IMPROVING EMG MOVEMENT CLASSIFICATION ACCURACY WITH RELATIVE ENTROPY

A Thesis

Presented to

The Faculty of the Department of Computer Science

California State University, Los Angeles

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

By

Kevin Osvaldo Delao

December 2020

The thesis of Kevin Osvaldo Delao is approved.

Navid Amini, Committee Chair

Negin Forouzesh

Eun-Young Elaine Kang, Department Chair

California State University, Los Angeles

December 2020

ABSTRACT

Improving EMG Classification Accuracy with Relative Entropy

By

Kevin Osvaldo Delao

Currently the Degrees of Freedom (DoF) possible from an Electromyography (EMG) controlled prosthetic arm is low due to the challenge of classifying movements. Low DoF leads to prosthetics having limited capabilities in their applications for rehabilitation. Machine learning algorithms offered the potential to improve the accuracy of identifying movements from an EMG dataset, but even still accuracy for identifying EMG movements remains low. Part of the reason as to why classification accuracy remains low is due to the shortage of publicly available algorithms for EMG classification, which prevents experienced computational researchers from tackling the challenge of classification accuracy. Additionally, EMG movements recorded from non-invasive electrodes tend to overlap due to similarities in signals properties. Movement overlap in turn creates difficulties for machine learning models to properly identify movements in an EMG dataset. This work will present research in order to begin to develop a publicly available set of machine learning algorithms for EMG classification that uses the Kullback-Leibler Divergence (KLD) to reduce movement overlap. By applying KLD to resolve signal overlap and create optimal movement orderings per patient we take steps at improving the classification accuracy of movements from EMG data. Achieving a five percent increase in classification accuracy across fifty movements for both amputated and non-amputated patients shows the promise of KLD at improving the DoF possible with an upper limb prosthesis.

ACKNOWLEDMENTS

I would like to thank Dr. Navid Amini for all his help in the time I have been here at Cal State LA. It was through his support that not only have I finished my thesis project, but also managed to get into competitive summer research programs and was also included in my first research publication. His continued support has allowed me to thrive as a researcher and I plan to use what I have learned and apply it in my PhD studies. His advice on his experience as a researcher has definitely helped me solidify my choice to continue on as a researcher and apply for PhD programs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure

CHAPTER 1

Introduction

Surface Electromyography (sEMG) classification has historically been a task that required a great deal of effort to accomplish [2]. When sEMG recordings were first utilized to record electrical signals from muscles, machine learning algorithms were not as widely utilized as today. Researchers were originally required to spend large amounts of time carefully inspecting a sEMG signal to determine which movements were occurring in various parts of the signal [6]. Researchers in the field of neural prosthesis were especially hindered by the time-consuming nature of sEMG analysis, because there was no method to create an autonomous algorithm that would be able to quickly recognize sEMG signals and translate them to movement commands that a prosthetic device could then replicate.

The introduction of machine learning finally allowed researchers to have a fast and autonomous method to analyze large amounts of sEMG data. Prosthetic researchers utilizing machine learning were able to create algorithms capable of classifying small sets of movements from sEMG recordings with high accuracy. Various papers began to publish results stating that classification accuracies of over ninety percent have been achieved when using machine learning for sEMG classification [17,21]. While the results in these papers were promising, the problem was that the methods used in such research papers were difficult to replicate. Research groups did not make their algorithms publicly available and the sEMG data they worked with could not be released due to the privacy of patients. This meant that other research teams working on sEMG movement classification had to essentially start from scratch to develop machine learning algorithms to classify their data. The inability for research teams to replicate each other's work also meant that performance for sEMG classification could not be compared due to sEMG recordings drastically varying depending on setup and patient.

The NinaPro Project introduced one of the first publicly available sEMG databases that allowed researchers to finally have a method to compare classifier performance. The NinaPro Project consists of nine databases each containing recordings from 10-40 patients performing between 30-53 different hand movement variations [3]. Each of the databases have been described in detail in terms of acquisition protocol and have been validated to ensure that the sEMG data present is usable for machine learning classification [3]. The creators of the NinaPro Project hoped that by having a publicly available source of sEMG data, computational researchers would be able to build algorithms for movement classification and have a benchmark to compare against. One reason why having a benchmark for sEMG classification is important is that currently the movement classification accuracy for sEMG data containing more than twenty movement variations is low [3]. Having a publicly available sEMG database was meant to allow researchers to build upon each other's work in order to eventually build a classifier able to distinguish between many movements with high accuracy.

An issue that concerns the classification of movements within sEMG data is the accessibility of code. The introduction of the NinaPro Project finally allowed researchers to have publicly available sEMG data, but it is still rare for researchers to make their code accessible to anyone. This is in stark contrast to other fields of research such as Natural Language Processing or Image Recognition where there are many code tutorials describing in detail on how to use machine learning to recognize text or images. The lack of any publicly available algorithms for sEMG classifications creates a barrier preventing novice and veteran researchers from trying to tackle the problem of sEMG classification. Another issue that concerns sEMG movement classification is the variability and separability between movements. Humans produce unique sEMG signals that are different from any other person [6]. This variability that exists between sEMG signals causes problems for machine learning algorithms, because a machine learning algorithm that trains on one patient's sEMG data may not perform well on another patient's sEMG dataset. Within patient variability is also a concern for sEMG classification, because as the number of movements to classify increases so does the overlap between movements [1]. In order to improve the classification accuracy for sEMG data, variability and overlap need to be considered when using machine learning to classify movements from EMG data.

The work presented here will go over the research conducted which aims to establish a publicly available set of algorithms for sEMG classification. Part of what this project aims to achieve is to have a heavily documented set of algorithms for sEMG classification that will allow other researchers to utilize the code for prosthetic training or to build upon the algorithms for their own research. The other aim of this project will be to improve sEMG classification accuracy by utilizing the Kullback–Leibler divergence which has seen success in sound research to separate sound patterns [4]. Additionally, this thesis will extend past work on applying the Kullback–Leibler divergence to EMG data in order to improve classification accuracy [22]. The Kullback–Leibler divergence will be used to create machine learning algorithms that will consider the variability between patients, while also maximizing the separability criteria in order to reduce the overlap between movements [11]. The exact details of how the Kullback–Leibler divergence is used to improve sEMG classification will be discussed in Chapter 4.

The work will be organized as follows: Chapter 2 will discuss the sEMG data that will be used, namely databases one and three of the NinaPro Project. Chapter 3 will go over data preprocessing such as filtering, windowing, feature extraction, and gesture identification that will be performed on the sEMG data. Chapter 4 will go over in detail the machine learning algorithms that will be used and why they were chosen. Chapter 5 will go over the results obtained from the classifiers. Chapter 6 will discuss the research results. Lastly, Chapter 7 will go over the conclusions for the research conducted for this thesis.

CHAPTER 2

Data Information

In order to build a classifier capable of distinguishing between movements in an sEMG dataset, the sEMG data used for training and testing needs to be valid. If the sEMG data used for classification was not replicable in real world recording setups, then the classifier performance on said dataset would be useless. Fortunately, the NinaPro Project has been thoroughly validated to ensure that each of the recording setups used to record the sEMG data stored is replicable in the real world [3].

The databases that will be used in this work will be databases one and three of the NinaPro Project. Only two databases were used due to the long computation required for each database and time constraints. Each database differs drastically in terms of setup and recording protocol and as such each database will be described in detail in this section to ensure that differences between their acquisition and experimental protocols are conveyed. Additionally, Database 2 was discussed briefly as well due to its data overlap with Database 3.

Database 1 Analysis

Database 1 in the NinaPro Project was the first attempt at creating a publicly available sEMG database. The database consists of sEMG recordings from twenty-seven intact patients performing fifty-two hand movement variations. The actual sEMG recordings were collected using ten active double differential OttoBock MyoBock 13E200 sEMG electrodes which were placed on various muscles on the arm [2]. Specifically, eight electrodes were placed around the forearm and two electrodes were placed on the extensor and flexor muscles of the forearm [2]. The sEMG signal recorded by the electrodes is a raw sEMG signal, but the actual sEMG recordings stored on NinaPro are not raw signals [2]. The OttoBock MyoBock electrodes provide signals that are bandpass filtered and Root Mean Rectified meaning that the sEMG signals stored do not contain negative values. Rectification is conducted on sEMG signals because often features are extracted from sEMG data that require taking the average [2,6]. If negative sEMG values are present, then the average of the sEMG signal will be zero.

Kinematic hand and wrist data were collected using a 22-sensor CyberGlove II dataglove and a 2-axis Kubler IS40 inclinometer [2]. Kinematic data was recorded to allow researchers to use data to study the position of the movements conducted by the patients and to use the kinematic data as ground truth. Ground truth is needed because sEMG signals are inherently noisy so in order to determine what movement a patient conducted at a given time, kinematic data is needed for verification.

The acquisition protocol for the electrodes and inclinometer were acquired at a constant interval of 100Hz using a DAQ (Digital Acquisition) card [2]. The Cyberglove was recorded over a Bluetooth-tunneled serial port at 25Hz [2].

The experimental protocol for the movement exercises consisted of the twenty-seven patients replicating movements seen on a screen that were divided into three sets as seen in Figure 1. The movements were twelve finger movements, eight isometric and isotonic hand configurations, nine wrist movements, and twenty-three grasping movements [2]. Exercises consisted of each patient performing ten repetitions for each of the fifty-two movements [2]. Each repetition lasted for five seconds and was followed by a three second rest period. The rest period was not classified as a movement, but it is classified as a movement by the machine learning algorithm which will be discussed in Chapter 4.

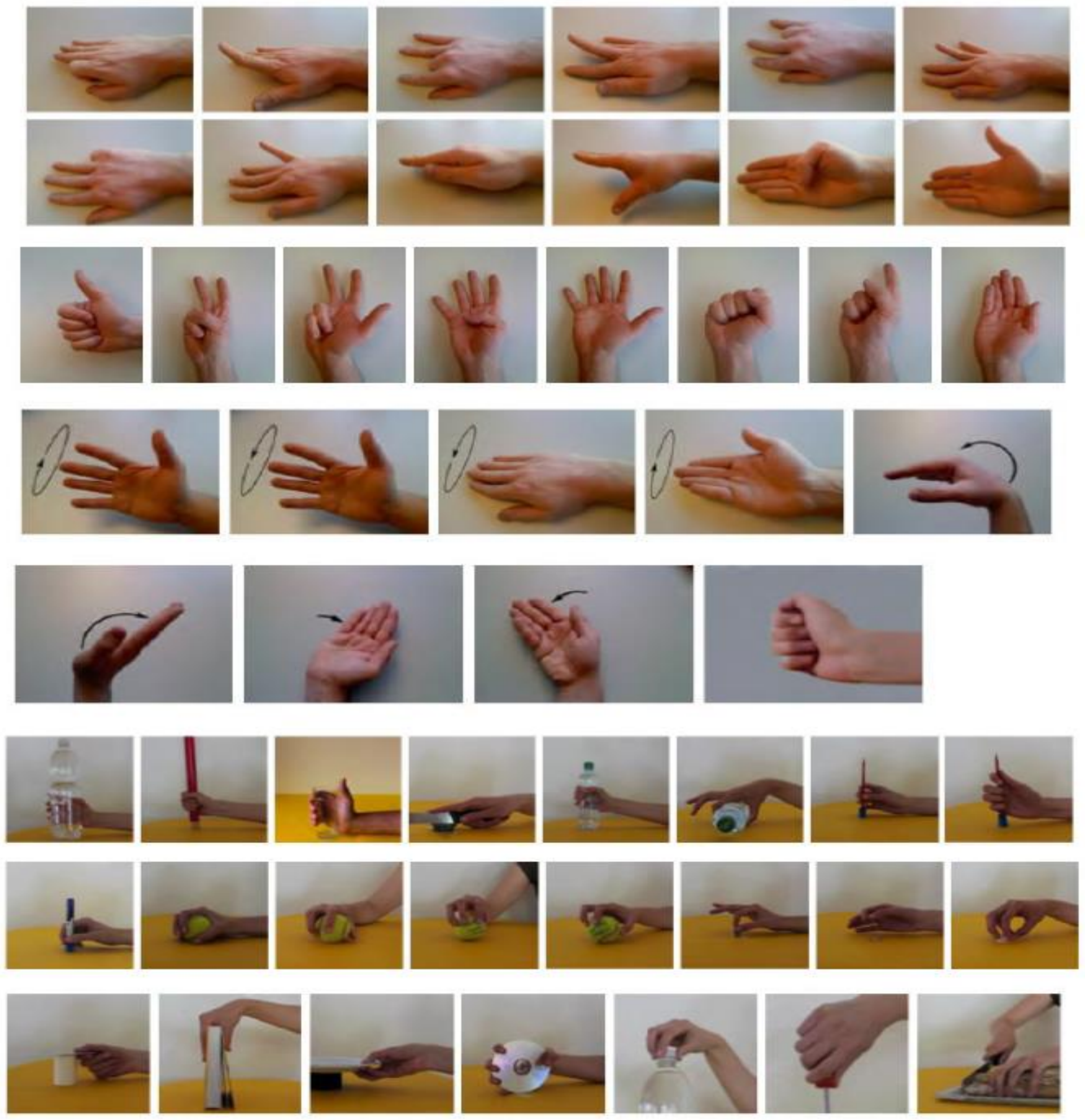


Figure. 1. Database 1 Movements

Database 2 Analysis

The second database in the NinaPro database is like database one but differs in some aspects of the acquisition and experimental protocol. One immediate difference in the acquisition protocol of database 2 is that movements were recorded using Delsys double-differential sEMG electrodes instead of OttoBlock electrodes [3]. The experimental protocol differed in that database 2 contained forty intact patients in total and each patient performed fifty movements [3]. Demographics of the patients in database 2 were also different in that database 2 contained forty intact subjects consisting of twenty-eight males, twelve females, thirty-four right-handed, six left-handed, and ages of 29.9±3.9 years [3]. It is unknown how much of a difference the ratio of male to female will affect classifier performance as Body Mass Index (BMI) has been shown to affect sEMG recording accuracy [1].

The experimental protocol for movements exercises was like database 1. Patients performed movement exercises by replicating the actions they saw on a screen. Each exercise had six repetitions and there was a rest period between each repetition. The movements chosen for database 2 involved various grasping and flexions as seen in Figure 2 [3]. Figure 2 contained various grasping and hand extensions.

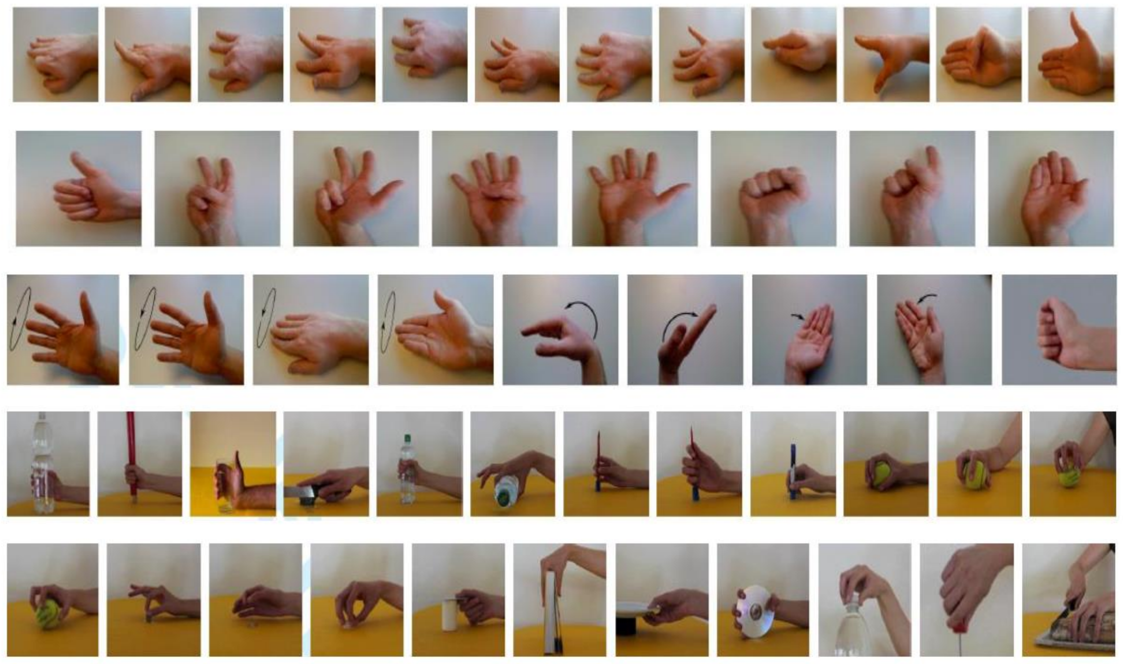


Figure. 2. Database 2 and 3 Movements

Database 3 Analysis

Database 3 of the NinaPro Project was the first of the databases to include trans-radial amputee patients as opposed to just intact patients. This was vital because amputees differ greatly in terms of the nerve endings present in the arm muscles. Trans-radial amputees have far less nerve endings in their arm muscles due to surgery and as such the sEMG recordings from amputees are much noisier [3]. Regardless of the increased noise present for amputees, it is nonetheless crucial to include amputees in sEMG recordings. The reason being is that if researchers want to develop more advanced prosthetic devices for amputees, then it is imperative to have an algorithm capable of classifying the sEMG data for amputee patients.

Currently non-invasive myoelectric prosthetic devices are some of the most popular prosthetic devices to use for amputee patients due to their non-invasive application, but myoelectric prosthetics suffer from only being able to perform a limited range of motions. Part of the reason why motions are limited is due to low classification accuracy for sEMG data from amputee patients. Having a publicly available benchmark for sEMG data for amputees will allow for the development of algorithms capable of classifying sEMG with higher accuracies and lead to more advanced prosthetic devices capable of more complex movements.

The database itself is similar to database 2 in that it involves using 12 Delsys double-differential sEMG electrodes for the recordings [3]. Database 3 also involves the same fifty movements seen in Figure 2. Repetitions per movement were also like database 2 in that each patient performed six repetition per movement with rest in between each movement.

The main differences for database 3 are in the demographics of the patients and the experimental protocol. Database 3 only includes eleven amputee patients, with each patient varying in type of amputation as seen in Table 1 [3].

TABLE 1. DATABASE 3 PATIENT DEMOGRAPHIC

Subject	**Amputated Hand**	**Remaining Forearm** (%)	**Years Since Amputation**	**Phantom Limb Sensation** (0-5)
1	Right	50	13	2
2	Left	70	6	5
3	Right	30	5	2
4	Right and Left	40	1	1
5	Left	90	1	2
6	Left	40	13	4
7	Right	0	7	0
8	Right	50	5	2
9	Right	90	14	5
10	Right	50	2	5
11	Right	90	5	4

Due to amputated patients missing their upper limb, repetitions were performed by a patient thinking of the movement they saw on the screen. It is important to note that ground truth data is not possible with amputee patients as they cannot operate sensors such as the CyberGlove on their missing limb [3]. In order to circumvent the problem of ground truth data, database 3 chose to have the stimulus serve as ground truth data [3].

CHAPTER 3

Data Preprocessing

In order to prepare the sEMG data for classification, several preprocessing steps need to be taken. Filtering for noise removal, windowing, feature extraction, and gesture identification are a few of the steps needed to be able to pass the sEMG data through a machine learning algorithm. This chapter will go over in detail each of the processing steps taken and why they were chosen.

Noise Sources

In order to discuss the use of filtering methods, it is first crucial to understand where noise sources come from for sEMG recordings. As mentioned previously one of the main drawbacks of using sEMG for muscle recordings is the large amount of noise present in the signal. The type of noise in sEMG signals has been heavily documented in the past by other researchers in order to categorize the types of noise that sEMG signals are prone to. The first type of noise that sEMG signals are prone to is electrical noise also called inherent noise that originates from electrical equipment [8]. Inherent noise is mainly removable through the design of the hardware used for recording, but some filters can help deal with the noise. The filtering method used to remove this type of noise will be discussed in the filtering section.

Movement artifacts is another type of noise that is heavily present in sEMG recordings. Movement artifacts is a type of noise that originates from the cables that connect the sEMG electrodes to the amplifier and the contact between the electrodes themselves and skin [8]. Normally preparation such as cleaning or shaving of the skin is performed to prevent movement artifacts from occurring. This type of noise was not possible to remove through software alone in this project, so it was not focused on for filtering methods.

Electromagnetic noise originating from the body and from the environment is noise that is present in all sEMG signals regardless of the quality of the recording setup. In order to properly deal with electromagnetic noise often a high pass filter is utilized [8]. Database 1 had its useful information in the low frequency band of 0-25Hz so a high pass filter to remove electromagnetic noise was not needed for database 1.

Another source of noise that is common in nearly all sEMG recordings is muscle crosstalk. In the simplest terms muscle crosstalk is an sEMG signal from an unwanted muscle group [8]. The sEMG signals from muscle groups that are not being focused on can interfere with the sEMG signals of interest and can lead to incorrect analysis of the sEMG signals. The main method to remove muscle crosstalk is in the design and placement of the electrodes. Filtering methods have almost no effect on the removal of muscle crosstalk and because of this no filtering methods were used in the removal of muscle crosstalk in this project [8].

Electrical activity from the heart also called Electrocardiogram (ECG) can also interfere with sEMG signals. Placement of electrodes closer to the trunk and shoulder muscles can lead to an increase in ECG signals that can contaminate the sEMG signals of interest. Placement of electrodes on the arm muscles as was done in database 1-3 can still be affected by ECG artifacts. One of the main difficulties in removing ECG noise is that ECG signals tend to overlap with sEMG signals making it hard for filtering methods to recognize and remove ECG artifacts [6]. There have been suggestions to remove ECG artifacts by using a high pass 100hz filter, but because sEMG signals in database 2 and 3 contain relevant information in the 100hz range a high pass filter was not used for database 2 and 3 [8]. Database 1 has relevant information in the low frequency domain, so ECG noise was not a concern for database 1.

Filtering Steps

As mentioned previously the sEMG signals stored for database 1 are not raw sEMG signals. The sEMG signals for database 1 contain signals that have been bandpass filtered and root mean squared (RMS) rectified [1, 2]. What this prefiltering entails is that the sEMG signals for database 1 contain signals that have a bandwidth of between 0-25Hz. Normally raw sEMG signals have a bandwidth of 15 to 500Hz, so the bandwidth for database 1 is much smaller [6]. An RMS rectified signal means that the signal does not contain any negative values as all raw sEMG contain. Due to the prefiltering onboard the OttoBlock electrodes the relevant information for the sEMG signals in database 1 lie in the low frequency spectrum band [9]. In order to remove noise from the signal and remove any high frequency noise that originates from electrical equipment, a low pass second order Butterworth filter with a cutoff frequency of 5Hz was used for each of the ten channels. Butterworth filters have long been used methods to remove noise from sEMG signals as Butterworth filters seem to be able to remove sEMG noise without distorting the signal [16]. As can be seen from Figure 3 there is a drastic reduction in the noise present after the Butterworth filter was passed through each channel. Figure 3 was generated by applying a low pass 5Hz Butterworth filter to an EMG signal from a patient from database 1. The Butterworth filter removed any EMG signals above 5Hz and the Butterworth filter also smoothed out the EMG signal to remove unnecessary EMG activity.

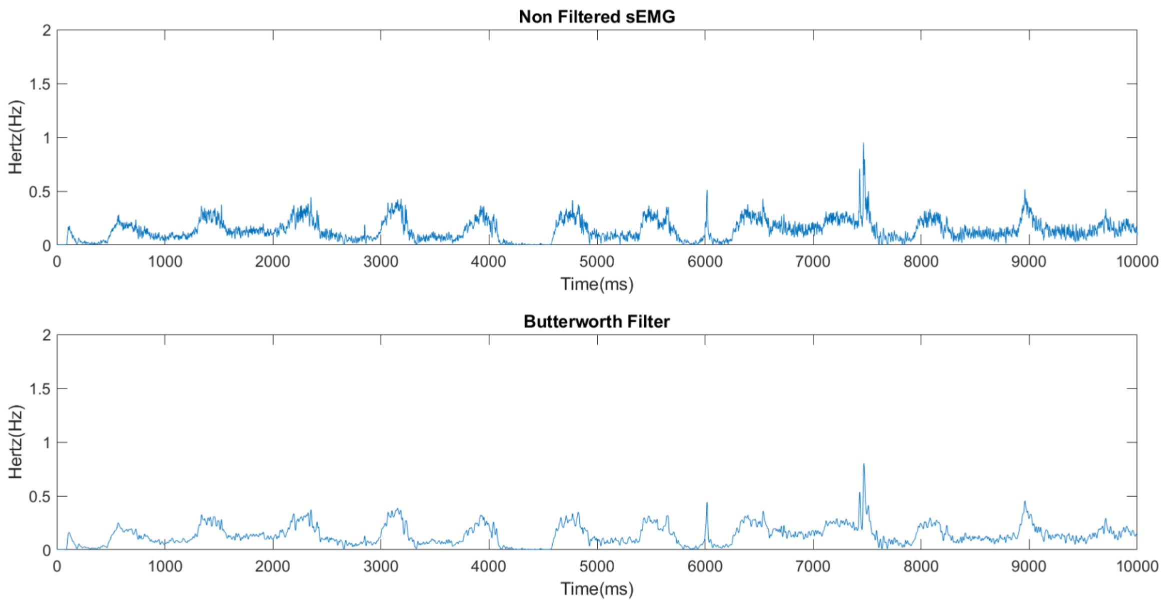


Figure. 3. Database 1 Filtering

Database 3 requires a few extra filtering steps as opposed to database 1, because database 3 contains raw sEMG data. The sEMG data in database 3 had to be rectified first in order to allow for the extraction of features that take the mean of the signal. The absolute value of the signal was taken in order to fold over the negative sEMG values to the positive domain as seen in Figure 4. Figure 4 was generated by taking the absolute value of the EMG signal of a patient from database 3. For Figure 4, only a portion of the EMG signal is shown in order to make it easier to visualize taking the absolute value of the EMG signal.

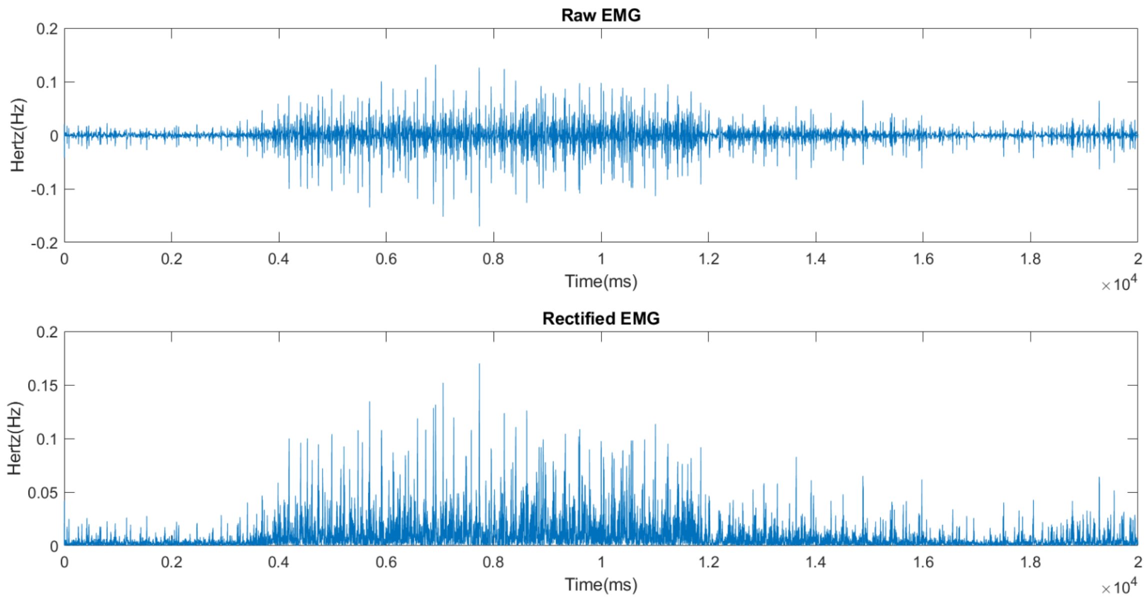


Figure. 4. Database 3 Rectification

The next step in processing the sEMG data for database 3 involves the removal of the noise present in the signal. Like database 1 a Butterworth filter with a cutoff frequency of 500Hz was applied to the signal as seen in Figure 5. Figure 5 was generated by applying the Butterworth filter to the EMG signal of a patient from database 3. The Butterworth filter caused EMG signals from database 3 to be smoothed out by reducing unnecessary EMG activity for signal analysis. Additionally, the 500Hz cutoff meant that the EMG data for database 3 that was above 500Hz was removed and this was justified as raw EMG signals have important information below 500Hz [6]. The reason for the 500Hz cutoff frequency is due to the sampling rate for database 3 being 2kHz. After the application of the Butterworth filter, there is an immediate reduction to the noise present.

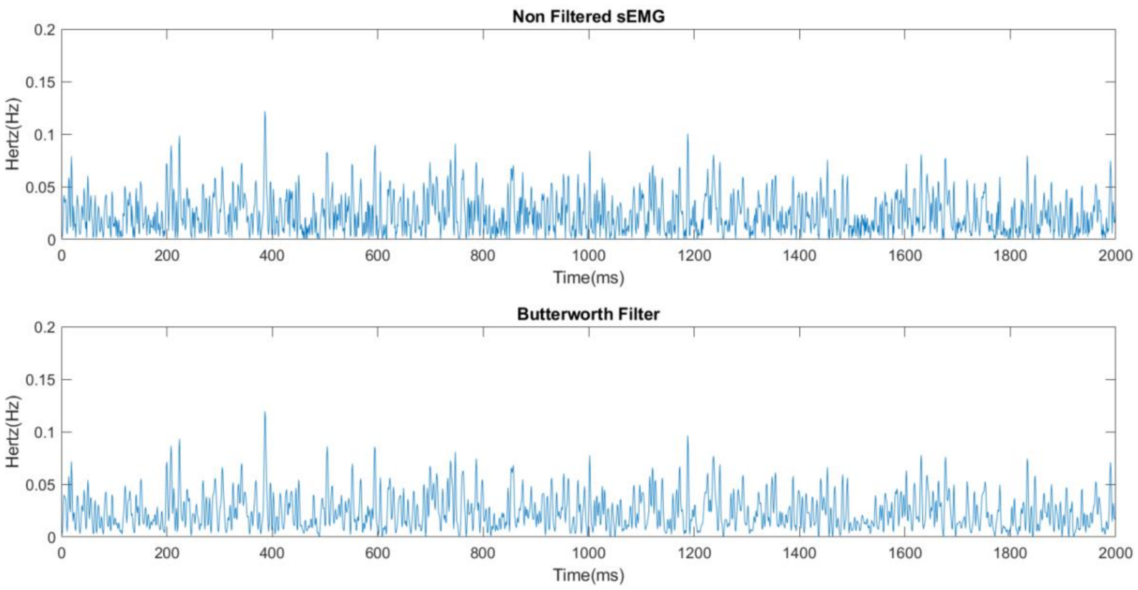


Figure 5. Database 3 Filtering

Windowing

To properly analyze the sEMG data in database 1, the sEMG signals need to be segmented into continuous windows. Windowing is needed, because sEMG data contains continuous movements. In order to determine what activity a given part of a sEMG signal corresponds to, not just one segment of an sEMG signal can be focused on. Following well known windowing techniques, the sEMG data in database 1 were segmented into 400 millisecond(ms) windows with an increment of 10ms [15]. Other popular window lengths include 100ms and 200ms windows [2,23]. Care does need to be taken when selecting a window length as bigger window lengths introduce more noise, but too small a window length can lead to misclassifications. A 400ms window length provides ample representation of the movement occurring in part of a signal without much noise being introduced. After the sEMG data of all twenty-seven patients were segmented into 400ms windows, the sEMG data was ready to be used for gesture identification and feature extraction.

Feature Extraction

The common features for sEMG signals can be divided into three main groups: time domain features, frequency domain features, and time-frequency domain features [18]. Features related to time for sEMG signals capture information related to the amplitude of the sEMG signal. Frequency domain features on the other hand capture information related to the frequency characteristics of the sEMG signal [18]. For time domain features there are currently thirteen well known features that are utilized for sEMG movement classification, for this project three-time domain features were chosen for classification [13]. Specifically mean absolute value, variance, and root mean square were the time domain features chosen due to their simplicity and performance in other literature [2]. Time-domain features are the most popular features to use for sEMG classification because they are easy to compute and do not require a transformation of the sEMG signal [18]. The drawback of using time-domain features is that they reduce sEMG signal information to scalar values, which leads to a loss of information [2]. Even with loss of information simple time-domain features have been shown to perform as well or better than more complicated frequency and time-frequency domain features [2].

The actual extraction of time-domain features was extracted from the windowed data by taking the mean absolute value, variance, and root mean square of 400ms windows across ten channels for database 1 and twelve channels for database 3. The feature data was then saved for later use when it will be used to calculate the movement ordering per patient and for the training of the machine learning algorithms.

Gesture Identification

The sEMG data for database 1 was prelabeled using an offline labeling calculation that determined which of the fifty-two movements including rest were occurring every 10ms [2]. As mentioned previously a single 10ms segment of movement is not enough to determine what movement action is occurring during a longer period. In order to properly determine what movement a patient performed during a timespan, majority voting was used to identify every 10ms window as belonging to a single movement in a 400ms window. Each 10ms second window was identified as belonging to either one of the fifty-two movements from database 1 or forty-nine movements from database 3 or rest, by taking a majority vote in a 400ms window length. The movement that had the most votes in the 400ms window was then assigned as the movement that occurred in that window. Increments of 10ms were continuously repeated until the end of the sEMG data, where in each increment majority voting in a 400ms window length were calculated and assigned to every 10ms segment.

CHAPTER 4

Machine Learning Classification

Kullback–Leibler Divergence

One of the issues for the classification of sEMG data containing many movements is the overlap between movements. Overlap leads to a poor classification boundary which in turn leads to low classification accuracy. In order to increase the movement separability there needs to be a method to be able to distinguish how similar movements are in each movement group. The Kullback-Leibler Divergence (KLD) that comes from the field of mathematical statistics will allow for the measurement of how similar movements are from each other and allow for a movement ordering that is specific per patient [24].

Using KLD made calculations that determine similarities between movements quicker as distributions can represent complex data in a simpler manner. Working with movement EMG data itself to determine optimal movement ordering per patient would have been much more computationally intensive as every possible movement pairing and movement comparison would have needed to be calculated to determine the most optimal movement ordering per patient.

Before describing how KLD will be utilized for movement separability, KLD needs to be described in terms of what it does. At its core KLD determines how one probability distribution differs from another expected distribution [19]. The measure of how different two distributions are or how one distribution resembles another is measured through divergence. The value of the divergence between two probability distributions calculated in Figure 6 will determine how similar the two distributions are. In Figure 6

P(x) represents the primary distribution where P is the distribution representing some data x. Q(x) is the reference distribution which approximates the information represented by P(x). Figure 6 will calculate a divergence value between P(x) and Q(x) by calculating the log difference between P(x) and Q(x) which will generate D where D represent the divergence between P(x) and Q(x). If two distributions match perfectly then the divergence value D will be zero and the closer the divergence value is to zero, then the closer two probability distributions are.

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)}\right)$$

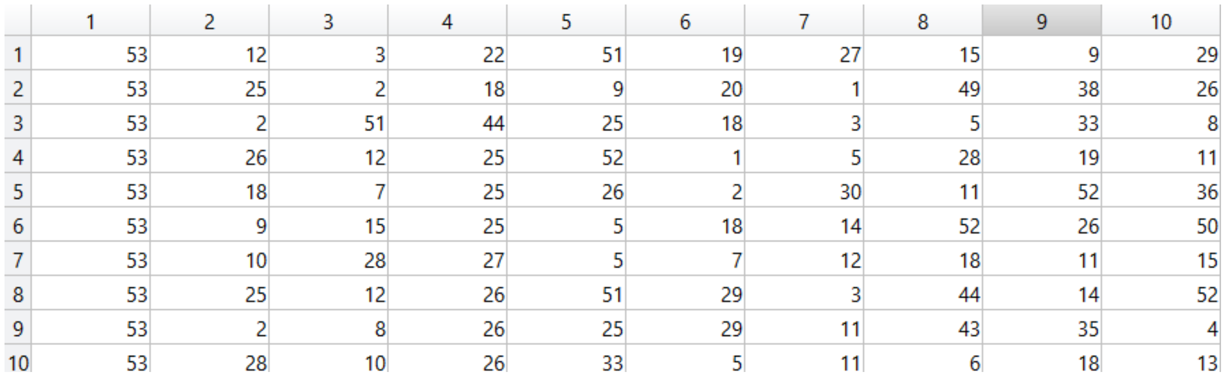
Figure 6. Kullback-Leibler Equation

Normally divergence is used for modeling to make sure that one distribution matches another in terms of the information it represents, but KLD can also be used for separability. Normally movements from sEMG recordings are added one at a time to a pool of movements in which a classifier creates a boundary to identify different movements [1]. In one of the first papers by the creators of the NinaPro Project, movements were added in a sequence starting from rest [1]. Adding movements in a sequence may seem like the most optimal method for classification, but patients often perform movements better or worse than each other. In order to properly address the varying performance for movements between each patient, movements should be added more selectively to a movement group.

To ensure that separability between movements is maximized, it is more optimal to add new movements to a movement pool by how much they differ from the movements already selected. Normally it would take far too much time to compare every already selected movement to every new movement we wish to add, but distributions allow for the representation of large amounts of data using a minimum amount of information. Using KLD we can create a single distribution to represent the movements already selected for classification and compute the divergence between the selected movement distribution to each of the not yet selected movement distributions. After computing the divergence, instead of choosing the not yet selected movement distribution with the smallest divergence we can instead pick the not yet selected movement with the largest divergence and add that movement next to the movement pool. The justification for choosing to add the not yet selected movement with the largest divergence value is due to the divergence value representing similarity between distributions as mentioned previously. If a movement holds a large divergence value then that would indicate that the two distributions do not match, but it also means that the information represented by the distributions are different. This is vital because movements that hold different information from each other will have less overlap. A movement that had the largest divergence value would differ the most from the already selected movements and would lead to a less chance of overlap when added to the movement pool for classification.

The process of computing the divergence between the selected movement distribution and each of the not yet selected movement distributions and then choosing the movement with the largest divergence to add is continued until there are no more movements left to add. By using KLD, each patient will have a unique ordering of movements as seen in Figure 7 that will ensure a decrease in movement overlap which will in turn lead to improved classification accuracy. Figure 7 shows the order in which movements are added to a group for machine learning classification for database 1.

Figure 7 is just an example showing that KLD will generate an ordering to add movements to a group for every patient, where the orderings will ensure that movements added initially have less of a chance of overlapping and all movements that do cause overlap are added last to a group for classification.



	1	2	3	4	5	6	7	8	9	10
1	53	12	3	22	51	19	27	15	9	29
2	53	25	2	18	9	20	1	49	38	26
3	53	2	51	44	25	18	3	5	33	8
4	53	26	12	25	52	1	5	28	19	11
5	53	18	7	25	26	2	30	11	52	36
6	53	9	15	25	5	18	14	52	26	50
7	53	10	28	27	5	7	12	18	11	15
8	53	25	12	26	51	29	3	44	14	52
9	53	2	8	26	25	29	11	43	35	4
10	53	28	10	26	33	5	11	6	18	13

Figure 7. Database 1 KLD Ordering

Environment

Before discussing the machine learning classifiers used in this project, it is first important to go over the language and environment used to create this project. MATLAB version 2019b was the sole coding language used to create this project. The reason why MATLAB was chosen over other coding languages is because the default format for the sEMG data from the NinaPro Project are MATLAB files. The MATLAB version used in this project also included the Machine Learning and Deep Learning ToolBox as well the Signal Processing ToolBox which were both required for filtering and machine learning classification. As mentioned previously the code used in this project will be made public on GitHub, but in order to be able run the code properly it is important to make sure that the ToolBoxes installed, and MATLAB versions are similar.

Machine Learning Classifiers

For database 1 because there were only ten repetitions that served as the train and test data, care needed to be taken in deciding how to split the data. Papers have mentioned that depending on how the data is split for sEMG recordings, there could be potential differences in classification results due to patients adapting to movements [3]. However, these papers also mention that adaptation to movements is rare and only impacts a few patients [3]. In order to compare the results of KLD ordering to the original ordering of movements seen in the first paper by the NinaPro Project, a split for database 1 was chosen using repetitions {1 3 5 7 9} as the training set and repetitions {2 4 6 8 10} as the testing set [1]. This partition for training and testing was chosen based on the methodology performed from previous work [1]. The split itself was chosen because adjacent repetitions run the risk of EMG overlap, thus to avoid signal overlap only every other repetition was chosen for training and testing [1]. For databases 3, the split was randomly selected where four out of the six repetitions were chosen for training and the remaining two repetitions were used for testing.

It is important to note that due to the rest movement representing a large portion of the movements, rest movements were randomly down sampled so that it only represented as many movements as the next most common movement. Random down sampling of the rest movement was done for databases 1 and 3 in order to not have a large bias for rest.

After the data was split and random down sampling was applied, the data was passed into six classifiers. Namely K-Nearest-Neighbor (KNN) with K equaling ten, Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) with a Radial

Basis Function (RBF) kernel, Linear SVM, Decision Tree (DT) with 150 splits, and Random Forest (RF) with fifty splits. KNN with a K value of ten was chosen based on the popularity of using ten as a standard. LDA was set up with the default parameters MATLAB provides as LDA is a simple linear classifier. SVM was set up with the multiclass one vs one parameter because SVM used a linear kernel. RBF was set up with the multiclass one vs all parameter as RBF kernels try to establish a classifier using all the data provided. DT was chosen with 150 splits because having many splits in decision trees leads to better classification accuracy results. RF was chosen to have 50 decision trees simply based on my past experience using RF with a moderate amount of decision trees for classification. The justification for choosing six standard machine learning classifiers over a Neural Network (NN) is because part of what the project aimed to achieve is to have a simple solution to a complex problem. Specifically, machine learning algorithms, that offer great performance and are easier to understand than NN's, will allow for computational researchers to more easily understand the code and more quickly utilize the code. Another reason why machine learning algorithms were chosen over NN's is because NN's require more data in order to have higher performance when compared to machine learning algorithms.

The results for the six machine learning algorithms after training on the training set and then testing on the testing set will be discussed in the following Chapter 5.

CHAPTER 5

Data Results

Database 1 Results

Using Mean Absolute Value (MAV) as the feature for the dataset from database 1, adding movements in a sequence resulted in most of the six classifiers having an accuracy on average of below seventy percent as seen in Figure 8. Adding movements in sequence resulted in accuracy that was consistent with the results from the first paper of the NinaPro Project [1]. With the KLD movement orderings seen in Figure 9, classifiers had an improved accuracy for nearly every movement except for movements forty-five and onwards which had similar results as movements ordered in a sequence. KLD movement ordering resulted in an average improvement of four percent when looking at movements 20-45. The reason for this similar performance had to do with the fact that movements that cause overlap are being eventually added to the end for classification which results in KLD ordering no better than sequential movement orderings.

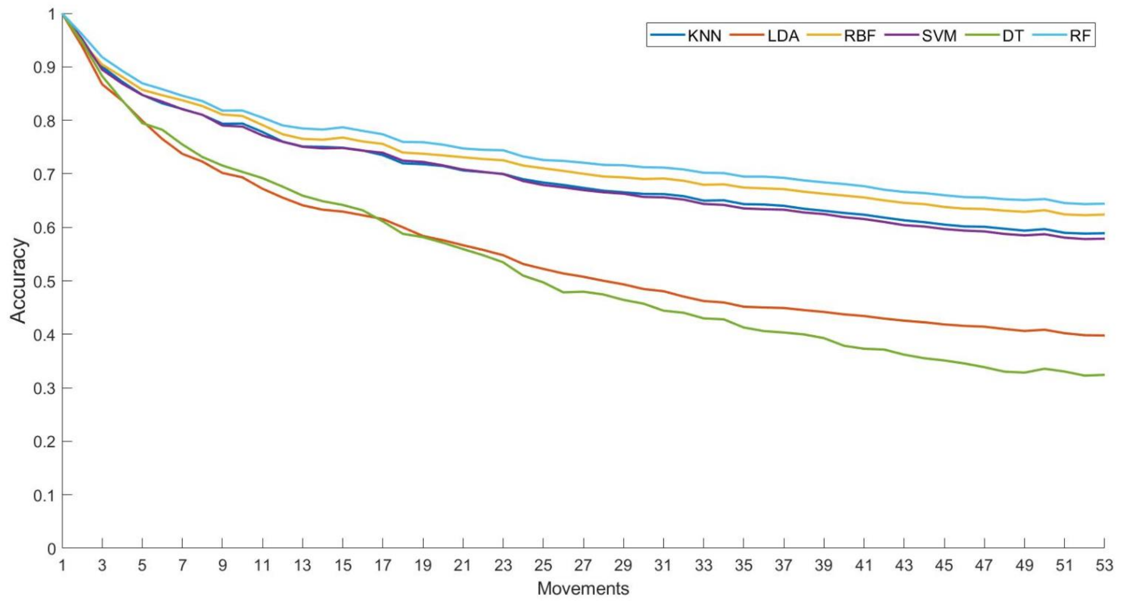


Figure 8. Database 1 Sequential MAV Feature

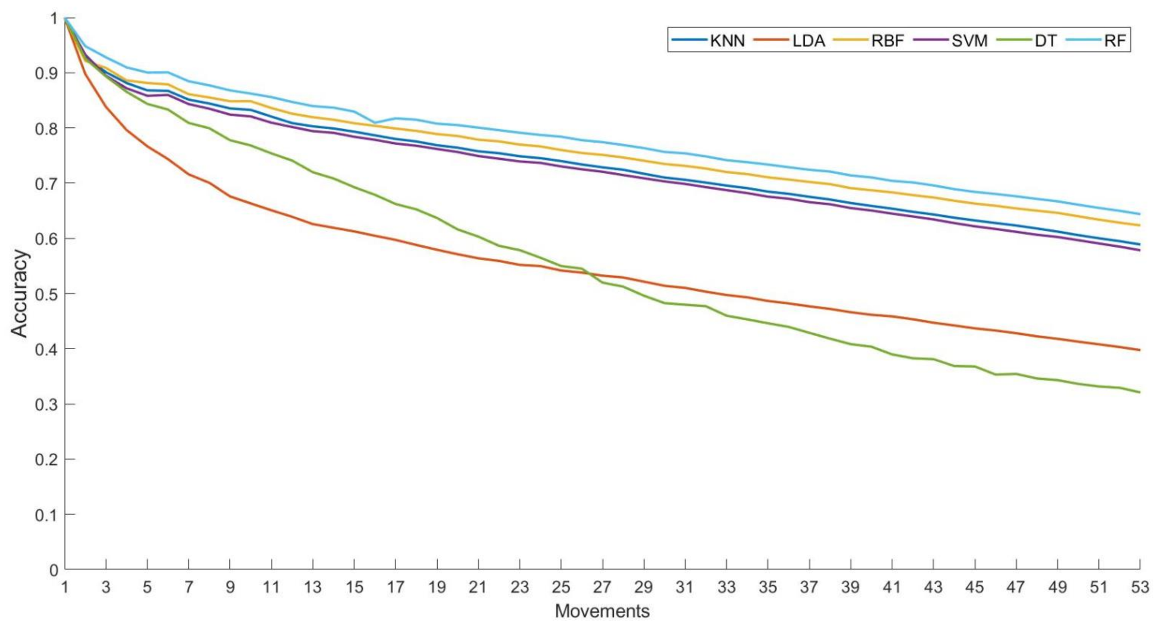


Figure 9. Database 1 KLD MAV Feature

Comparing the best performing classifier of both movement sequence ordering from Figure 8 and the KLD movement ordering from Figure 9, Random Forest (RF) can be seen to have improvements in accuracy for every movement as seen in Figure 10.

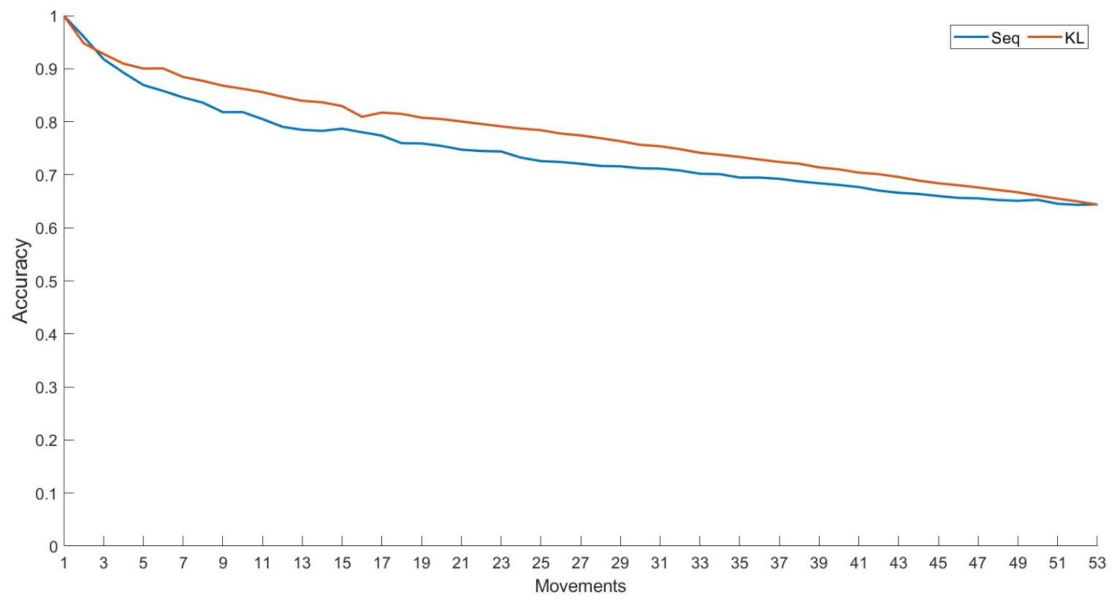


Figure 10. Database 1 Sequential and KLD Comparison

Using Root Mean Squared as the feature for classification, an identical performance to MAV for sequential movement ordering was seen as shown in Figure 11. Similarly, the KLD ordering of movements using RMS as the feature gave similar performance to MAV as seen in Figure 12. The significance of why two different features gave almost identical performance will be discussed in the following section. Comparing KLD and sequential movement ordering using RMS as a feature still showed KLD ordering giving a higher classification accuracy as seen in Figure 12.

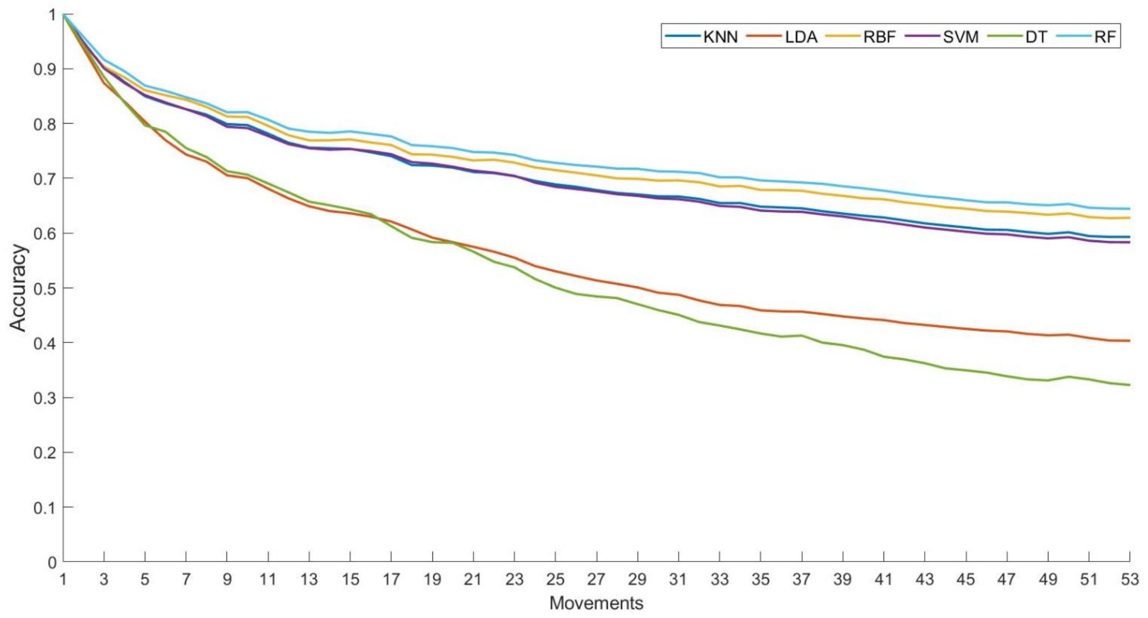


Figure 11. Database 1 Sequential RMS Feature

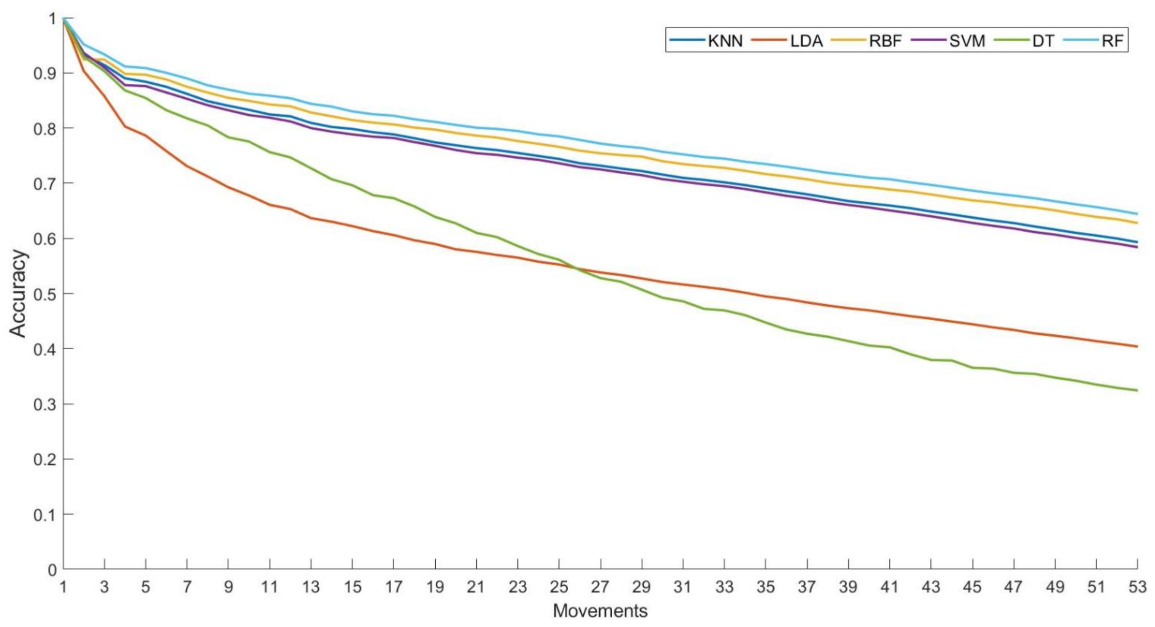


Figure 12. Database 1 KLD RMS Feature

Lastly using Variance (VAR) as the feature for classification gave a different accuracy performance for database 1. As seen in Figure 13 there is a lower accuracy using VAR as the feature for all six classifiers. Regardless of the decreased performance seen using VAR in the Sequential movement ordering, KLD still showed improved accuracy for the six classifiers as seen in Figure 14. Specifically, RF from KLD orderings performs much better than sequential movement orderings and still performs the best from the classifiers. Comparing RF from sequential and KLD movement orderings in Figure 15, we see an almost ten percent improvement to accuracy using KLD movement ordering as opposed to the standard sequential movement ordering.

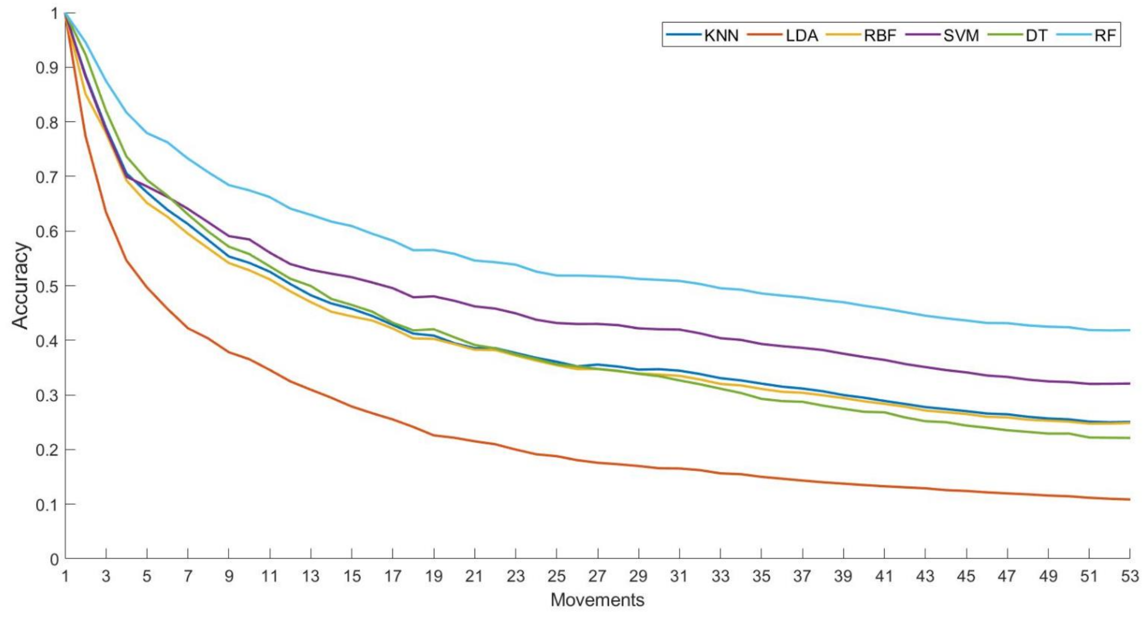


Figure 13. Database 1 Sequential VAR Feature

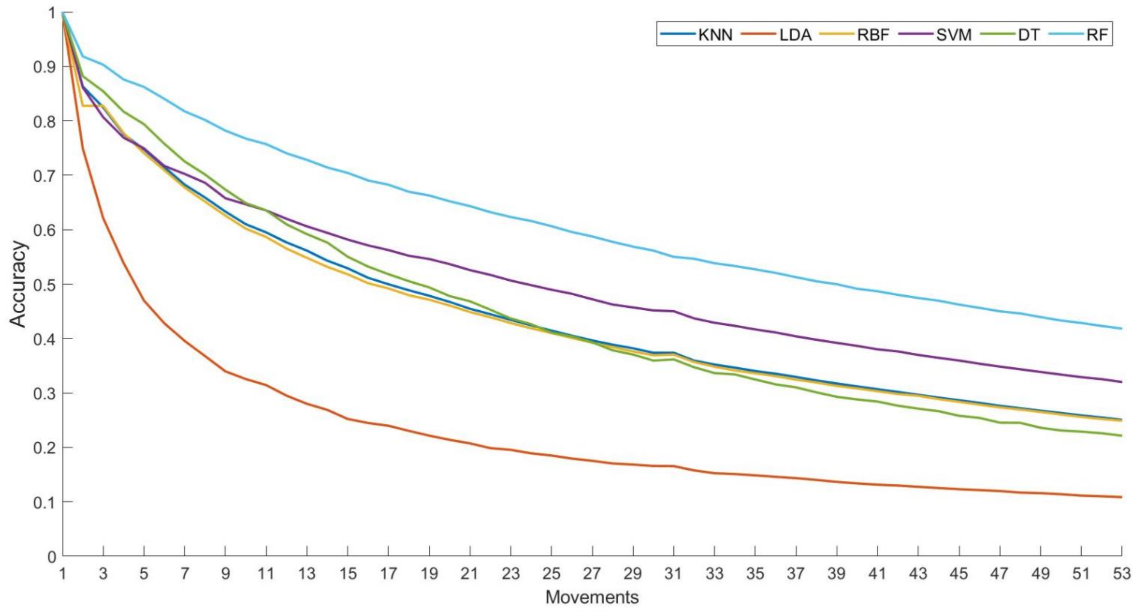


Figure 14. Database 1 KLD VAR Feature

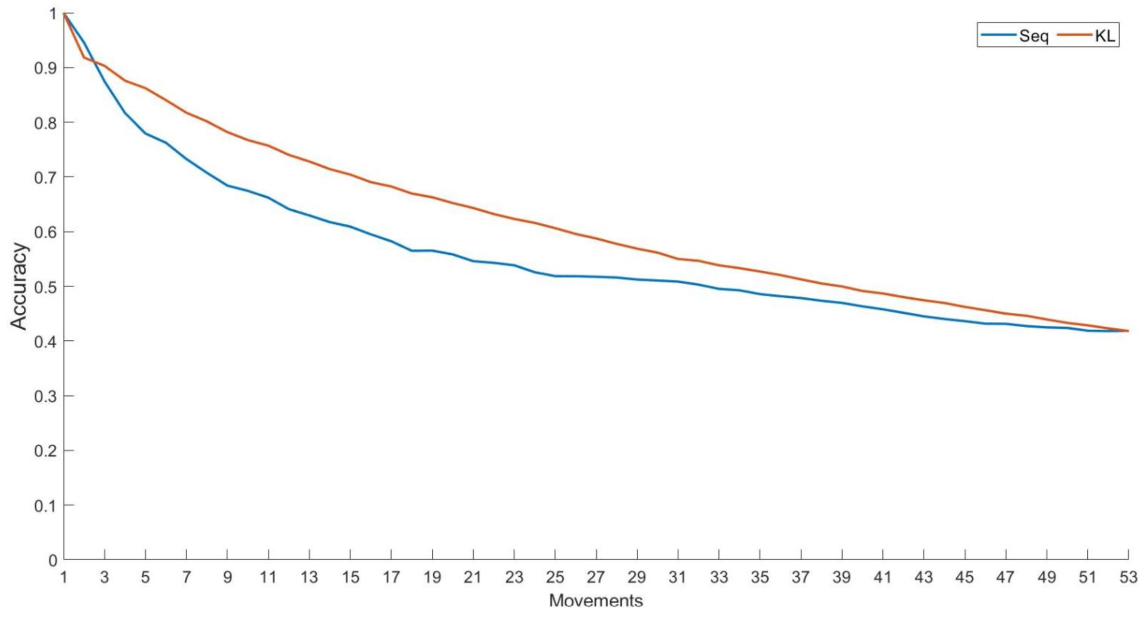


Figure 15. Database 1 Sequential and KLD Comparison VAR

Database 3 Results

Using MAV on Database 3, we see that for sequential movements there is much lower accuracy overall when compared to non-amputated patients. The reason as stated before is that amputated patients have damage to their nerve endings, which causes much higher noise in EMG recordings. We see that for sequential movement orderings in Figure 16, the accuracy hovers around thirty percent for the majority of movements. When KL ordering is applied in Figure 17 however, we see jumps of ten to twenty percent accuracy for most movements. Comparing the two results in Figure 18 we see a ten percent improvement for KLD when compared to sequential movements. Lastly using VAR as a feature in Figure 19, we see a ten percent improvement for KLD, but near the last ten movements KLD actually performed worse than sequential.

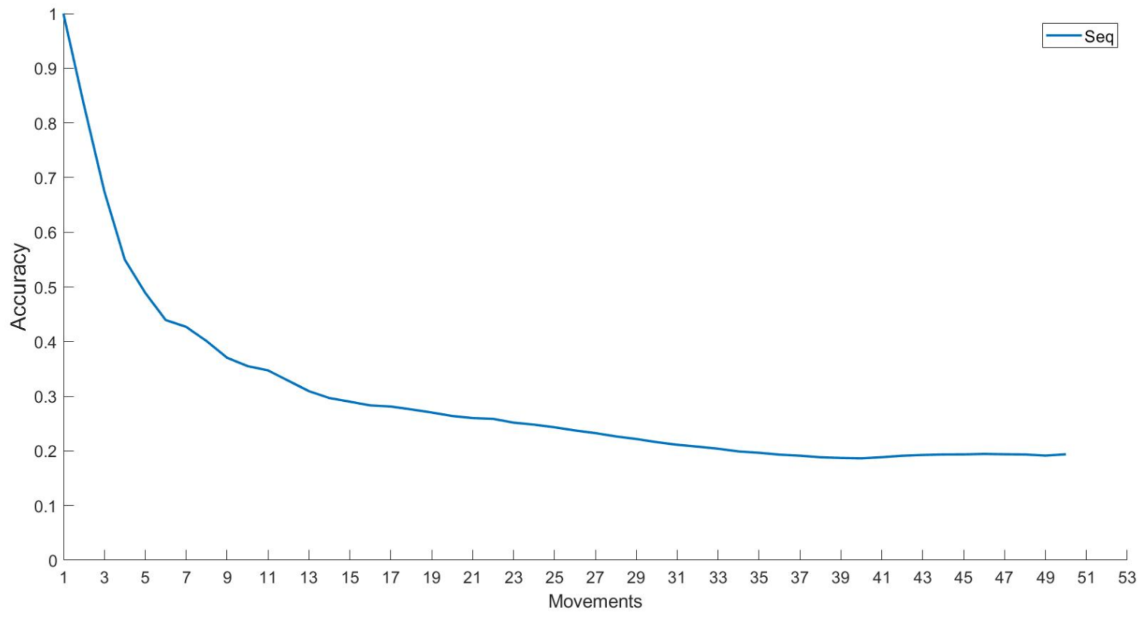


Figure 16. Database 3 Sequential RF MAV

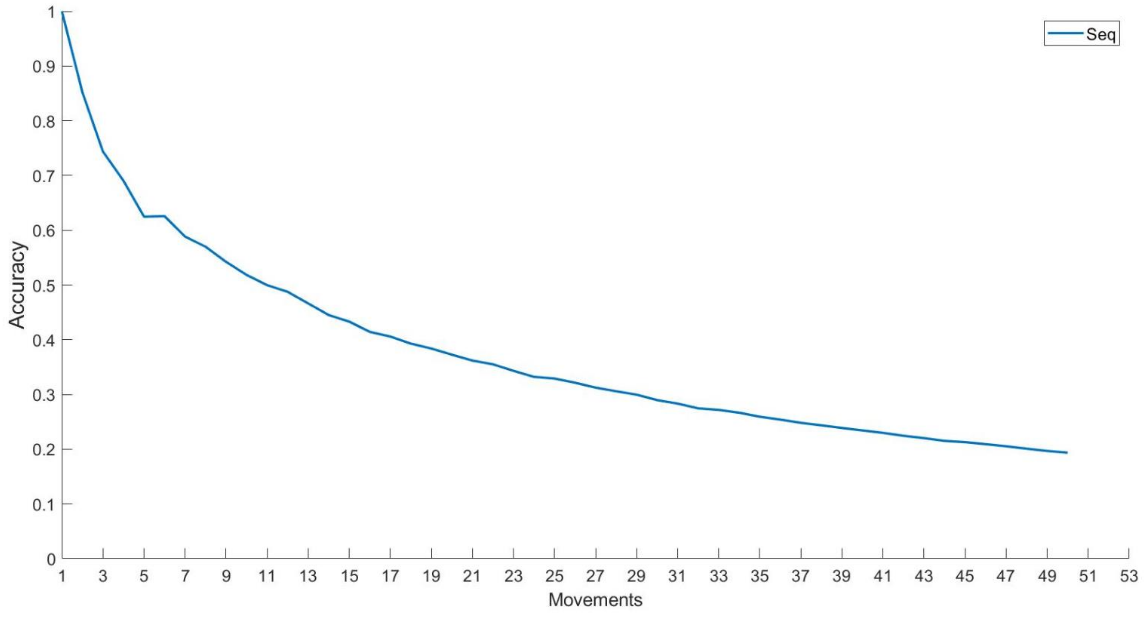


Figure 17. Database 3 KLD RF MAV

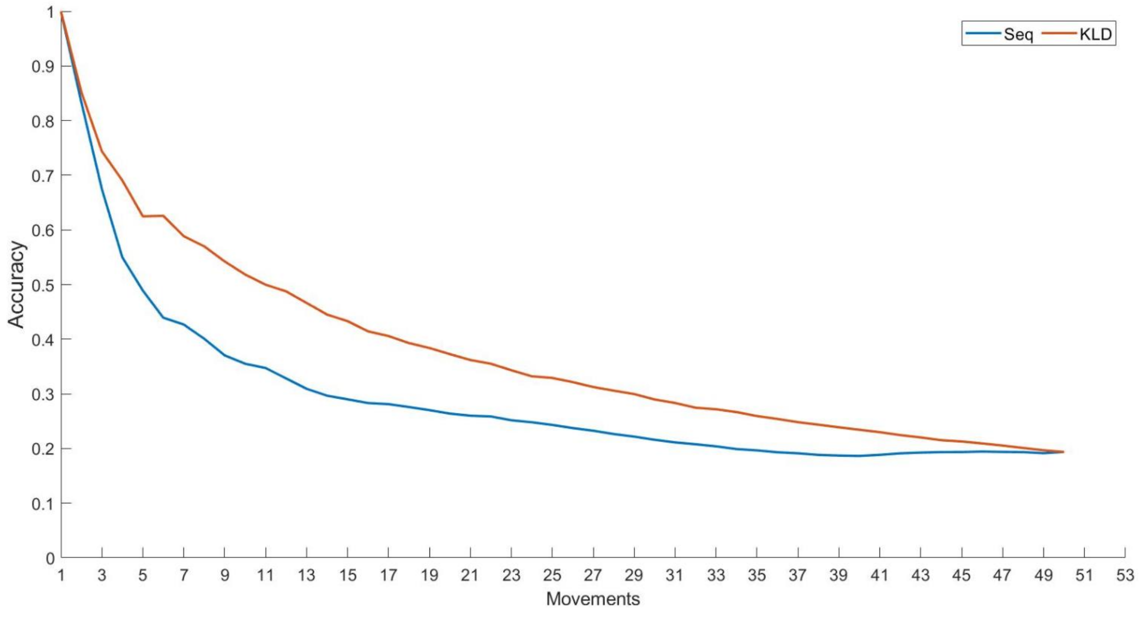


Figure 18. Database 3 Sequential and KLD Comparison MAV

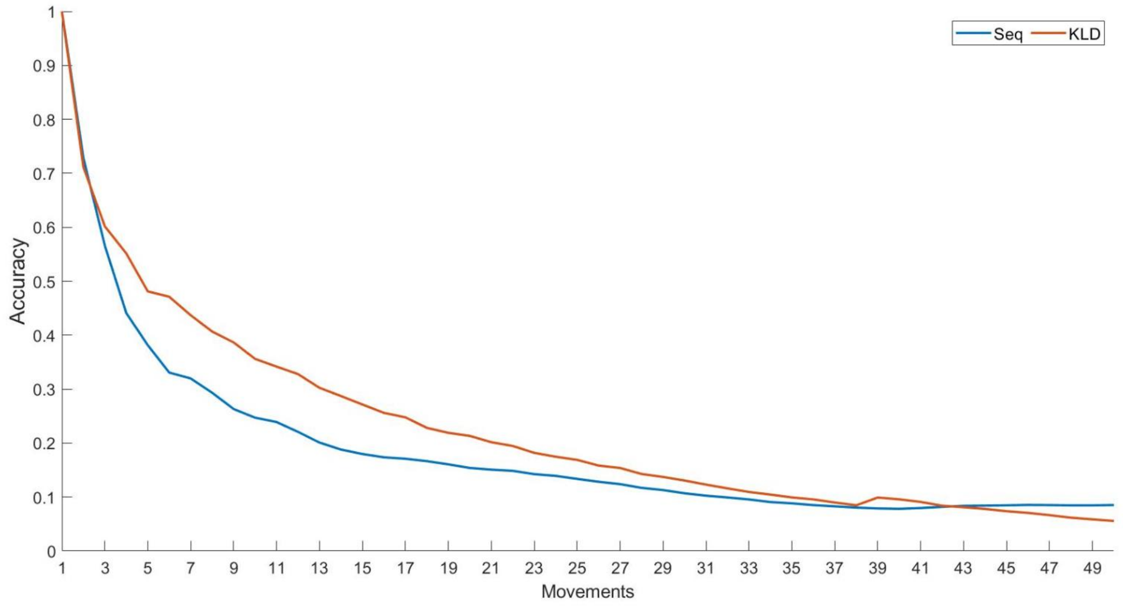


Figure 19. Database 3 Sequential and KLD Comparison VAR

CHAPTER 6

Discussion

Based on the results from Chapter 5, KLD ordering seemed to improve classifier performance on all three features used. Classifiers that had the best performance overall such as SVM with an RBF kernel, and Random Forest saw the largest improvements with KLD orderings. Specifically, RBF and RF saw an average of a ten percent improvement in accuracy for movements five to forty-five from database 1. Other classifiers such as SVM Linear, KNN, and DF saw smaller two percent and three percent improvements per movement. LDA performed the worst out of the six classifiers and the performance of LDA did not seem to improve with KLD movement ordering. LDA's poor performance could be attributed to the classification problem in question. Movements that overlap slightly are difficult for linear classifiers to classify even with decreases in movement overlap.

The performance of KLD movement ordering showed improvement per movement, but KLD seemed to converge in performance with sequential movement orderings for movements past forty. One explanation for the convergence in performance is that KLD movement orderings must eventually add movements with lots of overlap. KLD tries to add movements one at time by how they differ from other movements, but eventually all the movements that have a lot of overlap will be added last. Movements with lots of overlap that are added near the end will reduce classifier performance and eventually cause KLD ordering to perform equally to sequential movement ordering as seen in the results chapter.

The three features used for classification represent different time-feature representations of the amplitude of the sEMG signal, but as seen in the results chapter MAV and RMS had identical performance. The explanation for similar performance has to do with how MAV and RMS function. MAV takes the mean absolute value from the signal and RMS squares the signal and then takes the square root. Signals from database 1 were already rectified so only positive values were used, which meant that MAV and RMS were both taking the mean of the signal which led to their identical performance.

CHAPTER 7

Conclusion

The work presented here demonstrated that statistics can be used in conjunction with machine learning algorithms in order to improve classifier performance for sEMG movement recognition. Using the Kullback-Leibler Divergence, movement overlap that reduces classifier performance was able to be reduced. Along with improvements in classifier performance, the research conducted for this thesis can used as a guide in order to allow computational researchers to delve into the sEMG classification problem.

Future work to improve the sEMG classification presented here could include utilizing Deep Learning to reduce sEMG noise and utilizing Deep Learning to classify sEMG movements. There has been success by researchers who have used Deep Learning instead of conventional machine learning algorithms when it came to classifying movements [20]. Deep Learning algorithms develop their own features so they could learn patterns of movement better than standard machine learning algorithms. Future steps will be to begin implementing a novel Deep Learning algorithm for the use of EMG classification.

REFERENCES

1. Atzori, M., Gijsberts, A., Heynen, S., Hager, A. G. M., Deriaz, O., Van Der Smagt, P., et al. & Müller, H. (2012, June). Building the Ninapro database: A resource for the biorobotics community. In 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob) (pp. 1258-1265). IEEE.
2. Atzori, M., Gijsberts, A., Kuzborskij, I., Heynen, S., Hager, A. G. M., Deriaz, O., et al. & Caputo, B. (2013). A Benchmark Database for Myoelectric Movement Classification. Transactions on Neural Systems & Rehabilitation Engineering.
3. Atzori, M., Gijsberts, A., Castellini, C., Caputo, B., Hager, A. G. M., Elsig, S., et al. & Müller, H. (2014). Electromyography data for non-invasive naturally controlled robotic hand prostheses. Scientific data, 1(1), 1-13.
4. Clim, A., & Zota, R. D. (2019). The Kullback-Leibler Divergence Class in Decoding the Chest Sound Pattern. *Informatica Economica*, *23*(1).
5. Atzori, M., Gijsberts, A., Müller, H., & Caputo, B. (2014, August). Classification of hand movements in amputated subjects by sEMG and accelerometers. In 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 3545-3549). IEEE.
6. Konrad, P. (2005). The ABC of EMG: A Practical Introduction to Kinesiological Electromyography.
7. Pizzolato, S., Tagliapietra, L., Cognolato, M., Reggiani, M., Müller, H., & Atzori, M. (2017). Comparison of six electromyography acquisition setups on hand movement classification tasks. PloS one, 12(10).

8. Chowdhury, R. H., Reaz, M. B., Ali, M. A. B. M., Bakar, A. A., Chellappan, K., & Chang, T. G. (2013). Surface electromyography signal processing and classification techniques. Sensors, 13(9), 12431-12466.

9. Kuzborskij, I., Gijsberts, A., & Caputo, B. (2012, August). On the challenge of classifying 52 hand movements from surface electromyography. In 2012 annual international conference of the IEEE engineering in medicine and biology society (pp. 4931-4937). IEEE.

10. Mankar, V. R. (2011). EMG signal noise removal using neural networks. In Advances in Applied Electromyography. IntechOpen.

11. Nilsson, N., Håkansson, B., & Ortiz-Catalan, M. (2017). Classification complexity in myoelectric pattern recognition. Journal of neuroengineering and rehabilitation, 14(1), 68.

12. Reaz, M. B. I., Hussain, M. S., & Mohd-Yasin, F. (2006). Techniques of EMG signal analysis: detection, processing, classification and applications. Biological procedures online, 8(1), 11-35.

13. Phinyomark, A., Hirunviriya, S., Nuidod, A., Phukpattaranont, P., & Limsakul, C. (2011). Evaluation of EMG feature extraction for movement control of upper limb prostheses based on class separation index. In 5th Kuala Lumpur International Conference on Biomedical Engineering 2011 (pp. 750-754). Springer, Berlin, Heidelberg.

14. Cram, J. R. (1998). Introduction to surface electromyography. Aspen publishers.

15. Englehart, K., & Hudgins, B. (2003). A robust, real-time control scheme for multifunction myoelectric control. IEEE transactions on biomedical engineering, 50(7), 848-854.

16. De Luca, C. J., Gilmore, L. D., Kuznetsov, M., & Roy, S. H. (2010). Filtering the surface EMG signal: Movement artifact and baseline noise contamination. Journal of biomechanics, 43(8), 1573-1579.

17. Tenore, F. V., Ramos, A., Fahmy, A., Acharya, S., Etienne-Cummings, R., & Thakor, N. V. (2008). Decoding of individuated finger movements using surface electromyography. IEEE transactions on biomedical engineering, 56(5), 1427-1434.

18. Zecca, M., Micera, S., Carrozza, M. C., & Dario, P. (2002). Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews™ in Biomedical Engineering*, *30*(4-6).

19. Clim, A., Zota, R. D., & TinicĂ, G. (2018). The Kullback-Leibler Divergence Used in Machine Learning Algorithms for Health Care Applications and Hypertension Prediction: A Literature Review. Procedia Computer Science, 141, 448-453.

20. Côté-Allard, Ulysse, et al. "Deep learning for electromyographic hand gesture signal classification using transfer learning." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.4 (2019): 760-771.

21. Kanitz, G. R., Antfolk, C., Cipriani, C., Sebelius, F., & Carrozza, M. C. (2011, August). Decoding of individuated finger movements using surface EMG and input optimization applying a genetic algorithm. In *2011 Annual International*

Conference of the IEEE Engineering in Medicine and Biology Society (pp. 1608-1611). IEEE.

22. Hartwell, A., Kadirkamanathan, V., & Anderson, S. (2016, August). Person-specific gesture set selection for optimised movement classification from emg signals. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 880-883). IEEE.
23. Ashraf, H., Waris, A., Gilani, S. O., Kashif, A. S., Jamil, M., Jochumsen, M., & Niazi, I. K. (2020). Evaluation of windowing techniques for intramuscular EMG-based diagnostic, rehabilitative and assistive devices. *Journal of Neural Engineering*.
24. Hershey, J. R., & Olsen, P. A. (2007, April). Approximating the Kullback Leibler divergence between Gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07* (Vol. 4, pp. IV-317). IEEE.