Machine Learning Approach for Myoelectric Pattern Classification of Hand Gesture Intention

MSc Research Project
Data Analytics

Ricardo Salomao da Silva Junior
Student ID: x18147607

School of Computing
National College of Ireland

Supervisor: Paul Stynes

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ricardo Salomao da Silva Junior
Student ID:	x18147607
Programme:	Data Analytics
Year:	2019
Module:	MSc Research Project
Supervisor:	Paul Stynes
Submission Due Date:	12/08/2019
Project Title:	Machine Learning Approach for Myoelectric Pattern Classification of Hand Gesture Intention
Word Count:	4449
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th September 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Machine Learning Approach for Myoelectric Pattern Classification of Hand Gesture Intention

Ricardo Salomao da Silva Junior
x18147607

Abstract

The record of electric activity generated by muscles during contraction or relaxation is called myoelectric signal. Myoelectric patterns can be analysed for identifying motion intention by finding relationships between muscle activity and resultant body movement. The interpretation of myoelectric signals in order to identify movement intention can be very useful for the development of new human machine interface technologies and creation of intelligent robotic prosthetic devices for amputees for example. This proposed study aims to overcome the challenge of myoelectric signal classification by developing a machine learning approach for a multi-class classification method of myoelectric signals of four different types of hand movements. The selected dataset with around ten thousand of samples was obtained from kaggle. Genetic algorithms are used to select the best combination of variables and then three models are developed for comparison purposes. Results show evidence that the tuned SVM with radial function presents the best performance with 93% of correct classification rate followed by the ensemble model Random Forests with 92.3% while KNN provides the worst results in the myoelectric pattern classification, 69%.

1 Introduction

Myoelectric signals are characterized by the difference of electric potential produced by the muscle activity during contraction and relaxation during body gesture or movement. This electric signal is represented by a sinusoidal wave plotted in a graph representing the amplitude of the electric signals beneath the myoelectric sensors attached to the skin (Stewart; 2009).

Myoelectric signals can be analyzed to identify neuromuscular disorders (Khan and Singh; 2016; Subasi; 2013). Another use for this signal is the classification of movement intention. The accurate classification of myoelectric signals can lead to the development of new human machine interaction technologies or even creation of robotic prosthetic devices for disabled people (Ahsan et al.; 2009; Crawford et al.; 2005).

Classification problems in the biological area are generally a hard task to be solved. This is because individuals have particular differences in their body structure and functions. Physical and chemical composition variations from the different organisms can interfere to their responses to stimulus producing slightly different responses.

Myoelectric signal is a biological feature and, as explained in the last paragraph, has slightly differences between individuals. To find relationships between myoelectric patterns and gesture intention is a non trivial task as it can be easily affected by noise interference during the data capture and also biological, physical and chemical factors.

The aim of this research is to compare machine learning approaches for myoelectric pattern classification of hand gesture intention using data obtained through a wearable device for myoelectric data capture of upper limb motion. Four classes of movement are in study, and three models are compared.

The section 2 of this research presents a summary of related works in the area of myoelectric pattern classification with emphasis to the methodologies and techniques used by the authors. Then, the methodology with all necessary steps to conduce this research is explained in section 3. In the following section, the implementation particularities of each model are discussed. Based in evaluation criterias presented later in this document in section 5, it is provided a comparison of the implemented models giving further explanation of the specific parameters selected to achieve the results. Finally, a conclusion of this document in last section is delivered with the findings and future works to continue with this research.

2 Related Work

In this section the state of the art models for myoelectric signals classification are reviewed. The gathered preview works in the field of this research are divided by the selected algorithms as well as the features selection technique. The subsections 1, 2 and 3 reunite the related works using KNN, SVM and Random Forests respectively. The subsection 3 presents the related works with the choice of features selection technique in this research.

2.1 K Nearest Neighbour

The K Nearest Neighbour algorithm (KNN) is one of the most basic and popular machine learning techniques. This supervised algorithm consists in finding the closest k neighbours of a given data point based in a distance metric. The new data point is then classified with the class of the majority between the k neighbours (James et al.; 2014; Cover and Hart; 1967; Richard O. Duda; 2001).

Benalczar et al. (2017) presented a real time KNN based classification method for hand gesture using myoelectric data captured by MYO armband. The method proposes an individual specific gesture recognition through an initial training process using five pre-defined hand gestures: fist, wave in, wave out, open and pinch. The model achieved best performance results than the proprietary system of the MYO because of the custom made nature of the training phase. Other models and feature extraction techniques, like random forests or genetic algorithms, could be implemented in this work in order to improve the recognition accuracy.

A machine learning approach to translate sign language into text was developed by Estrada Jiménez et al. (2017). The method consists in a supervised classification model based in KNN algorithm. This approach was able to identify 61 static and dynamic gestures captured by flex, contact, and inertial sensors attached to a polyester-nylon glove. A 12 dimensional vector containing information from sensors in the glove was obtained for each class of movement by each individual before training phase. Experimental results shows classifications rate above 91%. For comparison purposes, other classification techniques like SVM or Random Forests could be implemented in order to observe the changes in the classification rate.

Using a variation of the KNN method, Khan and Singh (2016) classified myoelectric signals as neuropathic, myopathic and normal. The classifier was deeloped after extracting features from the motor unit action potentials (MUAPs) that are affected when in presence of neuromuscular disorder. Effects in the MUAPs classification could be analyzed using other models like Random Forests.

2.2 Support Vector Machine

Support Vector Machine (SVM) can give satisfactory performance with both linear and non linear problems. The versatility of this method is given by the choices of kernel functions available for different types of decision problems. Moreover, it is possible to specify personalized kernels for specific types of problems. The algorithm searches for the best hyperplane that divides the data space to obtain the best accuracy discarting outliers, if necessary. It finds the optimum decision boundary between the support vectors, being effective in high dimensional spaces and memory efficient because it only uses a set of support vectors (Cortes and Vapnik; 1995; James et al.; 2014; Richard O. Duda; 2001).

Abreu et al. (2016) proposed a machine learning approach for classification of myoelectric data of 20 letters from the brazilian signal language captured with MYO device. The method consists in training twenty binary svm capable of distinguish one letter from the rest. In case of more than one svm recognize the letter the winner classifier is the one with higher estimated probability. In order to obtain higher performance the authors tried to find the optimized svm parameters using a 10-fold cross validation technique. Results show that fine fingers gestures are hard to be perceived, also the classification is very sensitive to slightly changes in position and strength applied during the gestures. Optimization techniques, as features extraction, could be used to improve the classifier.

A real time classification method of myoelectric signals for robotic control is presented by Crawford et al. (2005). Myoelectric electrodes were placed on the forearm skin of 3 subjects for capture of seven physiologically-informed muscles used in 8 different types of hand gestures. The selected model for the 8-class classification problem was a linear SVM, and an on line training phase of about 10 minutes of duration is required. Classifications accuracy over 90% were obtained from the model, but other variations of SVM could be used to investigate the changes in the performance.

A data set containing classes of 6 hand motions collected using an EMG signal detection module (Biopac Systems Inc.) was anlysed by Rekhi et al. (2009). Like Abreu et al. (2016), one against all approach was also used in this work. A Gaussian kernel function was selected. Using wavelet packet coefficients, a new features space was after EMG data acquisition. Feeding the extracted features to the multi-class SVM 96% of accuracy was obtained of the six degrees of freedom classification problem.

2.3 Random Forests

Random forests is an ensemble method composed by a number of decision trees. This machine learning algorithm works by training a number of trees and new observations are classified by the classification of the n decision trees. The random forest classify the new observation with the most voted class obtained from the n trees (Breiman; 2001; James et al.; 2014; Richard O. Duda; 2001).

Decision trees are largely used in the artificial intelligence field to classify biomedical signals, based on that a framework for classification of myoelectric data using random forests was proposed by Gokgoz and Subasi (2015). The method classified myoelectric signals as myopathic, ALS or normal. Using a concentric needle electrodes, intramuscular myoelectric signals were collected and analysed by the authors. The random forests method over performed the decision trees C4.5 and CART.

2.4 Genetic Algorithms

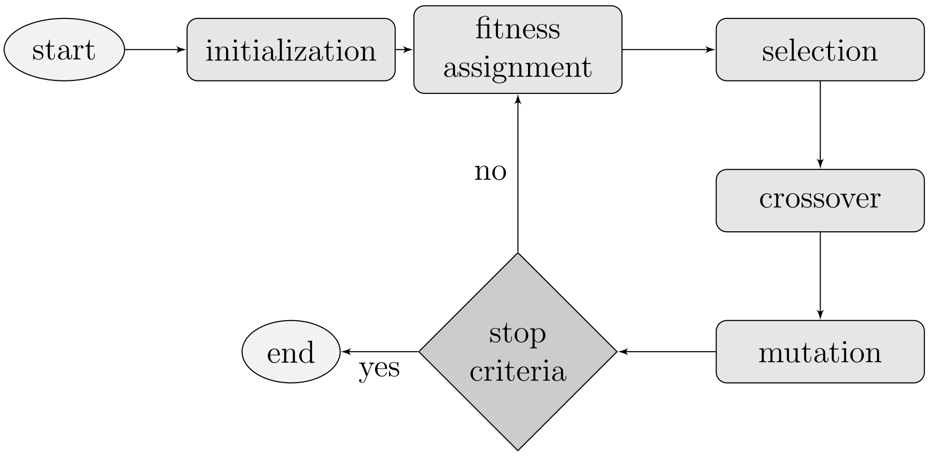

Figure 1: Genetic Algorithm Flow Chart Diagram

Genetic algorithms are used for optimisation problems and are inspired in the process of natural selection. Operations like selection, crossover and mutation are performed in order to obtain optimum solutions for a specific problem (Whitley; 1994; Holland; 1984).

Many problems in pattern recognition require a features reduction process. The classification rate is very sensitive to the choice of features. This can be treated as an optimisation problem, and genetic algorithms can give near-optimum solutions for those type of problems (Yang and Honavar; 1998).

Biomedical pattern recognition are non trivial problems. The high dimensional characteristic of biologic data can make the classification a hard problem to solve. Many alternatives are used to optimise the classification methods, one of them is selected by Huerta et al. (2006) for features reduction. The authors used genetic algorithms the most informative genes (features) using two well known cancer datasets.

2.5 Conclusion

The high dimensional behaviour of myoelectric data and noisy nature of the signal can affect the performance of traditional classifiers. Selecting the main features for the models can make a good impact in the classification rate.

Three machine learning approaches are observed in the related works on classification of myoelectric signals. For features selection of biomedical pattern classification problems the genetic algorithm appears as a established solution.

The aim of this project is to create a model for recognizing hand gesture intention using myoelectric data, weather the gesture is 0-rock, 1-scissors, 2-paper or 3-okay, for that, this work proposes a comparison of tree well known algorithms for myoelectric signal classification using genetic algorithms for features selection.

3 Methodology

In this Section the main steps of the process to execute this research are presented. This methodology is divided into five subsections: First, the data retrieval and data description are detailed. The actions for pre-processing the data set is presented in the second subsection. The third subsection is characterized by the performance of the feature selection. Then, in the fourth subsection, the machine learning approaches are modelled and finally evaluated in the last subsection.

3.1 Data

This step of the methodology explains the data acquisition process, the device used to capture the myoelectric signal and the data set description.

3.1.1 MYO Armband

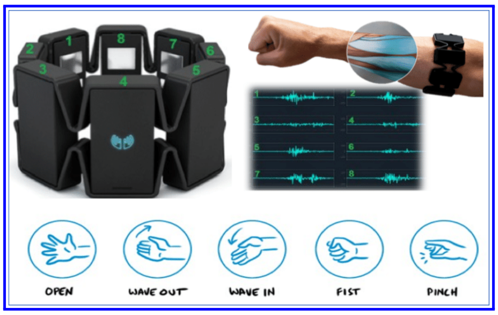

Figure 2: MYO armband (Benalczar et al.; 2017)

The data used in this work was captured using the myoelectric device MYO Armband, a wearable gesture controller that can be used as an interface control for computers. Basic actions on the computer can be controlled by this device based on the contraction and relaxation of muscles and movements of the arm.

The controller is composed by eight sensors that are placed around largest area of the arm. The sensors catch the myoelectric activity of the muscles beneath producing a record of 8 scalar measures.

Different arm gestures produce distinct myoelectric pattern records that can be used to trigger actions in the computer, like passing slides in a presentation for example.

For this research data generated by 4 different hand gestures was captured using the MYO Armband. Each row of data is composed by eight myoelectric readings of the eight sensors present in the device, generating an output csv file with 64 features and one gesture intention.

3.1.2 Data Description

One single reading of the MYO device generates an output with 8 scalar measures. Each value corresponds to the amplitude of the myoelectric activity captured by each sensor. Table 1 contains the data description of a single reading of the MYO device.

The data set used in this research is composed by rows of 40ms of myoelectric activity capture. Each row of data corresponds to a gesture record composed by 8 consecutive readings. So, instead of one single reading with 8 numbers, there are 8 consecutive readings resulting in a record with 64 features per row. This way the record can capture the entire movement avoiding eventual bad readings.

NAME	TYPE	DESCRIPTION
RS01	NUMERIC	Sensor 1 capture
RS02	NUMERIC	Sensor 2 capture
RS03	NUMERIC	Sensor 3 capture
RS04	NUMERIC	Sensor 4 capture
RS05	NUMERIC	Sensor 5 capture
RS06	NUMERIC	Sensor 6 capture
RS07	NUMERIC	Sensor 7 capture
RS08	NUMERIC	Sensor 8 capture

Table 1: MYO Single Reading Structure

Four distinct classes of gesture were captured for this study:

- ☐ Rock, labeled as 0 with 2910 samples;
- ☐ Scissors, labelled as 1 with 2903 samples;
- ☐ Paper, labelled as 2 with 2943 samples, and
- ☐ Okay, labelled as 3 with 2922 samples.

To distinguish each sensor measure in each one of the 8 readings, the columns of the data set were labelled following the role: R + "reading id [1-8]" + S + "sensor id [1-8]". So, for example, to get the capture of the 3rd sensor during the 6th reading the column id would be "R6S3".

3.1.3 Data Retrieval

The full data set with the above characteristics is available in Kaggle, the world largest data science and machine learning community.

The dataset "Classify gestures by reading muscle activity" is available through the link https://www.kaggle.com/kyr7plus/emg-4. The dataset is composed by four .csv files named 0-3, corresponding to the respective gesture records contained in each file.

The author of the dataset is "Cyber Punk Me" project [1], an open source prosthetic control system which aims to enable prosthetic devices to have multiple degrees of freedom.

[1] "Cyber Punk Me" `https://github.com/cyber-punk-me`

3.2 Pre-processing Data

The data preparation phase is a key step before the analysis. A clean and consistent data with only relevant information is very important to obtain the maximum performance of the model. The data cleaning and transformation process are detailed in this step of methodology.

3.2.1 Cleaning Data

One of the first actions that have to be done in a data set before analysis is the data cleaning. Dealing with null or missing values can be really tricky, the possible actions to solve this problem could be responsibly inputting new values or, in some cases, even removing the incomplete sample. Fortunately, using the command for checking missing values in R, the dataset obtained from kaggle has no missing values.

Next thing to do is to check and remove duplicated rows. As the observed number of rows after executing the R command for deleting duplicated rows is still the same as before it indicates that there were no duplicated values in the dataset.

Then it is time to check if the data types are appropriate in the dataset. The 64 features are measures of myoelectric sensors so they should be declared as scalar numeric. In other hand the gestures are the labels or classes for the records so they should be declared as factors.

3.2.2 Exploratory Analysis

After obtaining a consistent and clean data set it is time to perform a exploratory analysis. Finding the effects of attributes in the outcome results is a essential activity to gain insights for future analysis.

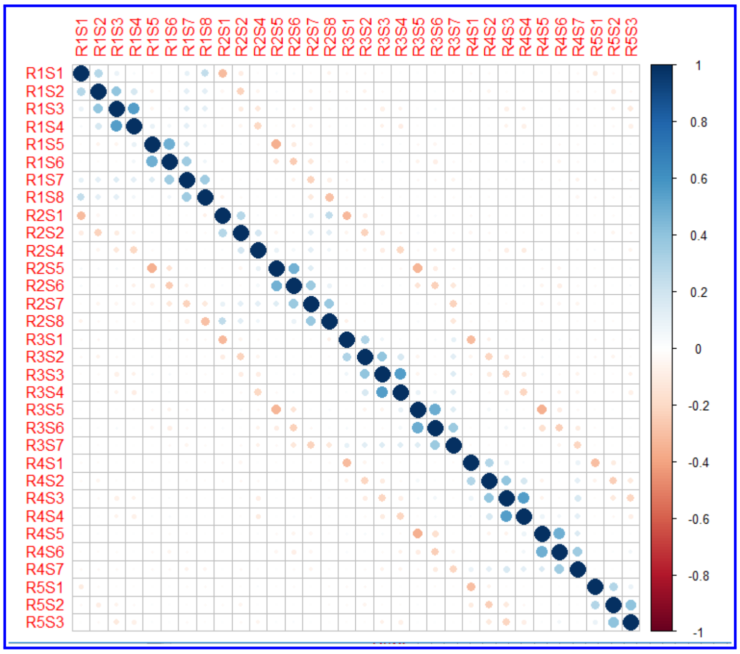

Figure 3: Correlation Plot

The correlation plot in Figure 3 shows the correlation between small subset of the independent variables. The circles in blue represent positive correlated variables, while the red ones represent the negative correlated variables. The scale of correlation is displayed in the right hand side of the graphic.

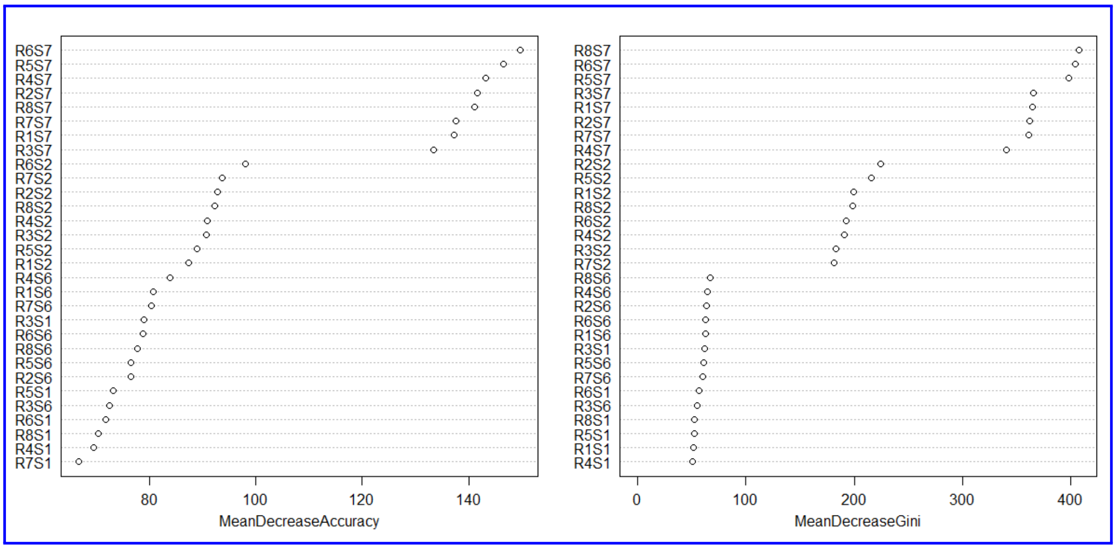

Figure 4: Variable Importance using Random Forests

Understanding the importance of each independent variable for the classification is a powerful tool to get insights to build the models. The variable importance plot in Figure 4 was built using random forests technique. It shows the capacity of prediction of some important variables in study.

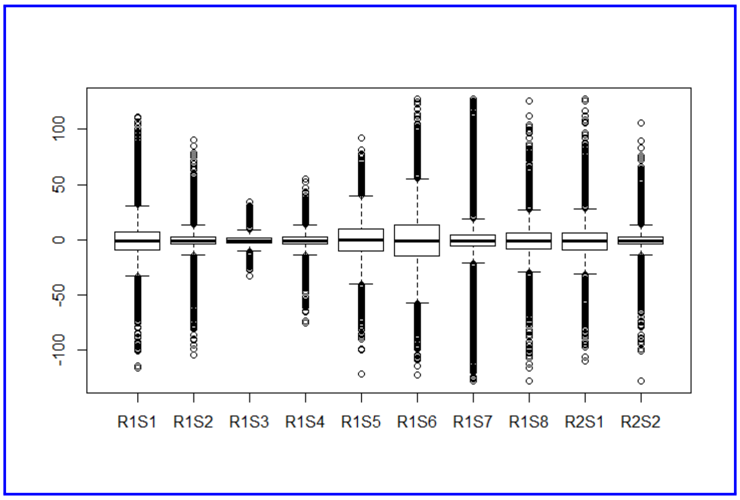

Figure 5: Checking outliers using boxplot

The boxplot presented in Figure 5 is a important visualization tool used to verify the distribution of data. Analyzing this graphic is possible to see that the data contains many outliers. This is a very important piece of information when choosing the models. The classifiers and selected parameters have to be robust solutions to deal with the high amount of outliers in the dataset.

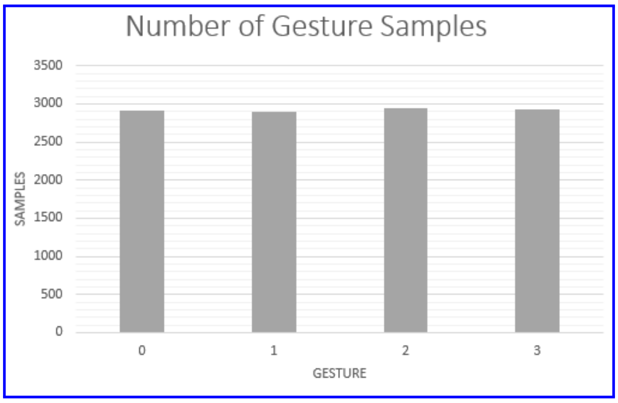

Figure 6: Gesture Samples

The exploratory data analysis shows that no inconsistencies, no major anomalies or class imbalance were detected in the data set used in this research.

3.2.3 Feature Scaling

The feature scaling process is important to minimize the effects of broad range value of features to the models performance. But, in other hand, if some of the features are almost constants except for a small noise-driven variation, this noise would be amplified greatly by normalization affecting the models performance.

The data set was submitted to a feature scaling process using the min-max normalization method, where each attribute was scaled in the range between 0 and 1. But later on it was observed that the classification rate of the models was negatively affected by this process. To avoid the negative effects of noise amplification, the featuring scaling process was skipped in this study.

After cleaning and executing the steps for pre-processing data, it is time to perform the feature reduction process.

3.3 Feature Selection

The features selection is a combinatorial optimization problem. Trying all the possibilities of subsets of features is impracticable or even impossible because the computational cost increases exponentially with the number of features.

Genetic algorithm is one of the most popular optimization algorithm. It is based on the evolutionary characteristics, were the selected solutions are combined and mutated with each other until an optimum solution is found.

The assigned fitness function used for features selection method is based in random forests with 10 folds cross validation method. The population size selected for each generation is 20 observations, and the number of total generations of the algorithm was 100.

The original subset features for the problem was reduced to 59 variables and this procedure presented no negative effects to the final classification rate.

```
ga_ctrl <- gafsControl(functions = rfGA, # Assess fitness with RF
                       method = "cv",     # 10 fold cross v alidation
                       genParallel=TRUE, # Use parallel programming
                       allowParallel = TRUE)

system.time(rf_ga3 <- gafs(x = trainX, y = y,
                           iters = 100, # 100 generations of algorithm
                           popSize = 20, # population size for each generation
                           levels = c(0:3),
                           gafsControl = ga_ctrl))
```

Figure 7: Parameters for Feature Selection with G.A.

3.4 Modeling

This step of the methodology describes the modeling process for each classifier. Based on the related works presented in section 2, three models were selected to build the classifiers. The details of each model is described in the next sessions.

3.4.1 K Nearest Neighbours

As it is widely known, this function works storing the entire training set and then classifying new elements based in a similarity measure that defines the k nearest neighbors and the class with the majority between them.

One of the key points to get the best performance of this classifier is to define the best value of k. To find the number of neighbours (k) that achieves the highest overall accuracy, the KNN model was trained and tested with different values of k. Starting from k = 1 adding one new neighbour in each iteration until k = 100, the model was trained and evaluated and the highest classification rate was achieved with k = 12 as observed in the Figure 8

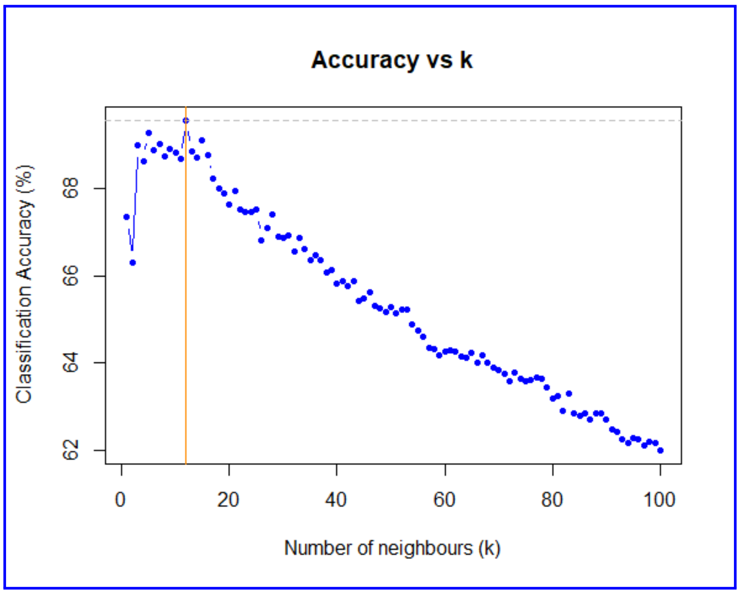

Figure 8: Finding the K which maximizes the accuracy

3.4.2 Support Vector Machine

SVM can give satisfactory performance with both linear and non linear classification problems, depending on the choice of the svm parameters and kernel function.

To find one appropriate kernel function for the problem the three most popular functions in the literature were evaluated. The findings presented in Figure 9 show that linear, polynomial and sigmoid functions produced lower classification rate than the radial basis function.

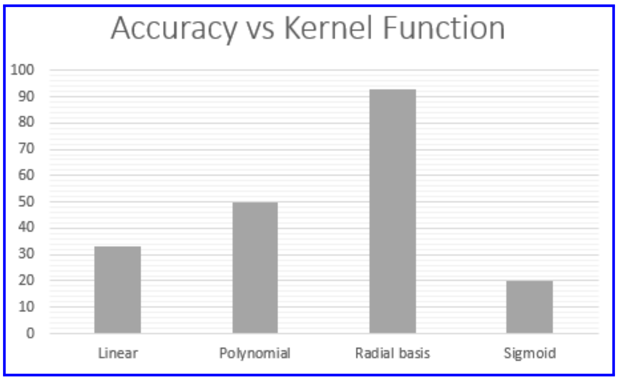

Figure 9: Finding the kernel function

Another important parameter to build the svm is the cost of constraints violation (C). To select value of cost, firstly the model's performance is evaluated with selected values for C between 1 to 50. The Highest accuracy was achieved with cost equals 7. Then, values for C were evaluated between the interval starting from 6.1 increasing 0.1 until 7.9. Using this method the best performance was achieved with cost 6.9.

Gamma is used to adjust the curvature of decision boundary of the model. Using similar approach to the cost, the best gamma observed was $\frac{9}{640}$ (0.0140625).

3.4.3 Random Forests

Decision trees can be good solutions for classification problems with non linearly separable data, but when the three grows very deep it can overfit the training set giving poor generalization and, consequently, lower performance. The implementation of random forests proposes a solution for this problem by creating an ensemble of n decision trees trained using different sections of the data reducing the variance effects.

The right choice of the number of trees in this model is essential to obtain higher classification scores. In order to find an optimized number of trees for this classifier, different numbers of trees were evaluated starting with 250 to 7000 adding 250 trees in each iteration. As shown in Figure 10, the highest accuracy was achieved with 3750 trees.

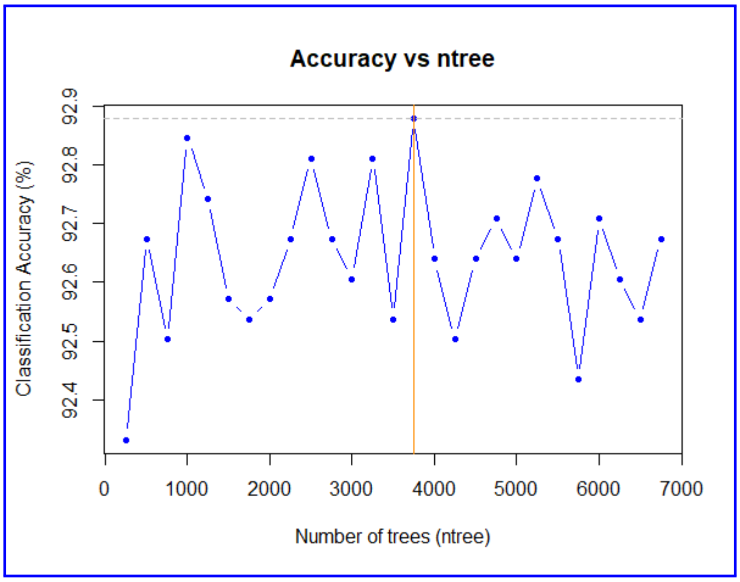

Figure 10: Finding the number of trees

3.5 Validation, Evaluation and Results

To compute the overall accuracy for each model, avoiding bias and noise interference from a given data partition, the validation method used to calculate the classification rate in this research is a 20 fold cross validation.

The entire dataset was divided into 20 mutually exclusive bins. Each bin was selected individually for test and the remaining 19 were used for training. At the end of this process the average accuracy is calculated for each model.

To calculate the other metrics a hold out validation method was performed. The original data set was split into 75% training and 25% testing.

In order to obtain a visual comparative analysis, the calculated metrics are displayed in charts comparing the results between the models.

4 Design

The work flow design of this research project is divided into six well defined steps: First, the data acquisition from Kaggle. Sencond, the pre-processing and transforming data. Then, selection of pertinent features. The fourth step is modelling the classifiers. The fifth, validation. And finally, the results presentation. The design of this work is described by the flow chart presented in Figure 11.

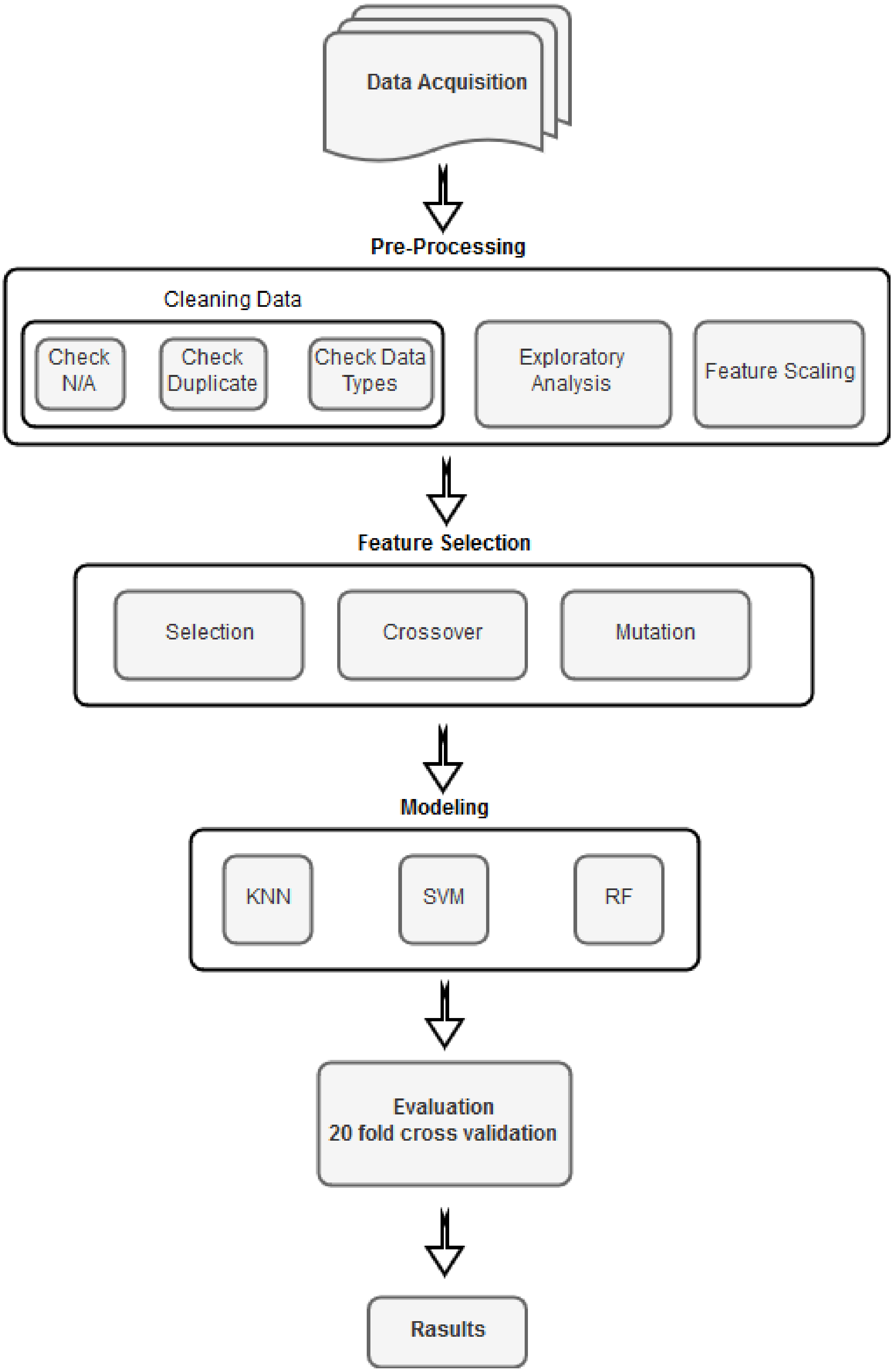

Figure 11: Design of the experiment

5 Implementation

5.1 Environment

This research was conducted using a HP Pavillon Notebook PC Model 15-au181na. The operational system was Microsoft Windows edition Windows 10 Home ©2018. The processor specifications was Intel (R) Core (TM) i5-7200U CPU @ 2.50GHz 2.71 GHz. Installed Memory (RAM) of 8.00 GB. And System type 64-bit Operating System, x64 based processor.

5.2 Programming Language

The programming language selected to implement this research was R. Developed by R Development Core Team, R is the most used programming language for development of statistical software.

This programming language contains libraries with techniques and tools that can be used for a variety of statistical analysis, machine learning problems and data visualization projects.

There are several versions of R released in the last over 20 years of existence, the version used in this research is 3.5, the most recent available until this date.

5.3 Software

The integrated development environment (IDE) used to execute the R programs created in this research is the RStudio Version 1.1.463, a professional software for R project development largely used by data analysts and statisticians.

RStudio is an open source software that contains a console and editor with real time sintax inspection for coding in R. It supports direct code execution, and has tools for workspace management, debugging and plotting graphics.

5.4 Libraries

Library	**Function**	**Implementation**
class	knn	*Used for implementing the K Nearest Neighbour in 3.4.1*
e1071	svm	*Used for implementing the Support Vector Machine in 3.4.2*
randomForest	randomForest	*Used for implementing the Random Forests in 3.4.3*
caret	confusionMatrix	*Used to create the confusion matrix for models evaluation*
corrplot	corrplot	*Used for building the correlation plots in Figure 3*
doParallel	registerDoParallel	*Used for execute the feature reduction with genetic algorithm in 3.3*
dplyr	%>%	*Used for checking duplicity like in Fig* ??

5.5 Final Implementations

5.5.1 Feature Reduction: Genetic Algorithms

```
registerDoParallel(4) # Registrer a parallel backend for train

ga_ctrl <- gafsControl(functions = rfGA, # Assess fitness with RF
                       method = "cv",    # 10 fold cross v alidation
                       genParallel=TRUE, # Use parallel programming
                       allowParallel = TRUE)

rf_ga3 <- gafs(x = trainX, y = y,
                   iters = 100, # 100 generations of algorithm
                   popSize = 20, # population size for each generation
                   levels = c(0:3),
                   gafsControl = ga_ctrl)
```

Figure 12: Genetic Algorithm Final Implementation

5.5.2 Model: K Nearest Neighbours

```
KnnPrediction <- knn(emgdata_training[,-60], emgdata_testing[,-60],
                     emgdata_training[,60],
                     k = 12, prob=TRUE)
```

Figure 13: KNN Final Implementation

5.5.3 Model: Support Vector Machine

```
svm.model = svm(emgdata[,60] ~ ., data = emgdata[,-60],
                kernel = "radial",
                cost=6.9,
                gamma=9/640)
```

Figure 14: SVM Final Implementation

5.5.4 Model: Random Forests

```
forest.model <- randomForest(emgdata[,60]~., data = emgdata[,-60],
                             ntree = 3750)
```

Figure 15: Random Forests Final Implementation

6 Evaluation

The performance criteria evaluated for this research are Overall Accuracy and Sensitivity. In the first sub section, the proportion of correct classifications of each model is analysed and compared. Then the following sub section informs the proportion of classified samples gestures correctly classified as such in each one of the machine learning approaches.

6.1 Overall Accuracy

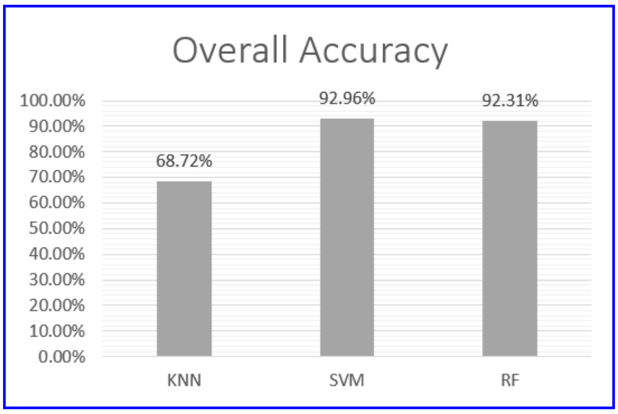

Figure 16: Overall Accuracy

The first evaluation metric to be analyzed in this work is the overall accuracy, the most used performance metric in machine learning. This is an important metric to observe how well the models perform in a general view. Here it is possible to compare the percentage of correct classifications for each model.

To avoid bias and noise effects of data partition for each model, the overall accuracy was calculated using a 20 fold cross validation technique.

The models SVM and Random Forests produced the best results for overall accuracy, over 92.96% and 92.31% respectively. In other hand, the KNN model obtained the worst classification rate, only 68.72% of the patterns were correctly classified.

6.2 Sensitivity

The second evaluation metric in analysis for the three developed models in this work is the sensitivity rate. Here it is possible to observe the rate of gestures intention from each class correctly classified as such.

Around 97% of samples from class 0-rock are correctly classified with Random Forests, a very similar result compared with SVM (96.3%). The same similarity in the results occurs with class 3-okay, when SVM provides sensitivity of 88.7% and Random Forests

88.4%. But the when it comes to the class 1-scissors the difference between SVM and Random forest is almost 10%. SVM and even KNN have greater sensitivity for class 1-scissors than Random Forests. Finally, the sensitivity from class 2-paper with random forests is more than 4% greater than SVM (90.9%).

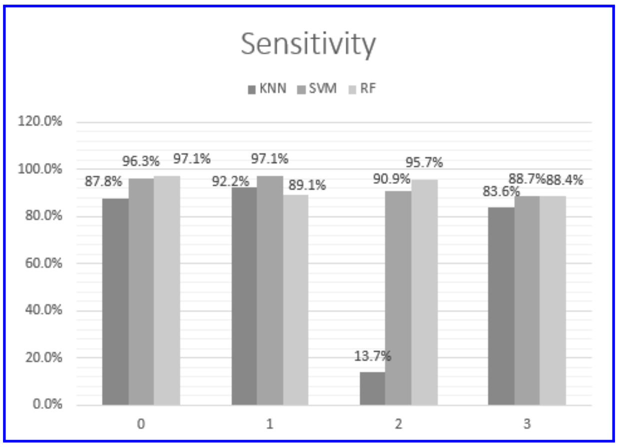

Figure 17: Sensitivity

7 Conclusion and Future Work

Hand gesture intention recognition is an important area of study for creation of robotic prosthetic devices and new human machine interaction techniques (Wang et al.; 2017; Tavakoli et al.; 2017; Mi et al.; 2016; Liu and Wang; 2018; Grif and Farcas; 2016; Rautaray and Agrawal; 2011; Wachs et al.; 2006). This work proposed a study comparison of tree machine learning approaches for myoelectric signal classification of hand gestures.

After pre-processing the data set containing samples of 4 hand gestures captured with MYO device, the genetic algorithms technique was used for features reduction of the initial space of features of the myoelectric data. The pre-processed, feature-reduced data was then classified with three algorithms: KNN, SVM and Random Forests.

The SVM approach with parameters selection provided good generalization of the problem, maximizing the performance of the classifier and minimizing the complexity of the developed model. The Random forests approach provided very close results to the SVM classifier, a likely reason for that is the fact that decision tree based models are less sensitive to outliers, one issue of the data set as seen in Figure 5. As observed in the results, KNN provided the worst performance between the three machine learning classifiers.

Support vector machines provides a powerful method for data classification. According to the results gathered in this research, the tuned SVM classifier with Genetic Algorithms feature reduction approach outperformed the Random Forests and KNN models.

One suggestion and consideration for future work would be to extend the number of classes of gestures and verify the performance of the implemented models. Another interesting future work would be to explore the findings with deep learning techniques in the myoelectric hand gesture recognition area. The use of neural networks, like Multi

Layer Perceptron for example, could be a good experiment to be conducted to after this research.

References

Abreu, J. G., Teixeira, J. M., Figueiredo, L. S. and Teichrieb, V. (2016). Evaluating sign language recognition using the myo armband, *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, pp. 64–70.

Ahsan, M. R., Ibrahimy, M. and Khalifa, O. (2009). Emg signal classification for human computer interaction a review, *European Journal of Scientific Research* **33**: 480–501.

Benalczar, M. E., Motoche, C., Zea, J. A., Jaramillo, A. G., Anchundia, C. E., Zambrano, P., Segura, M., Benalczar Palacios, F. and Prez, M. (2017). Real-time hand gesture recognition using the myo armband and muscle activity detection, *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pp. 1–6.

Breiman, L. (2001). Random forests, *Machine Learning* **45**(1): 5–32.
URL: *https://doi.org/10.1023/A:1010933404324*

Cortes, C. and Vapnik, V. (1995). Support-vector networks, *Mach. Learn.* **20**(3): 273–297.
URL: *https://doi.org/10.1023/A:1022627411411*

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* **13**(1): 21–27.

Crawford, B., Miller, K. J., Shenoy, P. and Rao, R. P. N. (2005). Real-time classification of electromyographic signals for robotic control, *AAAI*.

Estrada Jiménez, L. A., Benalcázar, M. E. and Sotomayor, N. (2017). Gesture recognition and machine learning applied to sign language translation, *in* I. Torres, J. Bustamante and D. A. Sierra (eds), *VII Latin American Congress on Biomedical Engineering CLAIB 2016, Bucaramanga, Santander, Colombia, October 26th -28th, 2016*, Springer Singapore, Singapore, pp. 233–236.

Gokgoz, E. and Subasi, A. (2015). Comparison of decision tree algorithms for emg signal classification using dwt, *Biomedical Signal Processing and Control* **18**: 138 – 144.
URL: *http://www.sciencedirect.com/science/article/pii/S1746809414002006*

Grif, H.-S. and Farcas, C. C. (2016). Mouse cursor control system based on hand gesture, *Procedia Technology* **22**: 657 – 661. 9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu Mures, Romania.
URL: *http://www.sciencedirect.com/science/article/pii/S2212017316001389*

Holland, J. H. (1984). *Genetic Algorithms and Adaptation*, Springer US, Boston, MA, pp. 317–333.
URL: *https://doi.org/10.1007/978-1-4684-8941-5*$_2$1

Huerta, E. B., Duval, B. and Hao, J.-K. (2006). A hybrid GA/SVM approach for gene selection and classification of microarray data, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 34–44.
URL: *https://doi.org/10.1007/11732242$_4$*

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2014). *An Introduction to Statistical Learning: With Applications in R*, Springer Publishing Company, Incorporated.

Khan, M. and Singh, J. K. (2016). A multi-classifier approach of EMG signal classification for diagnosis of neuromuscular disorders, *Journal of Bioengineering & Biomedical Science* **01**(s3).
URL: *https://doi.org/10.4172/2155-9538.s3-003*

Liu, H. and Wang, L. (2018). Gesture recognition for human-robot collaboration: A review, *International Journal of Industrial Ergonomics* **68**: 355 – 367.
URL: *http://www.sciencedirect.com/science/article/pii/S0169814117300690*

Mi, J., Sun, Y., Wang, Y., Deng, Z., Li, L., Zhang, J. and Xie, G. (2016). Gesture recognition based teleoperation framework of robotic fish, *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 137–142.

Rautaray, S. S. and Agrawal, A. (2011). Manipulating objects through hand gesture recognition in virtual environment, *in* D. Nagamalai, E. Renault and M. Dhanuskodi (eds), *Advances in Parallel Distributed Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 270–281.

Rekhi, N. S., Arora, A. S., Singh, S. and Singh, D. (2009). Multi-class svm classification of surface emg signal for upper limb function, *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, pp. 1–4.

Richard O. Duda, Peter E. Hart, D. G. (2001). *Pattern Classification*, second edn, John Wiley & Sons.

Stewart, G. J. (2009). *The Skeletal and Muscular Systems (Human Body: How It Works)*, Chelsea House Publications.
URL: *https://www.amazon.com/Skeletal-Muscular-Systems-Human-Body/dp/1604133651?SubscriptionId=AKIAIOBINVZYXZQZ2U3Atag=chimbori05-20linkCode=xm2camp=2025creative=165953creativeASIN=1604133651*

Subasi, A. (2013). Classification of emg signals using pso optimized svm for diagnosis of neuromuscular disorders, *Computers in Biology and Medicine*, Vol. 43, Elsevier, pp. 576–586.

Tavakoli, M., Benussi, C. and Lourenco, J. L. (2017). Single channel surface emg control of advanced prosthetic hands: A simple, low cost and efficient approach, *Expert Systems with Applications* **79**: 322 – 332.
URL: *http://www.sciencedirect.com/science/article/pii/S0957417417301574*

Wachs, J., Stern, H., Edan, Y., Gillam, M., Feied, C., Smith, M. and Handler, J. (2006). A real-time hand gesture interface for medical visualization applications, *in* A. Tiwari, R. Roy, J. Knowles, E. Avineri and K. Dahal (eds), *Applications of Soft Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 153–162.

Wang, N., Lao, K. and Zhang, X. (2017). Design and myoelectric control of an anthropomorphic prosthetic hand, *Journal of Bionic Engineering* **14**(1): 47 – 59.
URL: *http://www.sciencedirect.com/science/article/pii/S1672652916603773*

Whitley, D. (1994). A genetic algorithm tutorial, *Statistics and Computing* **4**(2): 65–85.
URL: *https://doi.org/10.1007/BF00175354*

Yang, J. and Honavar, V. (1998). *Feature Subset Selection Using a Genetic Algorithm*, Springer US, Boston, MA, pp. 117–136.
URL: *https://doi.org/10.1007/978-1-4615-5725-8*$_8$