As FoG is often resistant to other pharmaceutical treatments, this approach is increasingly becoming a critical area of research [3].

This paper analyzes the prediction of the onset of FoG using machine-learning (ML) techniques, including Support Vector Machines (SVM), ensemble methods, and neural networks. To provide context for this problem and highlight the limitations of existing approaches, previous research in the field of FoG detection is discussed in Section II. In Section III, five specific techniques are chosen to investigate further and compare. The background of the dataset used, and the pre-processing methods employed are outlined in Section IV. The results and analysis are also reported in Section IV. Finally, conclusions about the empirical evaluation are drawn in Section V.

II. Background

One of the first breakthroughs in understanding the clinical phenomenon of FoG was a research study by Moore et al. [4]. This study found a key relationship in power emitted by high-frequency components of leg movement during FoG from accelerometers placed on the body. This relationship was dubbed the Freeze Index (FI) and is defined as the power in the 'freeze' band (3-8 Hz) divided by the power in the 'locomotor' band (0.5-3 Hz) [4]. Values above a certain threshold allowed FoG to be correctly classified with reasonable accuracy. The FI remains the most important feature leveraged by other algorithms to classify FoG correctly.

Two years after the discovery of the FI, the first automatic detection algorithm for FoG was developed by Bächlin et al. [5]. This algorithm, known as the Moore-Bächlin FoG

Algorithm, or MBFA for short, is built off of Moore's FI, emphasizing low latency. The algorithm leverages a sliding window approach to lower the latency of Moore's algorithm to make it suitable for on-device detection [5]. MBFA still serves as a baseline for benchmarking the performance of new algorithms.

The data collected by Bächlin et al. during the development of the MBFA algorithm is referred to as the "Daphnet Dataset" and has become the most widely used dataset in literature for FoG [5]. The freezing events experienced by patients ranged from 0.5 seconds to 40.5 seconds, with most freezing events lasting less than 5.4 seconds [5]. Accelerometer data is recorded from the ankles, thighs and trunk locations. Figure 1 shows raw sensor readings from these locations before, during, and after a freeze event. This visualization highlights the challenges in detecting freeze events as the change in acceleration pattern is not intuitively clear, and sensors are noisy. In addition, freeze events must be distinguished from intentional stopping. This further highlights the importance of AI in predicting and detecting FoG, as discussed in Section I.

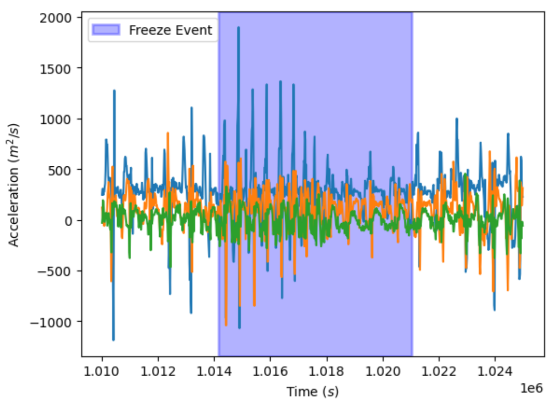

Fig. 1. Acceleration data collected during a freeze event.

To evaluate the collected IMU data, further work has gone into developing an extensive feature set to make the problem of detecting FoG suitable for machine learning classifiers. The first feature set for this domain was developed by Mazilu et. al [6]. In the time domain, the feature set contained the mean, standard deviation and variance of a given window [6]. In the frequency domain, the feature set contained the entropy, energy, freeze index, and power of a given window [6]. While the feature set developed by Moore et al. is widely used, optimizing the feature set for detecting FoG is an active area of research: [7], [8], [9], [10].

The most common approach mentioned in literature for detecting an FoG event is SVM with a radial basis function (RBF) kernel [11] [12]. SVM has been shown to perform better than MBFA in terms of sensitivity and specificity due to SVM's ability to consider multiple features in comparison to MBFA, which only considers two [11]. Furthermore, SVM has the benefit of generalizing as a machine learning method, whereas MBFA can only consider two thresholds [11]. However, it is important to note that SVMs have some limitations. For instance, they may struggle to handle high-dimensional feature spaces, which can be problematic when working with large datasets. Additionally, SVMs may be sensitive to noise and outliers, impacting their accuracy when detecting FoG.

Another common approach in literature for detecting an FoG event is ensemble methods, specifically using decision trees. One study used a combination of decision tree models, including Random Forest and AdaBoost, to detect FoG and achieved a high accuracy of 98.35% [6]. Another study used Extreme Gradient Boosting with decision trees to detect FoG and achieved an accuracy of 86.0% [13]. These results suggest that ensemble methods using decision trees can be effective for FoG classification. However, further research is needed to compare their performance with other machine learning models and ensemble strategies. Like any other method, ensemble methods have their limitations. For instance, random forests are usually reliable and precise, but they may face challenges with imbalanced datasets, a frequent issue in FoG detection [5]. Additionally, random forests can overfit, leading to poor generalization performance. Moreover, AdaBoost can be computationally intensive and may not be appropriate for real-time FOG detection applications.

The breakthroughs in predicting the onset of FoG using machine learning classifiers have prompted research into stimulating patients to aid recovery from an episode. As a result, the machine learning classifiers often need to be run on embedded platforms that serve as recovery systems; hence, classifications must happen on the portable device. Gokul H. et al. developed a gait recovery system using two resource-efficient algorithms: ProtoNN and Bonsai [14]. Both algorithms are developed from traditional machine learning algorithms but have a lower storage complexity to make them suitable for deployment on embedded devices.

Convolutional neural networks (CNN) are also used for detecting FoG onsets as an exploratory endeavour. CNNs are deep learning networks that forcibly connect between local neurons of adjacent layers to find spacial patterns. This makes them useful for extracting information from 2D data, such as images, where spatial patterns can be found and extracted from images. This can also be applied to finding patterns in time-series data to find trends such as IMU data mapping motion. The ideal application of CNN in this FoG detection would be to pattern-match irregularities in the gate. Bikias et al. have had some success in identifying freezing moments using this technique [15]. One of the major limitations of CNNs is the significant computational resources required. This can render them unsuitable for real-time FOG detection applications. Additionally, CNNs can be sensitive to noise in the input data, which can negatively impact their performance in FOG detection. Given that gait data can be prone to noise, this can be a substantial constraint for CNNs in this particular application.

III. Method

This paper explores four different classes of classifiers to evaluate their performance in classification tasks: (1) Support Vector Machines to represent traditional machine learning classifiers; (2) Random Forest and AdaBoost to represent ensemble methods; (3) Convolutional Neural Network to represent deep learning approaches; (4) ProtoNN to represent a class of resource-efficient classifiers.

*1) **Support Vector Machines (SVM):*** Support Vector Machines are supervised learning models which learn hyperplanes separating classes of data by maximizing the margin. In non-linear cases, SVMs use kernels to project lower dimensional data to higher linearly separable dimension spaces. SVMs are a common machine learning approach to FoG detection research due to their high performance on binary classification problems and effective implementation on microcontrollers. This paper includes SVM model implementation using the radial basis function (RBF) kernels based on previous work by a BarcelonaTech research group Rodríguez-Martín et al. in 2017 [11]. Implementation was modified using scikit-learn to use data from the Daphnet dataset and the common pre-processing steps described. This ensured the AI methods implemented could be fairly bench-marked and compared.

Two hyperparameters were considered: (1) C controls the amount of regularization and controls the balance between maximizing the margin versus maximizing the number of correct training classifications, and (2) γ controls the sphere of influence of individual data points. The hyperparameters were tuned using the training set for varying window lengths. Optimal hyperparameters were found using a custom scoring function maximized in Equation 1 [11]. Note that if the conditions are not met regarding sensitivity or specificity, the score is set to zero.

$$C_{opt}, \gamma_{opt} = argmax(\sqrt{Sensitivity_{C,\gamma} * Specificity_{C,\gamma}}) \\ \text{s.t. } Sensitivity_{C,\gamma} > 70\% \; Specificity_{C,\gamma} > 70\% \quad (1)$$

*2) **Ensemble Methods - Random Forest:*** Random forest is a machine learning algorithm that is used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make predictions [16]. Each decision tree is trained independently, and the final prediction is made by majority vote (for classification problems) of the predictions made by the individual trees [16].

The model was implemented using the scikit-learn library, from scratch. Table I portrays a summary of the hyperparameters that are used in the training of the Random Forest model.

TABLE I
SUMMARY OF HYPERPARAMETERS FOR THE TRAINING OF A RANDOM FOREST MODEL

Hyperparameter	**Description**
n_estimators	number of trees in forest
criterion	function to measure quality of split
max_depth	maximum depth of tree
min_samples_split	minimum samples required to split internal node
min_samples_leaf	minimum samples required to be at a leaf node
max_features	number of features to consider while splitting

To ensure that the model could be trained effectively, an overfitting check was performed before running the complete training algorithm; a very low number of trees, each with high individual depth, was used without any bootstrapping or over-sampling. The hyperparameters were fine-tuned using a random search, and the training process was executed for various window lengths. Once the model was trained, it was run on the testing set, and essential metrics were extracted for analysis.

*3) **Ensemble Methods - AdaBoost:*** AdaBoost (short for Adaptive Boosting) is an ensemble machine learning method that combines multiple "weak" learners to create a strong classifier [17]. Each weak classifier focuses on the examples that were misclassified by the previous classifiers. The weight of each example is adjusted at each iteration based on whether it was classified correctly or incorrectly by the previous classifiers. The final prediction is made by combining the predictions of all the weak classifiers [17].

The model was implemented using the scikit-learn library, from scratch. The only hyperparameters that were used in the training of this model were the number of estimators (decision tree) in the ensemble, and the learning rate which represents the weight applied to each classifier at each boosting iteration.

Next, the hyperparameters were fine-tuned using a random search, and the training process was executed for various window lengths. Once the model was trained, it was run on the testing set, and essential metrics were extracted for analysis.

*4) **Convolutional Neural Networks:*** Convolutional neural networks is a deep learning method that utilizes kernels to find patterns in data using convolution rather than just pure matrix multiplication [18].

The structure of the CNN used in this paper is a 3 layer network that utilizes 2 hidden convolution layers with 64 nodes each. A 50 percent drop out is applied in-between each layer in order to reduce overfitting. The output from the convolution layers are pooled and then flattened into a 1-node output to give label classifications. Two hidden layers were used in this model because this number is commonly used in literature with NN for gait analysis [15] [19].

The convolutional neural network developed for this paper is summarized visually in Figure 2.

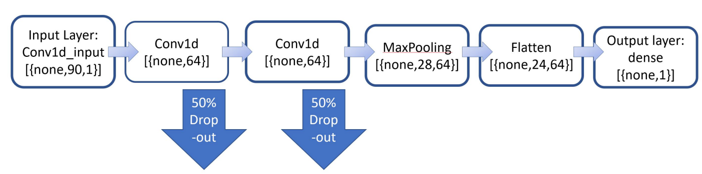

Fig. 2. Visualization of CNN model.

For the hyperparameter turning, the number of neurons and activation function was explored. A search covering 16, 32, and 64 neurons with activation functions ReLU, sigmoid, and tanh was used. After the best model was chosen based on specificity and sensitivity, the testing dataset was run the model and results extracted for later analysis.

*5) **ProtoNN:*** ProtoNN is a novel supervised machine learning algorithm based that's designed for deployment on small and resource-scarce devices [20]. ProtoNN is built off the k-nearest neighbors (kNN) algorithm but has several orders lower storage and prediction complexity [20].

ProtoNN is able to reduce its complexity while maintaining high accuracy using three key ideas: (1) learning a small number of prototypes to represent the entire training set; (2) projecting the data into a sparse low-dimensional space; (3)

adding an explicit model size constraint using discriminative learning [20].

The main hyperparameters that were tuned were $\hat{d}$, the projection dimension, and m, the number of prototypes. These parameters are especially important as they directly affect the size of the model.

Prior to running the entire training algorithm, two sanity checks were performed. The first sanity check that was done was the regularization check. In this check, the regularization strengths R_Z, R_B, R_W were increased to ensure that the total loss was increased as well. The second sanity check that was done was the overfitting check. In this check, the regularization strengths R_Z, R_B, R_W were set to zero for a small subset of the training data to ensure that overfitting would occur.

IV. Results & Analysis

A. Dataset

The dataset that was used for all empirical evaluation was the Daphnet dataset. As mentioned in Section II, it is the most widely used dataset in literature for FoG [5]. The Daphnet dataset was recorded in a lab setting to benchmark automatic methods for recognizing gait freeze with wearable acceleration sensors. Participants completed tasks including straight-line walking, walking with turns, and activity of daily living (ADL) tasks. The dataset contains data from 16 Parkinson's disease patients, each undergoing three 10-minute sessions [5].

B. Pre-processing & Feature Extraction

The pre-processing stage used in this paper's experiments was based on the work of Gokul H. et al. [14]. During this stage, the data collected from each accelerometer channel was stored as a one-dimensional window, each containing $w \times f_s$ time-steps. Raw IMU data is filtered using Butterworth and impulse response filters. The window length, denoted by w, and the sampling frequency, denoted by f_s, were set to 1 to 4 and 64 Hz, respectively, for the experiments conducted in this paper. This resulted in window sizes ranging from one to four seconds.

Windowing allows time-series data to be broken down into distinct samples in which features are then extracted within the corresponding window. Once windowed and processed, samples can be shuffled and split regardless of the time series. Increasing the window size in data analysis can capture more data and strengthen extracted trends, while also reducing noise. However, this comes at the cost of increased system latency. Alternatively, if the window size becomes too large, it can act as a low-pass filter and compromise the accuracy of the results. Since in FoG detection, response time is critical for intervention, the window sizes explored were limited to 4 seconds.

For the experiments conducted in this paper, five time domain (Refer to Table II) and five frequency domain (Refer to Table III) features were extracted from each window.

Furthermore, it was discovered that there was a significant disparity in the amount of data within each class. The number of indicated freezing events compared to the amount of normal gait. To overcome this issue of class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was also employed in the pre-processing stage. The pre-processed data was then split into training and testing sets in a 70:30 ratio, with stratification and shuffling to ensure a balanced representation.

TABLE II
A SUMMARY OF THE TIME DOMAIN FEATURES EXTRACTED FROM THE WINDOWS [14].

Name	Description
Mean	The average value of the signal
Variance	The average value of the squared differences of the signal from the mean
Standard Deviation	The square root of the variance
Root Mean Square	Square root of the mean of the squared signal
Mean Absolute Value	Mean of the absolute value of the signal in the window

TABLE III
A SUMMARY OF THE FREQUENCY DOMAIN FEATURES EXTRACTED FROM THE WINDOWS [14].

Name	Description
Entropy	Measure of the randomness of the frequency components
Energy	Sum of squared magnitudes divided by the window length
Peak Frequency	Maximum frequency in the power spectrum
Freeze Index	Power of signal in freeze band (3-8 Hz) divided by power in the locomotor band (0.5-3 Hz)
Band Power	Sum of power in freeze band and in locomotor band

C. Metrics For Evaluation

The metrics used to do empirical evaluation are tabulated in Table IV.

TABLE IV
SUMMARY OF METRICS USED TO EVALUATE MACHINE LEARNING MODELS.

Metric	Description
precision	the fraction of true positive predictions out of all the positive predictions
recall	the fraction of true positive predictions out of all the actual positive cases in the dataset
F1-score	the harmonic mean of precision and recall. It is a balanced measure that combines both precision and recall
accuracy	the fraction of correct predictions out of all the predictions made by the model
model size	the size of the model in kB

D. Overall Results

The results of models, trained with all 90 time and frequency features, are shown in Table V. The F1-scores reflect that a reduction of dimensions results in lower scores. Note that the other metrics evaluated, including test accuracy, precision, and recall, produced very similar trends and results to the F1-score and, thus, are omitted from Table V. Furthermore, there was no significant difference between each test's training and testing accuracy, indicating that the models did not overfit the training data for any window size.

Overall, all models performed very well, all trials resulting in a test F1-score of over 90%. The best performance marginally occurred at three and four-second windows, with slight variations between the different models.

TABLE V
SUMMARY OF RESULTS FROM TRAINING MODELS ON DAPHNET DATASET FOR WINDOWING SIZES $\in 1, 2, 3, 4$ SECONDS.

Classifier	F1-score (%)			
	w = 1	w = 2	w = 3	w = 4
SVM	96.60	98.15	99.13	99.04
Random Forest	96.35	97.62	98.60	98.35
AdaBoost	91.94	93.92	96.94	97.40
CNN	90.73	92.96	95.31	95.43
ProtoNN	90.86	92.58	94.71	94.53

E. Hyperparameter Tuning

*1) **SVM**:* The 10-fold cross-validation grid search was used to find optimal hyperparameters across a logarithmic grid from 10^{-2} to 10^{2} for each hyperparameter and each windowing size. This range was suggested by Rodríguez-Martín et al. and mentioned as reasonable search values in the scikit-learn documentation [11]. The results showed that within the defined search grid, there was no large effect on the results for either C or γ. The best parameters were $\gamma = 0.1$ and $C = 10^1, 10^2$. Since the mode of the C hyperparameter resulted in a tie, $C = 10^{1.5}$ was used for future tests.

Overall, the SVM models performed extremely well. The test F1-score for each window size is summarized in Table V. The model performed marginally best on a window size of three seconds.

*2) **Ensemble Methods**:* To find optimal hyperparameters for both Random Forest and AdaBoost, a 5-fold cross-validation random search with 50 iterations was conducted for each windowing size.

For Random Forest, the best parameters were selected by taking the mode of the parameters from the results of each windowing size. If there was a tie, an average discrete value was chosen. The optimal hyperparameters for Random Forest were determined to be: n_estimators = 300, criterion = gini, max_depth = 40, min_samples_split = 2, min_samples_leaf = 1, and max_features = log2.

The same method was used to determine the best parameters for AdaBoost as for Random Forest. The optimal parameters for AdaBoost were: n_estimators = 800 and learning_rate = 1.25.

The ensemble methods performed extremely well overall, as evidenced by the test F1-scores for each window size shown in Table V.

*3) **CNN**:* The result of tuning the CNN's hyperparameters was that the activation function $tanh$ and 64 neurons performed the best. The highest test accuracy was achieved with a window size of 4 seconds.

*4) **ProtoNN**:* During the tuning process for ProtoNN, $\hat{d}$ (projection dimension) and m (number of prototypes) were found to be the most critical hyperparameters. Table II displays a table of differences to show the effect of the two hyperparameters on the F1-score. As we can see, the projection dimension hyperparameter significantly impacts the model size and a sufficient number of prototypes are required to obtain accurate results.

TABLE VI
A TABLE OF DIFFERENCES TO SHOW THE IMPACT OF $\hat{d}$ AND M ON THE F1-SCORE. THE CHOSEN VALUES FOR THE HYPERPARAMETERS ARE BOLDED.

$\hat{d}$	m	F1-score (%)	Model Size (kB)
1	10	90.21	0.48
5	10	93.47	2.08
10	10	94.23	4.08
50	10	93.08	20.08
5	1	36.88	1.828
5	5	94.27	1.94
5	20	94.75	2.36
5	**40**	**95.02**	**2.92**

F. Feature Reduction Results

Feature reduction tests were performed to investigate the high-performing models using all 90 features. Several strategies were employed, considering the intended application for personal wearable technology. Reducing the number of features and smaller models ensures a more efficient microcontroller implementation. Although consideration for system latency guided this investigation, model latency was not directly measured as it would be microcontroller specific.

First, only the use of time domain features was considered. The benefit of this method is it does not use the more computationally heavy frequency domain features. These features, in the application, would require the use of the Fast Fourier Transform (FFT) and subsequent computation of informative frequency features.

Next, the features were limited to only frequency domain features. Although more computationally expensive, the model can use important frequency domain features such as Freeze Index while removing the computation of time-domain features.

Finally, Principal Component Analysis (PCA) was explored as a dimensionality reduction method. A plot of the cumulative sum of the explained variance ratio was created during the empirical evaluation of the effects of PCA dimensionality reduction, as shown in Figure 3. Based on the plot, it was decided that 15 principal components provided a reasonable balance between reducing dimensionality and preserving the explained variance of the information. This point is indicated with the vertical line. Although this method does not eliminate the need for all mathematical computations at runtime, it lowers the number of features trained, which affects the model size and can help continue to ensure no overfitting occurs.

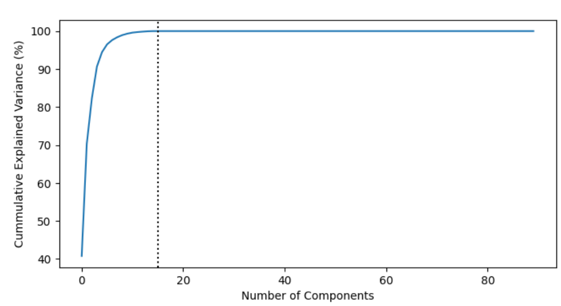

Fig. 3. Cumulative sum of the explained variance ratio of principal components.

From the initial investigation into hyperparameter tuning for the feature-reduced results, optimal hyperparameters were relatively constant for window size. Thus, hyperparameters were kept constant for the remainder of the tests, as found in Section IV-E. The tests were performed across all window sizes but provided similar trends; a window length of four seconds was reported.

The results of the feature reduction techniques on F1 Score across all ML techniques are observed in Figure 4. All models performed well across the dimensionality reduction results. For most models, the PCA-reduced and exclusively Frequency-domain features were comparable to the full feature set. An exception is the AdaBoost method, whose performance decreased across the feature reduction tests.

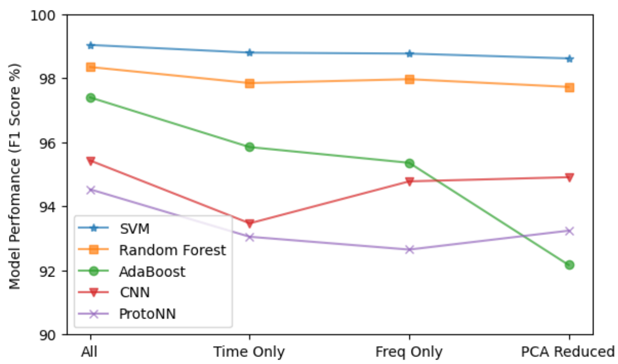

Fig. 4. The impact of different feature sets on the model performance measured with F1-score.

In addition to comparing model size between techniques and with feature reduction, the model size in kilobytes was also considered for all tests. The results are shown in Figure 5, where a logarithmic scale is used for the model size axis, as there is a significant variation in sizes across ML techniques. While reducing the number of features can lower the model size, the variation is minimal compared to the difference in model size between different ML techniques. ProtoNN stands out as requiring the least kilobytes by a large margin.

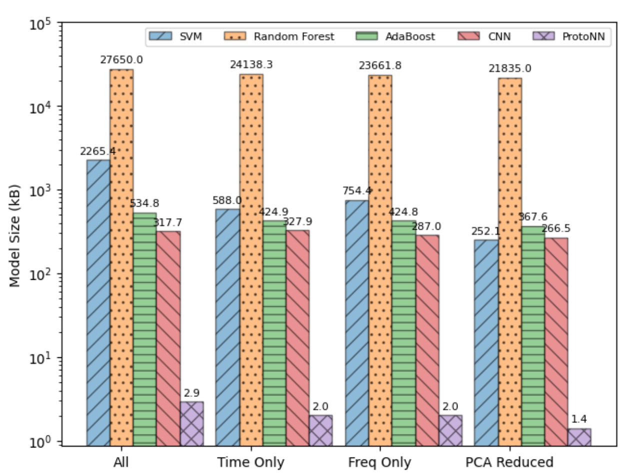

Fig. 5. The impact of different feature sets on the model size.

G. *Limitations*

The above methods achieved F1-scores above 90%, indicating high reliability in model post-processing labelling. However, in reality, these models will be implemented on devices are will need to classify data in real-time. Literature suggests that visual intervention to prevent falls must occur within 0.39 seconds of fall initiation [21]. As discussed earlier, this paper did not explore the time required for classification due to limitations in access to microcontroller implementation. The full system latency during classification and testing using on-device real-time data should be further explored.

In addition, the dataset used for this investigation includes IMUs attached to the shank, thigh, and trunk on one side of the patient's body. However, several other studies use different locations for IMU placement, making it challenging to compare results to other results in literature [15] [22] [23].

V. Conclusions

This paper examined a series of five machine-learning techniques to predict the onset of FoG. The results portrayed that SVMs perform the best with an F1-score of 99.04%. A series of tests conducted with reduced dimensionality portrayed that not all 90 features are necessary, as SVM reported an F1-score of 98.62% using 15 principal components.

Regarding memory constraints, ProtoNN had the smallest model size of 2.92 kB while maintaining a healthy F1-score of 94.53%. The other classifiers discussed in this report reported significantly higher model sizes. These results show that ProtoNN is a suitable classifier for applications in which resources are scarce, such as gait-recovery devices.

Future research into FoG detection should further investigate data collection, splitting, and processing. The Daphnet dataset was collected using laboratory-induced freezing. Expanding the data to include additional datasets of more naturalistic settings would provide better performance indicators for the intended application. In addition, test-train splitting could be adapted to reflect intended applications. Instead of training on all patients available, leave-one-out splits would better evaluate the performance of a universal model, which could be deployed on patients with personal freeze characteristics never seen during training. Finally, different feature reduction techniques, such as Recursive Feature Elimination (RFE), should be explored to determine the optimal set of reduced features to reduce the initial feature computation required.

While the classifiers studied in this paper reported excellent results, the problem cannot be declared as solved. Simply distinguishing between an FoG event and a non-FoG event is insufficient for improving patients' quality of life with Parkinson's Disease. As a result, the classifiers discussed in this paper can be enhanced to predict the preceding occurrence of FoG much before the patient is cognisant of its onset. One approach to solving this problem would be to leverage a more enhanced technique, such as transformers, to perform early prediction of FoG [24]. Another approach to solving this problem could be to look at more enhanced feature sets, such as the sets used in [7], [8], [9], and [10].

References

[1] John G Nutt, Bastiaan R Bloem, Nir Giladi, Mark Hallett, Fay B Horak, and Alice Nieuwboer. Freezing of gait: moving forward on a mysterious clinical phenomenon. *The Lancet Neurology*, 10(8):734–744, 2011.

[2] Yao Guo, Jianxin Yang, Yuxuan Liu, Xun Chen, and Guang-Zhong Yang. Detection and assessment of parkinson's disease based on gait analysis: A survey. *Frontiers*, 08 2022.

[3] Jessica M. Baker. Gait disorders. *The American Journal of Medicine*, 131(6):602–607, 2018.

[4] Steven T. Moore, Hamish G. MacDougall, and William G. Ondo. Ambulatory monitoring of freezing of gait in parkinson's disease. *Journal of Neuroscience Methods*, 167(2):340–348, 2008.

[5] Marc Bächlin, Jeffrey Hausdorff, Daniel Roggen, Nir Giladi, Meir Plotnik, and Gerhard Tröster. Online detection of freezing of gait in parkinson's disease patients: A performance characterization. *BODYNETS 2009 - 4th International ICST Conference on Body Area Networks*, page 11, 04 2009.

[6] Sinziana Mazilu, Michael Hardegger, Zack Zhu, Daniel Roggen, Gerhard Tröster, Meir Plotnik, and Jeffrey M. Hausdorff. Online detection of freezing of gait with smartphones and machine learning techniques. In *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*, pages 123–130, 2012.

[7] Albert Samà, Daniel Rodríguez-Martín, Carlos Pérez-López, Andreu Català, Sheila Alcaine, Berta Mestre, Anna Prats, M. Cruz Crespo, and Àngels Bayés. Determining the optimal features in freezing of gait detection through a single waist accelerometer in home environments. *Pattern Recognition Letters*, 105:135–143, 2018. Machine Learning and Applications in Artificial Intelligence.

[8] Thomas Bikias, Dimitrios Iakovakis, Stelios Hadjidimitriou, Vasileios Charisis, and Leontios J Hadjileontiadis. DeepFoG: An IMU-Based detection of freezing of gait episodes in parkinson's disease patients via deep learning. *Front Robot AI*, 8:537384, May 2021.

[9] Julià Camps, Albert Samà, Mario Martín, Daniel Rodríguez-Martín, Carlos Pérez-López, Joan M. Moreno Arostegui, Joan Cabestany, Andreu Català, Sheila Alcaine, Berta Mestre, Anna Prats, Maria C. Crespo-Maraver, Timothy J. Counihan, Patrick Browne, Leo R. Quinlan, Gearóid Ó Laighin, Dean Sweeney, Hadas Lewy, Gabriel Vainstein, Alberto Costa, Roberta Annicchiarico, Àngels Bayés, and Alejandro Rodríguez-Molinero. 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*, 139:119–131, 2018.

[10] Luis Sigcha, Nelson Costa, Ignacio Pavon, Susana Costa, Pedro Arezes, Juan Manuel Lopez, and Gruillermo De Arcas. Deep learning approaches for detecting freezing of gait in parkinson's disease patients through on-body acceleration sensors. *Sensors*, 04 2020.

[11] Daniel Rodríguez-Martín, Albert Samà, Carlos Pérez-López, Andreu Català, Joan M. Moreno Arostegui, Joan Cabestany, Àngels Bayés, Sheila Alcaine, Berta Mestre, Anna Prats, M. Cruz Crespo, Timothy J. Counihan, Patrick Browne, Leo R. Quinlan, Gearóid ÓLaighin, Dean Sweeney, Hadas Lewy, Joseph Azuri, Gabriel Vainstein, Roberta Annicchiarico, Alberto Costa, and Alejandro Rodríguez-Molinero. Home detection of freezing of gait using support vector machines through a single waist-worn triaxial accelerometer. *PLOS ONE*, 12(2):1–26, 02 2017.

[12] Wei Zhang, Zhuokun Yang, Hantao Li, Debin Huang, Lipeng Wang, Yanzhao Wei, Lei Zhang, Lin Ma, Huanhuan Feng, Jing Pan, Yuzhu Guo, and Piu Chan. Multimodal data for the detection of freezing of gait in parkinson's disease. *Scientific Data*, 9, 10 2022.

[13] Natasa K. Orphanidou, Abir Hussain, Robert Keight, Paulo Lishoa, Jade Hind, and Haya Al-Askar. Predicting freezing of gait in parkinsons disease patients using machine learning. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2018.

[14] H. Gokul, Prithvi Suresh, B. Hari Vignesh, R. Pravin Kumaar, and Vineeth Vijayaraghavan. Gait recovery system for parkinson's disease using machine learning on embedded platforms. In *2020 IEEE International Systems Conference (SysCon)*, pages 1–8, 2020.

[15] Thoma Bikias, Dimitrios Lakovakis, Stelios Hadjidimitriu, Vasileios Charisis, and Lentios J. Hadjileontiadis. Deepfog: An imu-based detection of freezing of gait episodes in parkinson's disease patients via deep learning. *Frontiers*, 05 2021.

[16] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, oct 2001.

[17] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[18] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. *International Conference on Engineering and Technology*, pages 1–6, 2017.

[19] Angel Peinado-Contreras and Mario Munoz-Organero. Gait-based identification using deep recurrent neural networks and acceleration patterns. *Sensors*, pages 1–26, 11 2020.

[20] Chirag Gupta, Arun Sai Suggala, Ankit Goyal, Harsha Vardhan Simhadri, Bhargavi Paranjape, Ashish Kumar, Saurabh Goyal, Raghavendra Udupa, Manik Varma, and Prateek Jain. ProtoNN: Compressed and accurate kNN for resource-scarce devices. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1331–1340. PMLR, 06–11 Aug 2017.

[21] David A.E. Bolton and James K. Richardson. Inhibitory control and fall prevention: Why stopping matters. *Frontiers*, 03 2020.

[22] Johanna ODay, Marissa Lee, Kirsten Seagers, Shannon Hoffman, Ava Jih-Schiff, Lukasz Kidzinski, Scott Delp, and Helen Bronte-Stewart. Assessing inertial measurement unit locations for freezing of gait detection and patient preference. *PubMed*, 02 2022.

[23] Scott Pardoel, Gaurav Shalin, Julie Nantel, Edward D Lemaire, and Jonathan Kofman. Early detection of freezing of gait during walking using inertial measurement unit and plantar pressure distribution data. *Sensors*, 02 2021.

[24] Luis Sigcha, Luigi Borzì, Ignacio Pavón, Nélson Costa, Susana Costa, Pedro Arezes, Juan Manuel López, and Guillermo De Arcas. Improvement of performance in freezing of gait detection in parkinson's disease using transformer networks and a single waist-worn triaxial accelerometer. *Engineering Applications of Artificial Intelligence*, 116:105482, 2022.