



# HHS Public Access

Author manuscript

*IEEE Int Conf Connect Health Appl Syst Eng Technol.* Author manuscript; available in PMC 2023 September 22.

Published in final edited form as:

*IEEE Int Conf Connect Health Appl Syst Eng Technol.* 2023 June ; 2023: 1–10.

## Parkinson's Disease Action Tremor Detection with Supervised-Leaning Models

**Minglong Sun,**

Computer Science Department, William & Mary, Williamsburg, United States

**Woosub Jung,**

Computer Science Department, William & Mary, Williamsburg, United States

**Kenneth Koltermann,**

Computer Science Department, William & Mary, Williamsburg, United States

**Gang Zhou,**

Computer Science Department, William & Mary, Williamsburg, United States

**Amanda Watson,**

The PRECISE Center, University of Pennsylvania, Philadelphia, United States

**Ginamari Blackwell,**

School of Nursing, Virginia Commonwealth University, Richmond, United States

**Noah Helm,**

School of Nursing, Virginia Commonwealth University, Richmond, United States

**Leslie Cloud,**

Department of Neurology, Virginia Commonwealth University, Richmond, United States

**Ingrid Pretzer-Aboff**

School of Nursing, Virginia Commonwealth University, Richmond, United States

### Abstract

People with Parkinson's Disease (PD) have multiple symptoms, such as freezing of gait (FoG), hand tremors, speech difficulties, and balance issues, in different stages of the disease. Among these symptoms, hand tremors are present across all stages of the disease. PD hand tremors have critical consequences and negatively impact the quality of PD patients' everyday lives. Researchers have proposed a variety of wearable devices to mitigate PD tremors. However, these devices require accurate tremor detection technology to work effectively while the tremor occurs.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

[msun05@email.wm.edu](mailto:msun05@email.wm.edu) .

#### CONFLICT OF INTEREST

Dr. Ingrid Pretzer-Aboff has a financial interest in Resonate Forward, LLC, a company with a commercial interest in vibration technology to treat PD tremors. This conflict has been reviewed and managed by the Virginia Commonwealth University office of sponsored programs.

This paper introduces a PD action tremor detection method to recognize PD tremors from regular activities. We used a dataset from 30 PD patients wearing accelerometers and gyroscope sensors on their wrists. We selected time-domain and frequency-domain hand-crafted features. Also, we compared our hand-crafted features with existing CNN data-driven features, and our features have more specific boundaries in 2-D feature visualization using the t-SNE tool. We fed our features into multiple supervised machine learning models, including Logistic Regression (LR), K-Nearest Neighbours (KNNs), Support Vector Machines (SVMs), and Convolutional Neural Networks (CNNs), for detecting PD action tremors. These models were evaluated with 30 PD patients' data. The performance of all models using our features has more than 90% of F1 scores in five-fold cross-validations and 88% F1 scores in the leave-one-out evaluation. Specifically, Support Vector Machines (SVMs) perform the best in five-fold cross-validation with over 92% F1 scores. SVMs also show the best performance in the leave-one-out evaluation with over 90% F1 scores.

## Keywords

Parkinson's Disease; Action Tremor Detection; Wearable Device; Supervised-learning

---

## 1 INTRODUCTION

Parkinson's disease (PD) is a chronic and continuous central nervous degeneration disease with more than ten million patients suffering worldwide[13]. PD has these main motor symptoms, including bradykinesia, postural instability, rigidity, freezing of gait, and tremors[29]. Typical PD tremors are hand and finger tremors, which can cause inconvenience and potentially severe consequences in patients' daily lives[2].

Currently, PD and PD tremors cannot be cured entirely. However, multiple treatments and mitigation methods are performed to decrease tremor severity. Traditional clinical approaches have been utilized to mitigate the symptoms, including deep brain stimulation surgery[3] and Levodopa replacement therapy [9]. However, these are not always successful and have side effects that can be more disruptive than the symptom itself. In recent years, less intrusive wearable devices like vibration devices have been utilized to mitigate tremors in the home environment[11, 38].

In order to design a system to mitigate or quantify PD tremors, we need to detect the PD tremors accurately. Typically, PD tremors have three types: rest, postural, and action tremors[17]. The patients' activities help distinguish the tremor types. Rest tremors are seen when patients sit or lie down without doing any activities. Postural tremors occur when patients maintain a stationary pose. Action tremor appears as patients are involved in activities such as writing and drawing. TremorSense[36], a system to detect PD tremors, has used a multi-layer Convolutional Neural Network (CNN) method. The CNN method detects rest tremors and postural tremors with above 95% accuracy. However, the accuracy and F1-score for action tremor detection was only 70%–80% in our evaluation. This problem motivates us to investigate more reliable features and robust models to classify PD action tremors.

To accurately detect PD action tremors, we need to solve the two following research questions:

RQ1: How to design features to accurately detect action tremors?

RQ2: How to design models to accurately detect action tremors?

In this paper, we introduced an approach to classify PD action tremors from daily activities. We used the dataset collected from 30 PD patients with a wearable device consisting of accelerometer and gyroscope sensors on the patient's wrists. We divide the data into instances with a time frame of 1.28 seconds. The sampling rate of the sensors was 100Hz. Each instance has  $1.28 \times 100 \times 6$  data points. We employed 4,000 action tremor instances and 4,000 activity instances. We extracted hand-crafted features based on domain knowledge and selected the features with the LASSO algorithm. Based on the t-SNE visualization tool, the high dimension distribution of the our hand-crafted features is more apparent than the data-driven TremorSense features. We evaluated our features using Linear Regression (LR), Support Vector Machines (SVMs), K-Nearest Neighbors (KNNs), and Convolutional Neural Network (CNN). The performance of all four models has more than 90% of F-1 scores and 92% accuracies.

Our contributions are summarized as follows:

- To answer research question 1, we extracted hand-crafted features based on domain knowledge and selected the features with the LASSO algorithm. From the t-SNE visualization results, our hand-crafted features have more apparent boundaries than the data-driven TremorSense features.
- To answer research question 2, we evaluated our features on four supervised learning models. The performance of all models has more than 90% of F-1 scores and 92% of accuracies, which indicates that different classification algorithms perform robustly and similarly with our features.

We organize our paper as follows. In Section 2, we first introduce the details of our dataset. Then, we present feature extraction and selection methods. Section 3 presents the machine-learning models we utilized to classify PD actions tremors. In Section 4, we introduce the classification performance of these machine-learning models. In Section 5, we discuss our limitations and propose potential works. In Section 6, we present the related works about tremor detection. In Section 7, we conclude our research results.

## 2 FEATURE EXTRACTION AND SELECTION

### 2.1 Dataset

We used the same dataset as TremorSense [36]. The dataset was collected in a clinical environment with the help of medical specialists. The dataset includes 30 patients performing seven UPDRS activities and three Fahn-Tolosa-Marin scale activities. Table 1 shows the specific activities, and Table 2 describes all the 30 patients' demographic information.

The data for the TremorSense dataset was collected using two UG sensor bands containing a three-axis accelerometer and a three-axis gyroscope [43]. The UG sensor band is shown in Figure 1. The sensor data was transmitted via Bluetooth to the TremorSense App. During the tests with patients, medical professionals used a camera to record patients upper extremity movement to serve as the ground truth data. The ground truth videos were then used to label the action tremor output recorded by the UG sensor band.

The UG sensor sampling rate is 100Hz in the TremorSense dataset. The dataset was divided into windows with a 1.28 seconds time windows and a 0.64-second step size. Since the UG sensor has a three-axis accelerometer and a three-axis gyroscope, each window includes 128\*6 data points. We selected 4,000 action tremor instances and 4,000 non-tremor activities from the dataset. The total data points are 6,144,000 (128\*6\*4000\*2), enough to train the TremorSense CNN classifier [36].

## 2.2 Features Extraction and Selection

As we evaluated the TremorSense data-driven features in Section 4, the accuracy is only about 80%. This result motivated us to select more hand-crafted features. Therefore, we extracted 6 \* 14 features from both the time and frequency domains. Table 3 displays the extracted features. The time-domain features include Min, Max, Mean, Standard Deviation, and Zero crossing values. These features are computed for each raw component of the accelerometer and gyroscope readings. Similarly, the frequency-domain features are Energy, Entropy, Spectral-centroid, and the first three dominant frequencies and their magnitudes between 2Hz-10Hz. After we extract these features, we use feature selection techniques to select the best features to predict action tremors.

Figure 2 shows the three axes directions of both sensors in the UG sensor. The white arrows indicate the direction of motion of PD action tremors. Intuitively, the PD action tremors have very little acceleration in the Y direction along the arms. Conversely, PD tremors have reasonable acceleration variation in both the X and the Z directions. This observation leads us to select more acceleration features in the X and Z direction. Feature selection is also beneficial for reducing high-dimension classification costs and helps avoid overfitting. The feature selection algorithm we have chosen in a Logistic Regression method is Least Absolute Shrinkage and Selection Operator (LASSO).

LASSO is a popular regularization and feature selection algorithm for machine learning classifiers. The LASSO model has a constraint on the sum of the absolute values of the model parameters, and the sum has to be less than an upper bound. The method applies a shrinking process, also known as the regularization process, where it penalizes the weights of the regression features and forces some of them to zero. If the features still have a non-zero weight after the shrinking process, they are selected to be a part of the model.

The LASSO starts with a linear regression model and can be expressed as :

$$Y_i = \beta_0 + X_i\beta_1 + \dots + X_n\beta_n \quad (1)$$

$X_j$  is the input features,  $Y$  is the prediction matrix, and the parameters  $\beta_0, \beta_1, \dots, \beta_n$  are the regression weights. The cost function of LASSO regression is as follows:

$$\sum_{i=1}^n \left( y_i - \sum_{j=1}^n x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^n \beta_j^2 \quad (2)$$

$\lambda \sum_{j=1}^n \beta_j^2$  is the LASSO penalty addition to the L1 optimization cost function. If  $\lambda$  is zero, then it will be a common L1 optimization problem. However, as  $\lambda$  becomes larger, LASSO will shrink the weight of less significant features to zero, which works perfectly for feature selection. During our feature selection implementation, we select the  $\lambda$  that perfectly drops the Y-axis of acceleration features and maintains other useful axes features.

Table 4 shows the selected features. As expected, the Y-axis features of the accelerometer are dropped. Also, the maximum and minimum value time-domain features are dropped for all axes. Those features are dropped because the maximum and minimum values are typically not predictable when the patients perform activities, and they are not related to the action tremors. In total, the selected features are  $5 * 3$  time-domain features and  $5 * 9$  frequency-domain features. Table 5 shows the top 20 selected features and their importance scores. The importance scores are calculated as follows:

$$\text{Importance score} = \frac{\text{Certain Feature Weight}}{\sum | \text{Weight} |} \quad (3)$$

The Top 20 features total weights are 54.96% of the total weights, representing the most important features. Next, we feed all the selected features into the supervised-learning models.

### 3 MODELING

We implemented multiple supervised learning classifiers with the selected to detect PD action tremors. We employed classic learning algorithms, such as Linear Classification Models, K-Nearest-Neighbors (KNNs), and Support-Vector Machines (SVMs). We also applied Convolutional Neural Networks (CNNs) to perform the classification. In the following section, we present each classifier we designed to differentiate PD action tremors from regular activities.

#### 3.1 Linear Classification Model

The first model we employ is a Linear Classification Model. The model  $Y$  can be presented as follows:

$$Y = \beta_0 + \sum_{j=0}^p \beta_j X_j \quad (4)$$

where  $X$  is the feature matrix, and  $\beta$  is the weight for each predictor.

$$\hat{\beta} = \operatorname{argmin}_{\beta} L(\beta) \quad (5)$$

where  $\hat{\beta}$  is the weight vector that can minimize the loss function  $L(\beta)$ . The loss function can be presented as follow:

$$L(\beta) = \|y - \beta X\|^2 \quad (6)$$

The final hyperplane that separates the two classes is:

$$\hat{Y} = X\hat{\beta} = X(X^T X)^{-1} X^T Y \quad (7)$$

$\hat{Y}$  is the projection of  $Y$  onto columns of  $X$ . The hyperplane separates the two classes of space. If a data point lies in a certain space, then the instance is classified as that category.

### 3.2 KNN Classification Model

The k-nearest-neighbors classification model is another frequently used supervised-learning model. A majority decision of the nearby data points can classify an input instance. According to a Euclidean distance function, an instance is categorized into the class with the highest prevalence among its  $K$  closest neighbors:

$$D = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (8)$$

Intuitively, if a particular instance is near most of the tremor class points, it is classified as a tremor instance. The KNN model has a principal parameter  $K$ ; as  $K$  increases, the model becomes more inflexible. We implement five-fold cross-validation to choose  $K$  with the highest accuracy to pick the  $K$ .

### 3.3 SVM Classification Model

Another well-known supervised-learning algorithm we employ is Support-Vector Machine. It is most suitable for binary classification, and we classify tremor and non-tremor instances in our model. The advantage of SVM compared with the previous algorithms is that SVM provides linear classification and non-linear classification. It can provide a more flexible model that is moderately helpful in classifying similar instances on the boundary between tremor and non-tremor instances.

The SVM maximizes the margin around the separating hyper-plane. A subset of training samples specifies the decision function called the support vectors. The equation that defines the decision surface separating the classes is a hyperplane of the following form:

$$W^T + B = 0$$

$W$  is a weight vector,  $X$  is the input vector, and  $B$  is the bias. In our case, the hyperplane separates tremor and non-tremor classes.

### 3.4 CNN Classification Model Design

In this subsection, we present our CNN design for action tremor classification. We classify each input action instance as a tremor event or a regular activity. Each instance includes time-domain and frequency-domain signal features. Our CNN model consists of eight layers. Next, we present each layer and its components of our CNN model in the following paragraphs.

**3.4.1 Input Layer.**—This input layer takes our 1-D (1 \* 60) instance into the CNN model. It primarily organizes the input size for the subsequent layers, and it has any parameters or features to learn.

**3.4.2 Convolutional Layer.**—Our CNN model has one convolutional layer since our input size is small and simple to learn. The input size of our model is 1 \* 60. We select a kernel size of 1 \* 3 and the number of kernels is eight. The stride direction is only horizontal, and we use a stride size of 1 \* 1. The convolutional layer generates the same amount of feature maps for each input instance as kernels. Therefore, the outputs of our convolutional layer are 1 \* 58 \* 8 feature maps.

**3.4.3 Batch Normalization Layer.**—To decrease the sensitivity to network initialization and prevent overfitting, we employ a batch normalization layer. Additionally, it can speed up CNN training and apply feature maps to the next ReLU layer. Eq. 10 illustrates the activation function. This function standardizes the input  $s_i$  by a mini-batch for each input instance. Eq. 11 demonstrates the output of this layer.

$$\hat{s}_i = \frac{(s_i - \mu_B)}{(\sqrt{\sigma_B^2 + \epsilon})} \quad (10)$$

$\mu_B$  is the mean and  $\sigma_B$  is the variance of the mini-batch.

$$y_i = k\hat{s}_i + \beta \quad (11)$$

where  $k$  is the scale factor,  $\beta$  is the compensation, and  $\hat{s}_i$  is the normalized output in Eq. 10.

**3.4.4 Relu Layer.**—The ReLU layer, also known as the rectified linear unit layer, is a function that uses a non-linear activation function to transform all the negative input values into zero. Eq. 12 shows the activation function:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (12)$$

The ReLU layer is more computationally efficient and helps avoid overfitting.

**3.4.5 Max Pooling Layer.**—The max pooling layer is primarily used to downsample the ReLU Layer's output. The output of the max pooling layer is the max value of the inputs by a pooling filter. The max pooling layer utilized a pooling filter to generate the output by selecting the max value of the input matrix. In our model, the pool size is 1 \* 2, and the stride size is 1 \* 2. The output of our Max Pooling layer is 1 \* 29 \* 8.

**3.4.6 Fully-connected Layer.**—This fully-connected layer is the neural network to calculate the weights and bias based on the previous input features. Each neuron decides the weight that prioritizes the most relevant category and generates the probability for the classification decision. We use 32 neurons in our Fully-connected layer and have 7,456 parameters to train.

**3.4.7 Softmax Layer.**—This softmax layer uses the softmax function to generate the probability for each instance that belongs to ‘*action tremor*’ and ‘*no action tremor*’. In this softmax layer, we set our classification probability boundary as 50%, which means if the probability of an instance is more than 50% of the ‘*action tremor*’, then it will be classified as a tremor instance.

**3.4.8 Classification Output Layer.**—The output of our classification layer is a binary result of ‘*action tremor*’ and ‘*no action tremor*’. For each round of training and testing in the evaluation phase, the outcome is compared against the ground truth to create the confusion matrix.

## 4 PERFORMANCE EVALUATION

We implemented two evaluation experiments, cross-evaluation and leave-one-out evaluation, to illustrate the performance of our supervised learning classifiers and selected features. We then compared our models and features with the-state-of-art model TremorSense. We utilized the 30 patients’ dataset mentioned in the previous section. The dataset included 4,000 tremor class instances and 4,000 non-tremor class instances. Since the action tremor always happens unpredictably, some patients had more action tremors than others. This issue had less influence on cross-evaluation. This issue had less influence on cross-evaluation. However, it will have an impact on the leave-one-out evaluation. Therefore, we picked the top 10 PD patients’ data that included more action tremor events, to test our model in the leave-one-out evaluation. Next, we present the evaluation results and the feature visualization in detail.

### 4.1 Cross-Evaluation

We trained and tested our classifiers using all 30 patients’ instances in cross-validation. We employed five-fold cross-validation methods to train and test for each round. We also evaluated the results with the TremorSense CNN classifier. Figure 4 shows the results of each classification confusion matrix results. Table 6 shows the overall accuracy, precision, recall, and F1-score for each classifier. The F1-scores for all four models are higher than 90%. SVM performs best with more than 92% of accuracy and F1-score. All four models fed with our hand-crafted features perform better than the TremorSense model fed with data-driven CNN features. The results demonstrate that our hand-crafted features are robust with different machine learning models.

### 4.2 Leave-one-out Evaluation

As we mentioned above, we selected ten patients for leave-one-out evaluation. In this experiment, we trained the classifiers with the nine patients’ data and tested the classifiers

with the one remaining patient's data. The total tremor instances are 2,000, and the total non-tremor instances are 2,000. Table 7 shows the leave-one-out results for each classifier. Even though the leave-one-out results are slightly less than those of cross-evaluation, the performance is still robust. We can also conclude that SVM has the highest performance with all confusion matrix parameters greater than 91%. Similar to the results of cross-evaluation, all four models perform better than the TremorSense model. The results demonstrate that our hand-crafted features and models are robust with new patients.

### 4.3 Features Visualization

We further demonstrate that our hand-crafted features perform better than the TremorSense data-driven features. We employed a tool, t-distributed stochastic neighbor embedding (t-SNE), to visualize the high-dimensional features. The t-SNE is a statistical technique to visualize high-dimensional data by assigning every data point a particular location in a 2-D or 3-D figure. Figure 6 shows the distribution of our hand-crafted features and TremorSense data-driven features in 2-D maps. The top two figures show the TremorSense features, and the bottom two figures show our hand-crafted features. We randomly selected 500 instances to plot in these figures. The red dots are the tremor instances, and the blue dots are the non-tremor instances. The X-axis comp-1 and The Y-axis comp-2 represent the two composite features from high dimensions. The different random states represent the random different projection angles to display the features in the 2-D dimension. We can conclude from the figures that our hand-crafted features have more specific boundaries than the TremorSense data-driven features. The results also demonstrate that our features perform better than TremorSense features with the supervised machine learning models.

## 5 RELATED WORKS

In recent years, many computer scientists have focused on solving PD-related problems with the assistance of a specialist in the field. Researchers utilize different sensors and algorithms to develop detection and quantification systems of various PD symptoms, including freezing of gait (FOG)[21, 37], hand tremors, speech difficulties[4, 24], etc. There have been increasing research works focusing on hand tremor detection and severity assessment in the past decade. They employ different sensors, such as accelerometers[16, 18–20, 22, 31, 32], gyroscopes[34, 35], electromyography (EMG)[5,23], WiFi-based sensors[6], and other motion sensors[26–28, 30, 40] to build detection devices. Based on various signals from these sensors, different models, such as threshold models[16, 18–20, 22, 31], machine-learning models[10, 41, 42], and deep learning models[6, 36, 39], are developed to detect hand tremor events.

Researchers have explored new methods to detect PD hand tremors in the last three years. In 2021, TremorSense[36] proposed a CNN model to detect three types of tremors using the 3-D time-domain accelerometer and gyroscope data. The dataset they collected includes 11 UPDRS activities and three Fahn-Tolosa-Marin activities. In our paper, we employed the same dataset and used the action tremor activities and normal activities as a comparison group. The accuracy of detecting all types of tremors in the TremorSense was more than 94%. When we applied the TremorSense CNN model to the data of only action tremors with

everyday activities, the evaluation results were 70% to 80%. However, we beat TremorSense using our hand-crafted features, and our classification accuracy and F-1 score were more than 90%. In 2021, Tong et al. [39] used their designed IMU sensors to detect PD hand tremors. They used a CNN model to classify tremor events and got an F-1 score of 94.31%. Even though the F-1 score was high, they only collected data from five PD patients who walked in a circle back and forth. In comparison, our dataset is more diverse and results are more reliable since we collected data from 30 PD patients with 14 clinically relevant activities. In 2021[6], Chen et al. employed Wi-Fi sensing for hand rest tremor detection. In a small room, they simulated the frequency of hand tremors from 3Hz to 7Hz and collected channel state information (CSI) signals. They used pre-trained models VGG19 and Resnet152 from Keras and customized the last multiple layers for tremor classification. In 2021, Rini et al.[33] developed a glove with five accelerometers on each finger. They collected data from six patients and calculated the acceleration frequency of each finger. If the frequencies from all five fingers were between 1Hz to 10Hz, it would be a tremor event. This detection algorithm does not consider when some fingers have 0Hz frequencies while the frequencies others are in the range of 1Hz to 10Hz. In 2020, Ibrahim et al. [12] fed the acceleration and angular velocity data into a hybrid convolutional-multilayer perceptron neural network to detect tremor events. The detection is limited to user-independent and task-independent activities, while our methods can work among different users and activities. In 2020 [1], Ahmed et al. utilized Photoplethysmograph (PPG) built-in smartwatch and EMG sensor to detect hand tremors. The highest accuracy of the tremor detection algorithm was 89%, and they collected data only from healthy subjects.

Researchers are using hand tremor signals to do more. They detect hand tremor signals to help assist wearable devices such as vibration devices to mitigate tremor symptoms in PD patients. They also use the signals to assess tremor severity[14, 15, 32], by applying similar sensors and machine learning algorithms to quantify tremor severity based on clinical standards such as UPDRS scores. In addition, some researchers employ hand tremor signals to detect PD[25] and quantify the severity of PD[8]. All these applications show the value and necessity of hand tremor signal detections.

## 6 DISCUSSION AND FUTURE WORK

Our paper used conventional motion sensors like accelerometers and gyroscopes to collect data. These sensors are practical and portable for PD patients to use in any environment. However, it is difficult for motion sensors to forecast tremor events. Motion sensors can detect tremor signals when a tremor has already happened. Future research can look into how to predict hand tremors and other PD symptoms using clinical sensors like EEG and EMG.

Our project collected the data in the clinical environment. We did the data processing and built tremor classification models offline. The future work for us is to develop an online system that can detect hand tremors in real-time with wearable devices and smartphones, and run a pilot study to test the robustness and accuracy of our system in different environments.

Our paper mainly selected supervised machine-learning models since we have labeled the action tremor events from tremor activities based on the videos. Future research can focus on unsupervised learning and deep learning models. Those models may potentially enhance the action tremor detection performance.

As we mentioned in the related work section, PD hand tremor signals may be used to evaluate the tremor severity and PD severity. Wearable technology can accurately detect tremors using our methods. PD patients may employ non-intrusive wearable technology to track the daily change in severity in a domestic environment setting. Additionally, accurate tremor detection helps wearable technology provide the appropriate care and mitigation. Our tremor detection techniques will be used in future studies to create these wearable devices.

## 7 CONCLUSION

Our paper focuses on detecting PD action tremors using the accelerometer and gyroscope sensor. We extracted hand-crafted features based on domain knowledge and selected the features with the LASSO algorithm. From the t-SNE visualization results, our hand-crafted features have more apparent boundaries than the state-of-art data-driven TremorSense features. We evaluated our features on four supervised learning models. The performance of all models has more than 90% of F-1 scores cross-validation results and more than 88% of F-1 scores leave-one-out results, which indicates that different classification algorithms perform robustly and similarly with our features. The performance of our models using hand-crafted features beat the TremorSense using data-driven features.

## ACKNOWLEDGMENT

This research was partially supported by the U.S. NIH NINDS under grant R01NS120560 and by the U.S. NSF under grant CNS-1841129 (CSR EAGER). The author would like to thank all co-authors who have contributed to collect the PD patients' data.

## REFERENCES

- [1]. Ahmed Nasimuddin, Bhattacharyya Chirayata, and Ghose Avik. 2021. A Novel Non-Parametric Approach Of Tremor Detection Using Wrist-Based Photoplethysmograph. In 2020 28th European Signal Processing Conference (EUSIPCO). IEEE, 1150–1154.
- [2]. Anouti Ahmad and Koller William C. 1995. Tremor disorders. Diagnosis and management. *Western journal of medicine* 162, 6 (1995), 510. [PubMed: 7618310]
- [3]. Benabid Alim Louis. 2003. Deep brain stimulation for Parkinson's disease. *Current opinion in neurobiology* 13, 6 (2003), 696–706. [PubMed: 14662371]
- [4]. Braga Diogo, Madureira Ana M, Coelho Luis, and Ajith Reuel. 2019. Automatic detection of Parkinson's disease based on acoustic analysis of speech. *Engineering Applications of Artificial Intelligence* 77 (2019), 148–158.
- [5]. Breit S, Spieker S, Schulz JB, and Gasser T. 2008. Long-term EMG recordings differentiate between parkinsonian and essential tremor. *Journal of neurology* 255, 1 (2008), 103–111. [PubMed: 18204805]
- [6]. Chen Shih-Yuan and Lin Chi-Lun. 2022. Subtle Motion Detection Using Wi-Fi for Hand Rest Tremor in Parkinson's Disease. In 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 1774–1777.
- [7]. Cummings Jeffrey, Isaacson Stuart, Mills Roger, Williams Hilde, Chi-Burris Kathy, Corbett Anne, Dhall Rohit, and Ballard Clive. 2014. Pimavanserin for patients with Parkinson's disease

psychosis: a randomised, placebo-controlled phase 3 trial. *The Lancet* 383, 9916 (2014), 533–540.

- [8]. Dai Houde, Zhang Pengyue, and Lueth Tim C. 2015. Quantitative assessment of parkinsonian tremor based on an inertial measurement unit. *Sensors* 15, 10 (2015), 25055–25071. [PubMed: 26426020]
- [9]. Dodd M Leann, Klos Kevin J, Bower James H, Geda Yonas E, Josephs Keith A, and Ahlskog J Eric. 2005. Pathological gambling caused by drugs used to treat Parkinson disease. *Archives of neurology* 62, 9 (2005), 1377–1381. [PubMed: 16009751]
- [10]. Fraiwan Luay, Khnouf Ruba, and Mashagbeh Abdel Razaq. 2016. Parkinson’s disease hand tremor detection system for mobile application. *Journal of medical engineering & technology* 40, 3 (2016), 127–134. [PubMed: 26977823]
- [11]. Haas Christian T, Turbanski Stephan, Kessler Kirn, and Schmidtbleicher Dietmar. 2006. The effects of random whole-body-vibration on motor symptoms in Parkinson’s disease. *NeuroRehabilitation* 21, 1 (2006), 29–36. [PubMed: 16720935]
- [12]. Ibrahim Anas, Zhou Yue, Jenkins Mary E, Trejos Ana Luisa, and Naish Michael D. 2020. The design of a Parkinson’s tremor predictor and estimator using a hybrid convolutional-multilayer perceptron neural network. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 5996–6000.
- [13]. Javor Andrija, Ransmayr Gerhard, Struhel Walter, and Riedl René. 2016. Parkinson patients’ initial trust in avatars: theory and evidence. *PloS one* 11, 11 (2016), e0165998. [PubMed: 27820864]
- [14]. Jeon Hyoseon, Lee Woongwoo, Park Hyeyoung, Lee Hong Ji, Kim Sang Kyong, Kim Han Byul, Jeon Beomseok, and Park Kwang Suk. 2017. Automatic classification of tremor severity in Parkinson’s disease using a wearable device. *Sensors* 17, 9 (2017), 2067. [PubMed: 28891942]
- [15]. Jeon Hyoseon, Lee Woongwoo, Park Hyeyoung, Lee Hong Ji, Kim Sang Kyong, Kim Han Byul, Jeon Beomseok, and Park Kwang Suk. 2017. High-accuracy automatic classification of Parkinsonian tremor severity using machine learning method. *Physiological measurement* 38, 11 (2017), 1980. [PubMed: 28933707]
- [16]. Kostikis N, Hristu-Varsakelis Dimitrios, Arnaoutoglou M, and Kotsavasiloglou C. 2014. Smartphone-based evaluation of parkinsonian hand tremor: Quantitative measurements vs clinical assessment scores. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 906–909.
- [17]. Krack Paul, Benazzouz Abdelhamid, Pollak Pierre, Limousin Patricia, Piallat Brigitte, Hoffmann Dominique, Xie Jing, and Benabid Alim-Louis. 1998. Treatment of tremor in Parkinson’s disease by subthalamic nucleus stimulation. *Movement disorders: official journal of the Movement Disorder Society* 13, 6 (1998), 907–914. [PubMed: 9827614]
- [18]. LeMoyne Robert, Coroian Cristian, and Mastroianni Timothy. 2009. Quantification of Parkinson’s disease characteristics using wireless accelerometers. In *2009 ICME International Conference on Complex Medical Engineering*. IEEE, 1–5.
- [19]. LeMoyne Robert, Mastroianni Timothy, Cozza Michael, Coroian Cristian, and Grundfest Warren. 2010. Implementation of an iPhone for characterizing Parkinson’s disease tremor through a wireless accelerometer application. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, 4954–4958.
- [20]. LeMoyne Robert, Mastroianni Timothy, and Grundfest Warren. 2013. Wireless accelerometer configuration for monitoring Parkinson’s disease hand tremor. *Advances in Parkinson’s Disease* 2, 02 (2013), 62.
- [21]. Mazilu Sinziana, Hardegger Michael, Zhu Zack, Roggen Daniel, Tröster Gerhard, Plotnik Meir, and Hausdorff Jeffrey M. 2012. 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*. IEEE, 123–130.
- [22]. Yu Meigal A, Rissanen SM, Tarvainen MP, Georgiadis SD, Karjalainen PA, Airaksinen O, and Kankaanpää M. 2012. Linear and nonlinear tremor acceleration characteristics in patients with Parkinson’s disease. *Physiological measurement* 33, 3 (2012), 395. [PubMed: 22370008]

- [23]. Milanov Ivan. 2001. Electromyographic differentiation of tremors. *Clinical Neurophysiology* 112, 9 (2001), 1626–1632. [PubMed: 11514245]
- [24]. Moro-Velazquez Laureano, Gomez-Garcia Jorge A, Arias-Londoño Julian D, Dehak Najim, and Godino-Llorente Juan I. 2021. Advances in Parkinson’s disease detection and assessment using voice and speech: A review of the articulatory and phonatory aspects. *Biomedical Signal Processing and Control* 66 (2021), 102418.
- [25]. Nalini M, Gayathiri R, Srimathi R, Vidyathmikaa R, and Jenifer S. 2022. Detection of Parkinson’s Disease Using Voice Changes and Hand-tremor. In *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*. IEEE, 1–4.
- [26]. Niazmand Khalil, Tonn Karin, Kalaras Anastasios, Fietzek Urban M, Mehrkens Jan-Hinnerk, and Lueth Tim C. 2011. Quantitative evaluation of Parkinson’s disease using sensor based smart glove. In *2011 24th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 1–8.
- [27]. Okuno Ryuhei, Yokoe Masaru, Akazawa Kenzo, Abe Kazuo, and Sakoda Saburo. 2006. Finger taps movement acceleration measurement system for quantitative diagnosis of Parkinson’s disease. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 6623–6626.
- [28]. O’Suilleabhain Padraig E and Dewey Richard B Jr. 2001. Validation for tremor quantification of an electromagnetic tracking device. *Movement disorders: official journal of the Movement Disorder Society* 16, 2 (2001), 265–271. [PubMed: 11295779]
- [29]. Politis Marios, Wu Kit, Molloy Sophie, Bain Peter G., Chaudhuri K Ray, and Piccini Paola. 2010. Parkinson’s disease symptoms: the patient’s perspective. *Movement Disorders* 25, 11 (2010), 1646–1651. [PubMed: 20629164]
- [30]. Rajaraman V, Jack D, Adamovich SV, Hening W, Sage Jacob, and Poizner H. 2000. A novel quantitative method for 3D measurement of Parkinsonian tremor. *Clinical neurophysiology* 111, 2 (2000), 338–343. [PubMed: 10680570]
- [31]. Rigas George, Tzallas Alexandros T, Tsalikakis Dimitrios G, Konitsiotis Spiros, and Fotiadis Dimitrios I. 2009. Real-time quantification of resting tremor in the Parkinson’s disease. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 1306–1309.
- [32]. Rigas George, Tzallas Alexandros T, Tsipouras Markos G, Bougia Panagiota, Tripoliti Evanthia E, Baga Dina, Fotiadis Dimitrios I, Tsouli Sofia G, and Konitsiotis Spyridon. 2012. Assessment of tremor activity in the Parkinson’s disease using a set of wearable sensors. *IEEE Transactions on Information Technology in Biomedicine* 16, 3 (2012), 478–487. [PubMed: 22231198]
- [33]. Rini Eka Mistiko and Haq Endi Sailul. 2021. Detection Hand Tremor Through Each Finger Movement Based On Arduino For Parkinson’s Patients. In *2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*. IEEE, 225–230.
- [34]. Salarian Arash, Russmann Heike, Vingerhoets FJG, Burkhard PR, Blanc Yves, Dehollain Catherine, and Aminian Kamiar. 2003. An ambulatory system to quantify bradykinesia and tremor in Parkinson’s disease. In *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, 2003*. IEEE, 35–38.
- [35]. Salarian Arash, Russmann Heike, Wider Christian, Burkhard Pierre R, Vingerhoets Franios JG, and Aminian Kamiar. 2007. Quantification of tremor and bradykinesia in Parkinson’s disease using a novel ambulatory monitoring system. *IEEE Transactions on Biomedical Engineering* 54, 2 (2007), 313–322. [PubMed: 17278588]
- [36]. Sun Minglong, Watson Amanda, Blackwell Gina, Jung Woosub, Wang Shuangquan, Koltermann Kenneth, Helm Noah, Zhou Gang, Cloud Leslie, and Pretzer-Abhoff Ingrid. 2021. TremorSense: Tremor Detection for Parkinson’s Disease Using Convolutional Neural Network. In *2021 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 1–10.
- [37]. Sun Minglong, Watson Amanda, and Zhou Gang. 2020. Wearable computing of Freezing of Gait in Parkinson’s disease: A survey. *Smart Health* 18 (2020), 100143.
- [38]. Swallow Lee and Siores Elias. 2009. Tremor suppression using smart textile fibre systems. *Journal of Fiber Bioengineering and Informatics* 1, 4 (2009), 261–266.

- [39]. Tong Lina, He Jiayi, and Peng Liang. 2021. CNN-based PD hand tremor detection using inertial sensors. *IEEE Sensors Letters* 5, 7 (2021), 1–4. [PubMed: 36789370]
- [40]. Yokoe M, Okuno R, Hamasaki T, Kurachi Y, Akazawa K, and Sakoda S. 2009. Opening velocity, a novel parameter, for finger tapping test in patients with Parkinson’s disease. *Parkinsonism & related disorders* 15, 6 (2009), 440–444. [PubMed: 19103505]
- [41]. Zhang Ada, Cebulla Alexander, Panev Stanislav, Hodgins Jessica, and De la Torre Fernando. 2017. Weakly-supervised learning for Parkinson’s Disease tremor detection. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 143–147.
- [42]. Zhang Ada, San-Segundo Rubén, Panev Stanislav, Tabor Griffin, Stebbins Katelyn, Whitford Andrew, De la Torre Fernando, and Hodgins Jessica. 2018. Automated Tremor Detection in Parkinson’s Disease Using Accelerometer Signals. In *2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 13–14.
- [43]. Zhao Hongyang, Wang Shuangquan, Zhou Gang, and Zhang Daqing. 2017. Gesture-enabled remote control for healthcare. In *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 392–401.



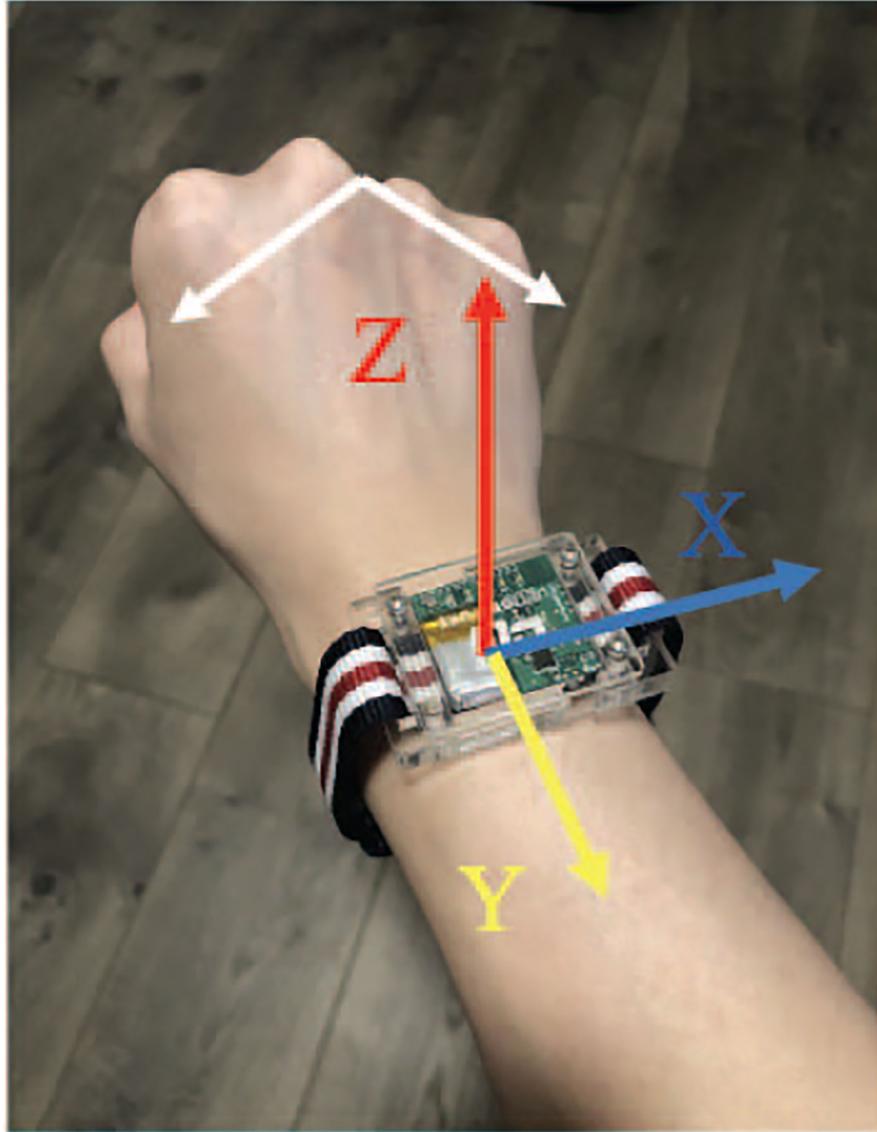
**Figure 1:**  
UG Sensor band

Author Manuscript

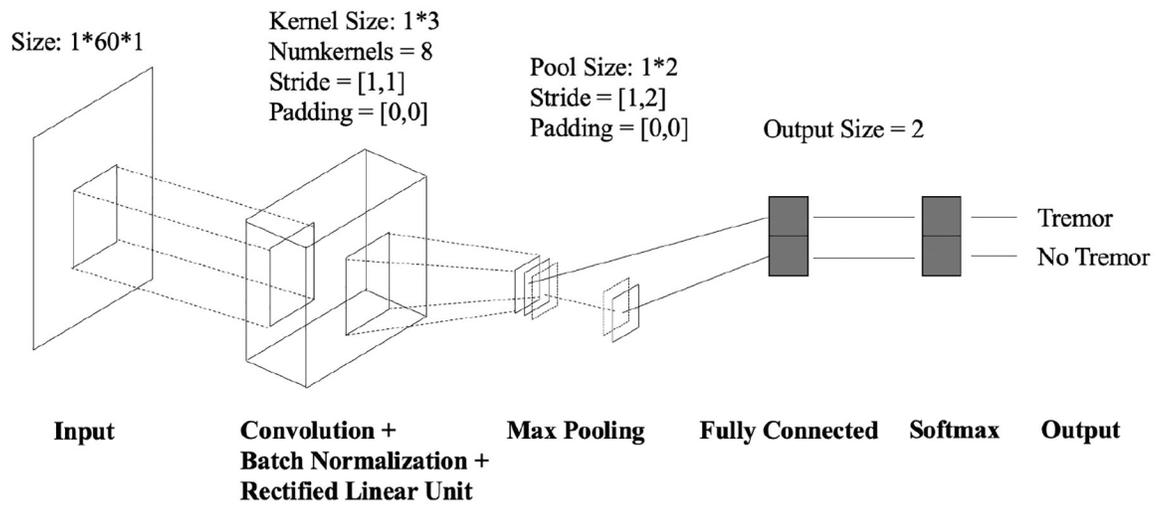
Author Manuscript

Author Manuscript

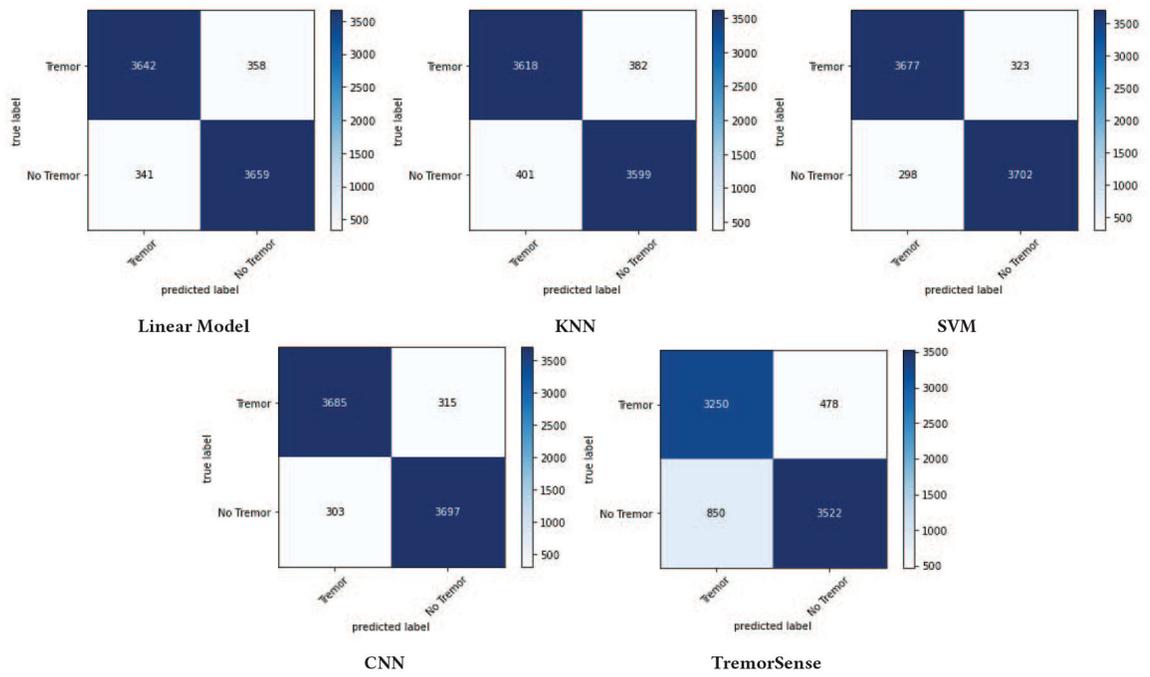
Author Manuscript



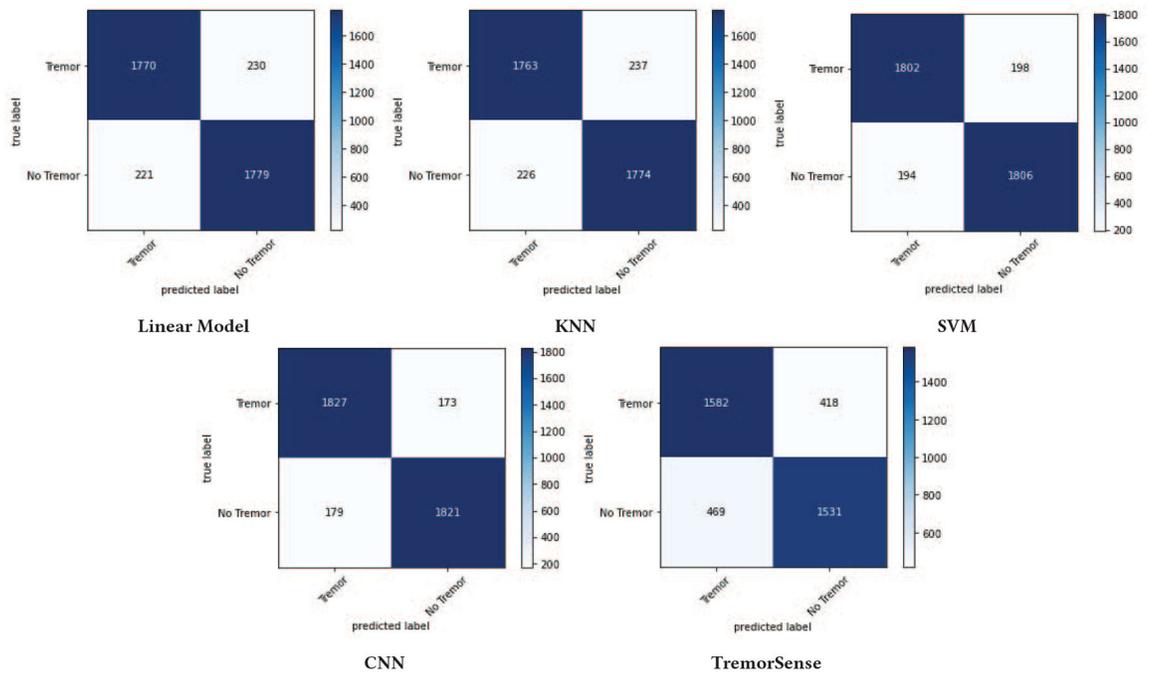
**Figure 2:**  
UG Sensor Axes



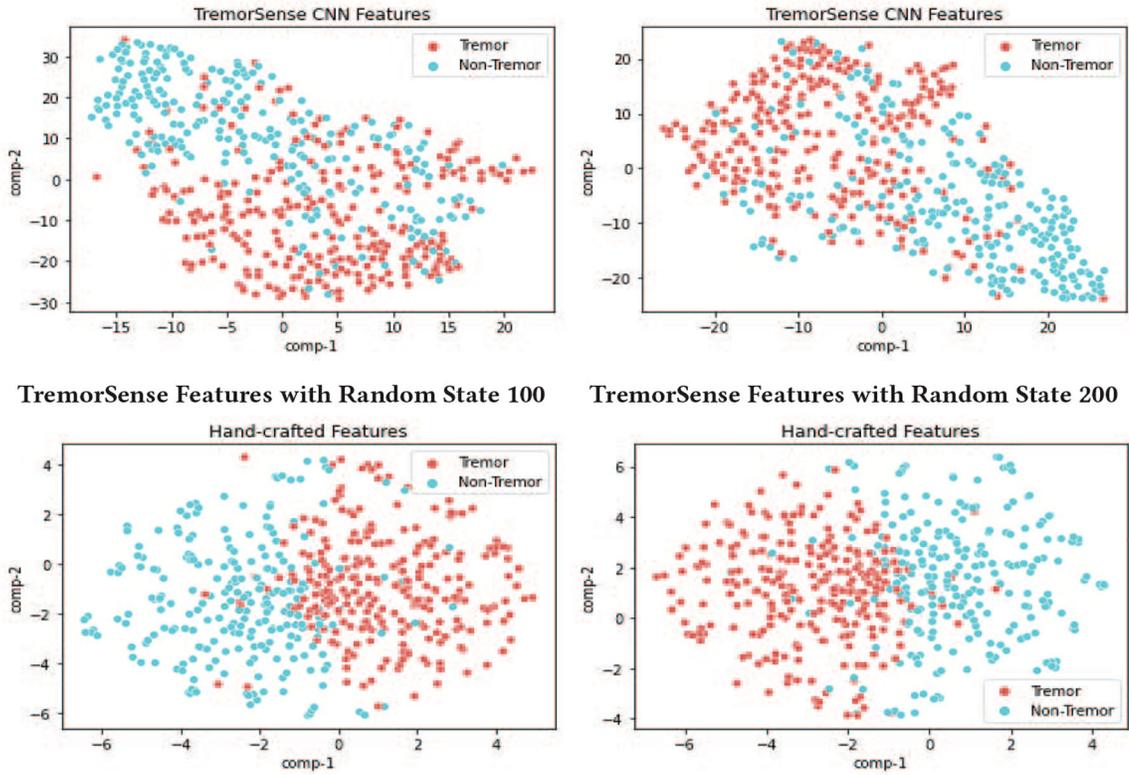
**Figure 3:**  
 Convolutional Neural Network Layers



**Figure 4:**  
Cross-Validation Confusion Matrix



**Figure 5:**  
Leave-one-out Confusion Matrix



Our Hand-crafted Features with Random State 100 Our Hand-crafted Features with Random State 200

**Figure 6:**  
TremorSense and Our Features in t-SNE 2-D Maps

**Table 1:**

## Scales and Activities

Number	Scale	Activity
1	UPDRS	Finger Tapping
2	UPDRS	Hand Movements
3	UPDRS	Hands Pronation-Supination
4	UPDRS	Toe Tapping
5	UPDRS	Leg Agility
6	UPDRS	Gait
7	UPDRS	Kinetic Tremor of The Hands
8	Fahn-Tolosa-Marin	Handwriting
9	Fahn-Tolosa-Marin	Drawing
10	Fahn-Tolosa-Marin	Pouring Water

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 2:**

## Patient Demographics

Characteristics	Details
Patient Number	30
Gender	18 Males / 12 Females
Patient Age	45 – 84 Years old
Patient Average Age	67.43 Years old
PD Symptom Onset Age	35 – 82 Years old
PD Diagnosed Age	38 – 82 Years old
Disease Duration Years	0–24 Years
Disease Duration Average Years	8.80 Years

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 3:**

## Extracted Features

Domain	Features
Time-domain Features	Min, Max, Mean, Standard deviation, Zero crossing / Mean value crossing
Frequency-domain Features	Energy: Sum of FFT component magnitudes of the input sensor signal. Entropy: It measures of the distribution of frequency components. Spectral-centroid: Spectral power distribution balancing point. First three dominant frequencies and their magnitudes between 2Hz-10Hz

## Selected Features

**Table 4:**

<b>Domain</b>	<b>Selected Features</b>
Time-domain Features	All-axis Mean, Std, Zero crossing / Mean value crossing except ACC(Y)
Frequency-domain Features	All-axis frequency-domain features except ACC(Y)

**Table 5:**

Top 20 selected features

Features*	Rank	Importance Score	Features	Rank	Importance Score
Gyro(Y) first Frequency	1	0.0483	Gyro(Y) Energy	11	0.0234
Gyro(Y) StD	2	0.0377	Gyro(X) first Frequency	12	0.0227
Gyro(Y) Mean	3	0.0325	Gyro(Y) first Magnitude	13	0.0197
Acc(X) first Frequency	4	0.0318	Acc(X) Entropy	14	0.0196
Acc(Z) first Frequency	5	0.0306	Gyro(X) StD	15	0.0195
Acc(X) StD	6	0.0297	Acc(X) Energy	16	0.0185
Gyro(Y) Entropy	7	0.0294	Acc(X) first Magnitude	17	0.0184
Acc(Z) StD	8	0.0292	Acc(Z) first Magnitude	18	0.0182
Acc(X) Mean	9	0.0288	Acc(Z) Entropy	19	0.0181
Acc(Z) Mean	10	0.0256	Gyro(Y) Zero crossing	20	0.0179

\* Acc: Accelerometer; Gyro: Gyroscope. (X), (Y), (Z), represent X, Y, Z axes of the sensors.

**Table 6:**

## Cross-evaluation Confusion Matrix

Classifiers	Accuracy	Precision	Recall	F1-Score
Linear Model	91.37%	91.44%	91.05%	91.24%
KNN	90.21%	90.02%	90.45%	90.51%
CNN	92.24%	92.50%	91.98%	92.21%
SVM	92.28%	92.40%	92.13%	92.26%
TremorSense	83.60%	87.18%	79.27%	83.04%

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 7:**

Leave-one-out Evaluation Confusion Matrix Results

Classifiers	Accuracy	Precision	Recall	F1-Score
Linear Model	88.70%	88.86%	88.50%	88.68%
KNN	88.43%	88.64%	88.15%	88.39%
CNN	90.20%	90.28%	90.10%	90.19%
SVN	91.20%	91.08%	91.35%	91.21%
TremorSense	77.83%	77.13%	79.10%	78.10%

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript