YAŞAR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

AN IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR HAND GESTURE RECOGNITION

AYBÜKE KEÇECİ

THESIS ADVISOR: ASST. PROF.DR. İBRAHİM ZİNCİR

MASTER OF SCIENCE IN COMPUTER ENGINEERING

PRESENTATION DATE: 12.07.2019

BORNOVA/İZMİR
JULY 2019

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Jury Members:	**Signature:**
Asst.Prof.Dr. İbrahim ZİNCİR Yaşar University	....
Asst.Prof.Dr. Samsun BAŞARICI Aydın Adnan Menderes University	....
Prof.Dr. Mehmet Süleyman ÜNLÜTÜRK Yaşar University	....

Prof.Dr. Cüneyt GÜZELİŞ

Director of the Graduate School

ABSTRACT

AN IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR HAND GESTURE RECOGNITION

Keçeci, Aybüke
Msc, Computer Engineering
Advisor: Asst.Prof.Dr. İbrahim Zincir
July 2019

Motion recognition is extremely important for human-computer interaction (HCl). A hand gesture recognition system contributes an organic, creative and brought up to date version of non-verbal communication. HCI has a wide range of applications such as computer recognition of gestures, medical systems, human-robot interaction, because gesture recognition make available to people with a characteristic and instinctive computer interface. The aim of the thesis is to map user residual muscle gestures to certain actions of a prosthetic such as open/close hand or rotate the wrist. For this propose, firstly, in order to decide which features are necessary, experiments were performed by removing some features in combination. With created datasets, experiments have been done with many of the artificial neural network algorithms. Random Forest was chosen among the most successful algorithms which are Naive Bayes, BayesNet, Multilayer Perceptron, Bagging, Hoeffding Tree and Random Forest.

Key Words: hand recognition, machine learning, artificial neural networks, human-computer interaction

ÖZ

EL HAREKETLERİNİ TANIMA İÇİN MAKİNE ÖĞRENMESİ ALGORİTMALARININ UYGULANMASI

Keçeci, Aybüke
Yüksek Lisans Tezi, Bilgisayar Mühendisliği
Danışman: Yrd.Doç.Dr. İbrahim Zincir
Temmuz 2019

Hareket tanıma, insan-bilgisayar etkileşimi (HCI) için son derece önemlidir. Bir el hareketi tanıma sistemi, sözlü olmayan iletişimin doğal, yenilikçi ve modern bir yolunu sağlar. İnsan-bilgisayar etkileşimlerinde geniş bir uygulama alanına sahiptir. El hareketlerinin bilgisayarla tanınması, tıbbi sistemler, insan-bilgisayar etkileşimi gibi birçok uygulamada yaygın olarak kullanılmaktadır, çünkü el hareketi tanıma, insanlara doğal ve sezgisel bir bilgisayar ara yüzü sağlar. Bu çalışmada, kullanıcının artık kas hareketlerini; el protezinin açık / kapalı el, el bileğini döndürmek gibi belirli hareketlerini haritalanması amaçlanmıştır. Bu problemleri çözmek için öncelikle, hangi özelliklerin gerekli olduğuna karar vermek için, bazı sütunları varyasyonları halinde çıkararak deneyler gerçekleştirildi. Oluşturulan veri kümesi ile yapay sinir ağları algoritmalarından birçoğu ile deneyler yapılmış, Naive Bayes, BayesNet, Multilayer Perceptron, Bagging, Hoeffding Tree and Random Forest olan en başarılı algoritmalar arasından Random Forest seçilmiştir.

Anahtar Kelimeler: el tanıma, makina öğrenimi, yapay sinir ağları, insan-bilgisayar etkileşimi

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor İbrahim ZİNCİR for his guidance and patience during this study.

I would like to thank my teachers Mehmet Süleyman ÜNLÜTÜRK and Murat KOMESLİ to guide academically during life of my education.

I would like to thank my juryman Samsun BAŞARICI for his great criticism and help in improving my thesis.

I would like to express my enduring love to my parents and friends, who are always supportive, loving and caring to me in every possible way in my life.

Aybüke Keçeci

İzmir, 2019

TEXT OF OATH

I declare and honestly confirm that my study, titled "AN IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR HAND GESTURE RECOGNITION" and presented as a Master's Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Aybüke Keçeci

Signature

………………………………..

August 7, 2019

TABLE OF CONTENTS

ABSTRACT ii

ÖZ iii

ACKNOWLEDGEMENTS iv

TEXT OF OATH v

LIST OF FIGURES viii

LIST OF TABLES ix

ABBREVIATIONS ix

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 LITERATURE REVIEW 4

CHAPTER 3 MACHINE LEARNING 9

3.1. MACHINE LEARNING BASICS 9

3.1.1. SUPERVISED LEARNING 11

3.1.2. UNSUPERVISED LEARNING 12

3.1.3. SEMI-SUPERVISED LEARNING 12

3.1.4. REINFORCEMENT LEARNING 13

3.2. MACHINE LEARNING ALGORITHMS IMPLEMENTED 13

3.2.1. NAIVE BAYES 13

3.2.2. LOGISTIC REGRESSION 14

3.2.3. BAYESNET 14

3.2.4. SMO 14

3.2.5. BAGGING 15

3.2.6. IBK 15

3.2.7. LWL 15

3.2.8. ADABOOST 15

3.2.9. CLASSIFICATION VIA REGRESSION 16

3.2.10. LOGITBOOST 16

3.2.11. ZEROR 16

3.2.12. HOEFFDING TREE 16

3.2.13. MULTILAYER PERCEPTRON 16

3.2.14. DECISION TREE(J48) 17

3.2.15. RIPPER (JRIP) 18

3.2.16. RANDOM FOREST 18

3.2.17. RANDOM TREE 19

3.2.18. DECISION TABLE 19

3.2.19. VOTING 19

3.2.20. WEKADEEPLEARNING4J 20

CHAPTER 4 IMPLEMENTATION AND EVALUATION 21

CHAPTER 5 CONCLUSIONS AND FUTURE RESEARCH 34

REFERENCES 36

APPENDIX 1 – Sample Method for Classification 41

LIST OF FIGURES

Figure 1 A hypothetical example of Multilayer Perceptron.. 17

Figure 2 An example of Decision Tree (Lindlöf, Bräutigam, Chawade, Olsson & Olsson, 2008) 18

Figure 3 Random Forest Logic 19

Figure 4 Static gesture(a) and dynamic gesture(b) ("Sketch Hand Gesture Vector", 2019). 21

Figure 5 Myo armband with 8 sensors (Tatarian, Couceiro, Ribeiro & Faria, 2018) 23

Figure 6 Structure of Records 23

Figure 7 Example of datasets 23

LIST OF TABLES

Table 4.1 Proportion Rate of Features 26

Table 4.2 Number of Removed Features 28

Table 4.3 Results with all features and 20 algorithms 30

Table 4.4 Results with deducted 15 features and 20 algorithms 30

Table 4.5 Results with deducted 49 features and 20 algorithms 31

Table 4.6 Results with deducted 2 features and 20 algorithms 31

Table 4.7 Results with deducted 41 features and 20 algorithms 32

Table 4.8 Results with deducted 1 feature and 20 algorithms.. 32

Table 4.9 Results with deducted 35 features and 20 algorithms 32

Table 4.10 Results of Random Forest According to Versions 33

ABBREVIATIONS

ABBREVIATIONS:

EMG Electromyography

HCI Human-Computer Interaction

SMO Sequential Minimal Optimization

J48 Decision Tree

MLP Multilayer Perceptron

SVM Support Vector Machine

JRip RIPPER

kNN k-Nearest Neighbor

TA Total Accuracy

TP True Positive

TN True Negative

FP False Positive

FN False Negative

CHAPTER 1
INTRODUCTION

This thesis strives to create a system having a gesture recognition capability by taking advantage of human-computer interaction. Gestures are physical phenomena that convey information and transmit meaningful information to the other side. Throughout our lives, we deliver the messages while communicating with each other in different ways. One of them is using movements such as shaking, eye movements, head movements, or the movement of any part of our body. Gesture recognition can be used in various applications such as sign language, human-robot manipulation, and so on. Gesture and gesture recognition terms, human to computer interaction, sign language, and computer perspective have begun to be met more over time. Motion recognition systems as a new input method provide an intrinsic and comfortable interaction path between human and computer. Considering the interaction between people and people, we can easily observe that we use different movements in our communication with each other. The phenomenon underlying the use of the gesture in modern times is that we use gestures from every moment of our lives. Hand and arm movements became the perfect choice to express simple ideas, to interpret by gesture recognition system and to transform them into corresponding events. Unique hand shapes have the potential to interact with the computer system to convey meaningful information for human-computer interaction, with different hand and arm movements classified on the basis of patterns or movements, orientations or finger positions. Recognizing, classifying and interpreting various simple hand gestures and using them in a wide variety of applications with computer vision require advanced technology.

Human-Computer Interaction (HCI) is a study of different discipline areas (such as engineering, educational sciences, sociology and psychology) taking designs that put user satisfaction first and try to find the right result for the more effective and beneficial use of computer systems, software or various technologies that continue to develop day by day as a goal. The field of Human-Computer Interaction (HCI), which consists of user, task, interface and context components, aims to develop interactive technologies through design, evaluation and application processes. HCI has become a research area that concerns both the individuals who are interested in computer science and those working in the field of computer science, as computers become increasingly personalized and have gained more importance since these years.

Nielsen (2012) defines the five main features of usability as follows:

- Learnability expresses how easy it is for users to perform basic operations when they first come across the interface.
- Efficiency refers to how quickly operations are performed after learning the interface.
- Memorability refers to the ability of users to re-use the interface after a period of pause.
- Errors indicate how users make mistakes, how serious these errors are, and how easily users can get rid of them.
- Satisfaction means that the interface is used with pleasure.

The development of technologies depends on usability. Technologies with increasing usability are more suitable for development. One of these technologies is interactive technologies. It can be briefly assessed by a combination of effectiveness, satisfaction, and efficiency. Effectiveness refers to how much users can accomplish tasks using the application. Productivity shows how long it takes users to do the job. Satisfaction is a measure of users' ideas when using the application.

The scope of HCI is as follows (Basarici, 2019):

- Human Factors
- Act Theatre
- Communications
- Social Psychology
- Sociology
- Philosophy
- Management
- Organizational Psychology
- Affective Computing
- Cognitive Psychology
- Cognitive Science
- Computer Engineering
- Software Engineering

Human computer interaction is a significant especially for users with disabilities, for example screen readers or SonicFinder for those with visual impairment, text communication, gesture, captions for the hearing impaired or speech I/O, eye gaze, gesture, predictive systems for the physical impairment can be used (Basarici, 2019).

In this thesis, the dataset that was studied to recognize gestures by reading muscle movement was gathered from the platform of open dataset of Kaggle. This dataset is created to match the user's muscle movements with certain actions such as on/off or wrist rotation. The data were collected with Myo armbands, calculates the electrical activities in the muscles with the EMG (Electromyography) sensors; and at that moment it senses which hand gesture is made. The dataset record has 8 consecutive readings on 8 sensors. The first 64 features have data of muscle activity and the last feature has a number customized for gesture that occur when recording data. The motion class label (0, 1, 2, 3) has four nominal value; 0 for punch, 1 for scissor gesture, 2 for card gesture, a game like rock paper scissors and 3 is okay sign.

CHAPTER 2
LITERATURE REVIEW

Kapur, Kapur, Virji-Babul & Tzanetakis (2005) proposed a system that recognizes emotions of a human according to full body skeletal gestures using the VICON motion capture system. They used the gestural data gathered using four emotions which are sadness, joy, fear, anger of five samples with five varied classifiers; logistic regression, Naïve Bayes with a single multidimensional Gaussian distribution modeling, a decision tree, multilayer perceptron and support vector machine using the Sequential Minimal Optimization (SMO) in their experiments. The total accuracies of experiments of the research are 85.6%, 66.2%, 86.4%, 91.2% and 91.8% respectively for five classifiers mentioned earlier.

Castellano, Villalba & Camurri (2007), described a procedure for Video Content Analysis (VCA) of gesture dynamics of human to detect and identify emotions. They compared outcomes of different classifiers such as simple 1-nearest-neighbor, decision tree (J48), and hidden Naïve Bayes. As they did not work with markers in their experiments, they were able to analyze human emotion behaviors in an ecological environment. Looking at the results of the procedure they offer, their findings show that QoM is the most effective cue about discriminating emotions.

Bhattacharya, Czejdo & Perez (2012) produced a system to examine and classify gesture using Kinect sensor data. In the study conducted, at first, they classified movements from a known vocabulary of signals in an altered data stream. In the second phase of work, this time unaltered data stream is used, and random gestures are seized. They studied with J48 and Support Vector Machine algorithms and results of described techniques of machine learning are 99.97% for SVM-linear kernel and 99.32% for J48.

Palkowski & Redlarski (2016) released an article, that represent an approach for a classification system of hand gestures utilizing 2-channel surface electromyography analysis. They work on six types of hand poses and succeeded with SVM classifier and Cuckoo Search optimization algorithm for selecting correct kernel function with 98.12% total accuracy.

In a comparative study about static hand gesture recognition (Trigueiros, Ribeiro & Reis, 2012), the system is presented for recognizing a particular human posture and communicating devices. The researchers compare four machine learning algorithms which are k-Nearest Neighbor (kNN), Naïve Bayes (NB), Artificial Neural Network (ANN) and Support Vector Machine (SVM) performing on two different hand features dataset. The first dataset includes the hand angle, the mean and variance of the segmented hand grey image, the area and perimeter of the binary hand blob and the number of convexity defects. The second one contains the hand angle, the mean and variance of the segmented hand grey image as the other dataset and the 36 binary values of the orientation histogram, and the 100 binary values of the hand radial signature as different. The result of the study was 96.99% in the first dataset and 85.18% in the second.

Bartlett et al. (2005) propose a methodical system that can be implemented to the recognition of facial gesture attributes when trained with an appropriate dataset. Since AdaBoost is not only a classifier but also a feature selection technique, it is used to select the features in the comparison test. Then, for the best performance, the researchers worked with the combination of AdaBoost and SVM, which they call AdaSVM. As a result of their experiments, they achieved a success rate of 93.3% for the dataset and 90.5% for spontaneous facial gestures.

Plawiak et al. (2016) suggest a system to recognize gestures such as hand language gathering from a glove with ten sensors. The researchers worked with twenty-two hand gestures of ten people. In the designed system, three machine learning algorithms are used such as a probabilistic neural network (PNN), support vector machine, and k-nearest neighbors (kNN) algorithm with tenfold cross-validation technique and sensitivity of recognition process 97.23% for PNN, 97.36% for kNN. The system recognizes gestures with 98.32% rate of sensitivity for SVM.

In a study about hand gesture recognition (Benalcazar et al., 2017), for real-time hand action recognition, the model is proposed based on EMG using Myo armband. The gestures that are wanted to be recognized in this experiment are; pinching, punching, opening and waving in and out. K-nearest neighbors classier is utilized for recognition.

Compared to the proprietary recognition system of the Myo armband, the suggested system is more successful with 86% total accuracy.

Guo, Li & Chan (2000) represent experiments on the Cambridge ORL face database, the first experiment consists of 400 images (200 samples for training and 200 for testing) about facial expressions such as eye movements, smiling action or facial details of 40 distinct persons. The researchers compare SVM to the Nearest Center Classification (NCC) and SVM has a lower error rate with 3.0%. Second experiment has a compound data set of 1079 face images of 137 persons, which consists of five database such as the Cambridge ORL face database, the Bern database has frontal views 30 persons, the Yale database for 15 persons and for each person, randomly selected ten of its 11 frontal view images, the Harvard database and selected five persons. In the comparison between SVM and the standard eigenface method which takes the nearest center classification (NCC), the error rate of SVM was lower with 8.79%.

Wu et al. (2015) suggest a real-time American sign language recognition system using 40 most used words in daily conversations from the American Sign Language vocabulary. Four classifiers are used in evaluation which are decision tree (J48), support vector machine (libSVM), nearest neighbor and NaiveBayes. To assess the functioning of the system, in the experiment several scenarios are analyzed; self-cross validation, all cross-validation and leave one subject out of the test. Experiments results show that when the system is trained for each particular case, it made more successful predictions with 95.94% recognition rate.

The real-time pattern recognition system developed in the study conducted by Lu et al. (2016) detects and predicts six different hand movements (including hand closing; hand opening; thumb, index and middle fingers closing; thumb, index and middle fingers opening; middle, ring and little fingers closing; and middle, ring and little fingers opening) using a four-channel surface electromyography (EMG) signal and moves the exoskeleton with this prediction process. The researchers used linear Bayes classifier in their study and tested the system with two types of patients as samples which are spinal cord injury (SCI) and neurologically intact subjects. Their control accuracies are approximately 95% for SCI subjects and 98.1% for neurologically intact subjects.

Mane et al. propose a technique for identification of hand movements by classifying the single channel sEMG. They utilized a combination of wavelet analysis and ANN which the architecture of ANN consists of 3 layers feed-forward network with 2 inputs neurons, ten neurons in the hidden layer and 3 output neurons. In order to minimize mean square error (MES), the system was supported with backpropagation learning during the training phase. They worked with three hand gestures such as opening, closing, wrist extension. For the sEMG pattern discrimination system, the average rate of detection of the real-time ANN system is 93.25%.

CHAPTER 3
MACHINE LEARNING

3.1. MACHINE LEARNING BASICS

When a person encounters a stimulus, instead of evaluating it from scratch, he evaluates it with his past learnings. He tries to interpret new knowledge by resembling those in the past. This human-specific situation is also discussed in machine learning. Machine learning examines the previous examples and results and it learns how to do it again and makes generalizations about new situations. In other words, it can be said that the researches of machine learning are based on using data from the past to reveal new information and to make predictions for the future by reflecting this information on the behavior of machines. In this respect, machine learning is closely related to fields such as statistics, probability theory, human-computer interaction, data mining, pattern recognition, artificial intelligence, adaptive control, and theoretical computer science.

Alan Dix et al. (2003) describe that human-computer interaction (HCI), which aims to develop user-oriented interfaces instead of designer-oriented interfaces, is an interdisciplinary field of study related to fields such as human behavior, psychology, cognitive sciences, computer technologies and software engineering, as well as fields of ergonomics, graphic and industrial design, sociology, anthropology and educational sciences. As academic discipline HCI is about studying people interacting with technology particularly computer technology and as a design discipline is understanding how to design systems for people or about how you create that technology so did works right for people.

Manipulating and analyzing large amounts of data is not an easy process. Therefore, in some cases, it is not possible to write a program with the information at hand. This process is facilitated by machine learning, and problems such as human effort being spent on simplifying problems and adhering to certain formulas are eliminated. It is possible to find out the information that cannot be seen in the first place by machine learning. For example, by analyzing the products it sells, a company can discover the links between these products and give direction to future campaigns. In this direction, machines can make the contribution of the workforce of people, it is much easier to reach the information with the programs which are written by sharing the mental load of the human with machines.

Machine learning is the idea of creating generic algorithms that can tell you interesting things about a particular set of data without having to write code. Instead of writing code, you feed this general algorithm with data, and the algorithm creates its own logic based on this data.

The concept of machine learning dates back to 1950 when Alan Turing wrote his Computing Machinery and Intelligence (Turing, 1950), which pointed to artificial intelligence. Alan Turing put forward an experiment, which we now know as the Turing test, which is the roof of many artificial intelligence problems, to test whether it is logically possible to say whether a machine is thinking or not. (If the machine cannot determine the distinction between human beings and the computer, the machine has passed the test.) The concept of machine learning was born this way.

Machine learning is mainly a sub-branch of computer science developed in 1959 from numerical learning and model recognition studies of artificial intelligence. In 1959, Arthur Samuel describes machine learning as the “field of study that gives computers the ability to learn without being explicitly programmed” (Samuel,1959). Machine learning is a system that learns the structure and function of algorithms which can learn as a structural function and make a prediction on data. Such algorithms work by constructing a model to perform data-based estimates and decisions from sample inputs rather than strictly following static program instructions.

Once for all, machine learning has become a widespread use and with the development of machine learning tools and methods, there has been intense interest in both business and academic environments for the development of algorithms and software tools due to the business demands of the subject matter. With the development of machine learning tools and methods, ways to get better results due to the constant growth of data and the complexity of algorithms have been researched. The evaluation of machine learning methods based on different academic disciplines is mainly related to statistics and mathematics such as artificial intelligence and machine learning is done through the basic theories of statistics science. Developments in mathematics, statistics and informatics and computer science are reflected in this area. For this reason, machine learning is a field that has a wide range of application area, open to development, not only academically but also as a business area that attracts the attention of the business world.

The main concepts of machine learning are supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning.

3.1.1. SUPERVISED LEARNING

Supervised learning is based on the logic of installing the training dataset and test dataset on the system, making the necessary labeling for each data in the dataset and thus establishing a relationship between the input dataset and the output dataset. The main objective is to make effective predictions of the dataset, whose results are not known, from the classification of the known dataset (Mohri, Talwalkar & Rostamizadeh, 2012). Supervised learning problems can be grouped as regression, classification and sorting problems. Classification is the process of keeping past information in certain classes and finding out which class the new data is included. For instance, the output variable is a classification problem when verbal data is a category such as "female" or "male" or "blue" or "green". This is similar to the statistical classification. Another example is if the output variable is a numeric data, this is a regression problem. Classification Methods are also categorized as follows;

- Binary Classification is a classification with two possible results. For example: Gender classification (Male / Female)
- Multi-class Classification is a classification of more than two classes. If there are more than one different data belonging to a class, these different data are detected, and each is assigned to a single label. Ex: In an animal class there may be a cat or dog, but they cannot be in a class together but divided into classes.
- Data in the multi-label classification must be associated with more than one class. For example, an article may relate to both health, sport, and people. Difference between multi-class classification and multi-label classification; In multi-class classification, two different data cannot belong to one class together, but must

be divided into classes within itself, but in multi-label classification, one data can be associated with multiple classes.

3.1.2. UNSUPERVISED LEARNING

In unsupervised learning, the user has no intervention in the system. Only input data is given to the system but no marking is performed. The system automatically makes discoveries and tries to reveal the relationship network. The learner only receives inactive training data and pre-information for all other points not seen. It can be difficult to quantitatively evaluate a learner's performance because it is not a labeled example in this environment (Mohri, Talwalkar & Rostamizadeh, 2012). The fact that unsupervised learning separates it from supervised learning is that it is not a result and a trainer. The structure of the data is learned and modeled by algorithms. The purpose of unsupervised learning is to increase the available data to model the data into the general structure. Unsupervised learning problems can be grouped as clustering and relationship problems. Clustering refers to the grouping of data with similar properties in a dataset. Similarities in the same cluster are surplus, but there are few similarities between clusters. In other words, clustering is the process of finding clusters of similar information when the classes of historical information are not known or given. Grouping of social media accounts according to the status of the followers can be given as an example of clustering. Regression is the problems with continuous values instead of the corresponding classes.

3.1.3. SEMI-SUPERVISED LEARNING

Supervised learning and unsupervised learning may be inadequate if there are few tagged data and a large number of unlabeled data. In this case, the ideal method is to try to get information about unlabeled data from a small number of labeled data and to classify them. This method is also called semi-supervised learning. The most fundamental difference between supervised learning and semi-supervised learning is the untagged dataset. In supervised learning, a semi-supervised situation is an opposite situation while the number of data to be estimated is low and the number of labeled data is high. The learner receives a training reference consisting of both recorded and not recorded data as training data and is also available for all points not seen. Semi-supervised learning is commonly used in environments where data can be easily accessed, although it is expensive to obtain recorded data (Mohri, Talwalkar &

Rostamizadeh, 2012). As examples of semi-supervised learning, regression, classification, sorting tasks as various problems that arise in practice can be described.

3.1.4. REINFORCEMENT LEARNING

Reinforcement learning is the learning method in which a system that perceives its environment and can make decisions on its own can learn how to make the right decisions in order to achieve its goals. In other words, reinforcement learning is used in processes where there is no need for prior knowledge, i.e. without training data and where precise methods do not work. This method is frequently used in robotics, game programming, disease detection, and factory automation. There is an instructor in reinforced learning, but it cannot give much detail to the system as in supervised learning. Instead, when the learning system makes a decision, it rewards the system for situations where it is correct and punishes it for mistakes. The aim is to check whether the learning system is the target of possible situations and to remember all the true or false situations.

3.2. MACHINE LEARNING ALGORITHMS IMPLEMENTED

In this section, we summarize the machine learning algorithms used to implement the proposed dataset in this research.

3.2.1. NAIVE BAYES

Naive Bayes is a classification algorithm that classifies data with probability principles. In simple terms, a Naive Bayes classifier assumes that the existence of a particular property in a class does not depend on the presence of any other feature. Even if these properties depend on each other or the presence of other features, all of these properties independently contribute to the possibility that this fruit is an apple. Therefore, it is known as Naive.

Probability operations are performed on the trained data and the presentations to the system are provided to classify the new data according to the previous probability value. Another example is that if thousands of articles are required to be categorized according to which field they are written (medicine, technology, literature), it is necessary to learn which category belongs to the articles according to the probability values of certain words mentioned in certain articles (as the word of the word health is an article about medicine). This algorithm may be useful for a problem like this.

3.2.2. LOGISTIC REGRESSION

Logistic Regression is a classification algorithm. Classification problems are within the scope of Supervised Learning. Logistic regression predicts the probability of a result that can have only two values (i.e., it can be divided into two). The prediction is based on the use of one or several predictors (numerical and categorical). Linear regression is not appropriate for the values that can be expressed in a binary system as yes/no, exist / not, because it can estimate the value outside the range of 0 and 1. Logistic regression produces a limited logistic curve with values between 0 and 1. Logistic regression is similar to linear regression, but the curve is created using the natural logarithm of the probabilities of the target variable instead of the probability. The algorithm has been slightly customized to cope with sample weights even though the standard Logistic Regression algorithm does not handle instance weights (Xu, 2011).

3.2.3. BAYESNET

The characteristics of Bayesian networks are the statistical networks and the selection of branches that switch between nodes according to statistical decisions. Bayesian networks are directional acyclic networks and each node represents a separate variable. In addition, the sequence between these variables can also be represented by Bayes Networks.

3.2.4. SMO

Sequential Minimal Optimization Algorithm (SMO) is essentially an algorithm that uses support vector machines. The SMO Algorithm is applied to train the support vector classifier using a multinomial kernel. This application globally replaces all lost values and converts nominal attributes to binary ones. It also normalizes all attributes with predefined values. In order to overcome the problems in quadratic programming (QP), a nonlinear programming type whose objective function is a quadratic polynomial whose objective function is limited by a linear constraint and a non-negative constraint to satisfy the maximum or minimum requirement on variables, Sequential Minimal Optimization is performed.

3.2.5. BAGGING

Bagging has been proposed by L. Breiman (Breiman, 1996). It is a method aimed at retraining the basic learner by deriving new training sets from an existing training set. Instead of sampling is done, the training set is produced by random selection by putting a sample set consisting of n samples in Bagging. Each selected sample is returned to the reverse training set. Some examples are not included in the new education set, while others take place more than once.

Successful basic learners are provided with discretion by teaching learner with randomly selected training sets. Thus, ensemble success is achieved. Each base in the community is trained with educational clusters containing different examples produced in this way, and results are combined with majority voting (Brownlee, 2019).

3.2.6. IBK

IBK can be characterized as a straightforward strategy that expands the calculation of k-nearest neighbors by lessening the space used in memory and this algorithm is used for classification. Efficiently, IBK depends on likeness computations between samples like KNN and it classifies the target neighbor according to the distance from the source neighbor that constitutes a training sample (Aha, Kibler & Albert, 1991).

3.2.7. LWL

The LWL algorithm, defined as Locally Weighted Learning, is similar to other learning methods but behaves differently when classifying a new sample. This algorithm establishes a new Naïve Bayes model by looking at the cluster weight of learning samples (Frank, Hall & Pfahringer, 2002).

3.2.8. ADABOOST

AdaBoost, an ensemble of machine learning algorithms, is used for classification and regression problems. Initially, it starts with equal distribution for each sample and finds the weakest classifier according to the classification performance. It then focuses on the misclassified samples by updating the weights. As a result of a certain number of iterations, the strongest weak classifiers are brought together to form a strong classifier and increase classification success. AdaBoost uses short decision tree models, each of which has a single decision point and is often referred to as decision stump (Brownlee, 2019).

3.2.9. CLASSIFICATION VIA REGRESSION

The Classification Via Regression algorithm classifies with the regression approach and when making this classification a single regression model is created for each value of the class (Witten, Frank & Hall, 2011).

3.2.10. LOGITBOOST

The LogitBoost algorithm is used for the achievements of linear logistic regression. This classifier deals with multi-class problems by applying the regression structure as the basic learner (Witten, Frank & Hall, 2011).

3.2.11. ZEROR

ZeroR is the simplest machine learning algorithm available. The classifier first groups data according to the values of the target class or variable. It then compares the number of data accumulated in groups and decides on all the data that belong to that group (Witten, Frank & Hall, 2011).

3.2.12. HOEFFDING TREE

The Hoeffding Tree algorithm is a decision tree classifier that works effectively in large datasets by reading each instance at most one time and processing it at an appropriate time interval. In addition, the Hoeffding Tree algorithm eliminates the storage problems of traditional decision tree algorithms such as ID3, C4.5, and SLIQ. Even highly complex decision trees make it possible to create an acceptable computational cost. In each node of the decision tree, the algorithm uses the statistical value, called the Hoeffding limit, in deciding how to break the node. One of the important features of the Hoeffding Tree algorithm is that the decision tree, which is the result of the algorithm, is almost the same as the classifiers that use all samples to test each node (Domingos & Hulten, 2000).

3.2.13. MULTILAYER PERCEPTRON

Multilayer Perceptron (MLP) contains a classifier that uses backpropagation to classify the samples. This network can be manually configured, created by an algorithm, or both. The network can also be viewed and modified during the training period. The multi-layer sensor is applied to solve some difficult and different problems successfully. Very popular is the logic of the back propagation of the error. The striking features of the multi-layer perceptron are as follows:

Each neuron model in the network contains nonlinearity at the exit. An important point here is that the non-linearity is smooth transition according to the sharp-transition function used in the perceptron of Rosenblatt.

The network has one or more hidden neurons that do not belong to the outlet or inlet. These stored neurons allow the network to learn complex tasks.

In the network, each neuron is interconnected. A change in connections causes a change in synaptic connections and weights (Witten, Frank & Hall, 2011).

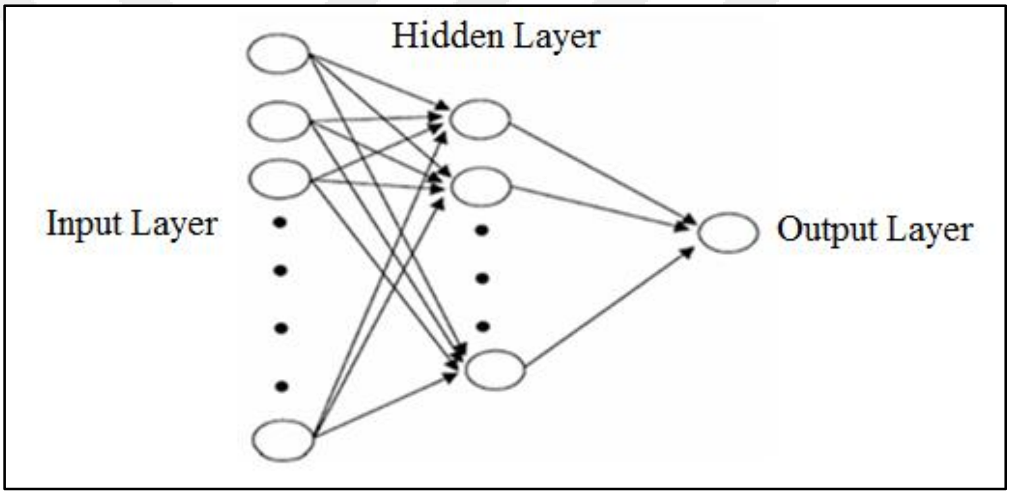

Figure 1 A hypothetical example of Multilayer Perceptron

3.2.14. DECISION TREE(J48)

The J48 alias C4.5 decision tree is the highest-rated algorithm based on algorithms such as Naive Bayes, ID3, which has the ability to automatically process data from the data. It is an iterative algorithm that divides the samples from where the gain is best. The tree structure starts with the process of dividing the subjects and choosing the best root variable of the tree and building it from top to bottom. Thus, the gain is maintained at the highest level (Gupta, 2017).

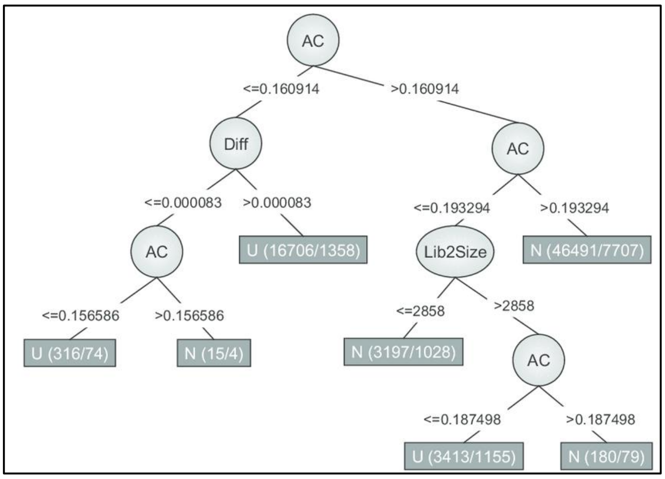

Figure 2 An example of Decision Tree (Lindlöf, Bräutigam, Chawade, Olsson & Olsson, 2008)

3.2.15. RIPPER (JRip)

The RIPPER Algorithm presented by William W. Cohen as an advanced version of the learning algorithm IREP (Incremental Reduced Error Pruning) is a rule-based classification technique using the “if… then…” rule sequence. Its aim is to develop effective rules on noisy data and compete with the C4.5 algorithm. The RIPPER rule learning algorithm covers all positive examples and consists of a set of rules in which the algorithm performs effectively on noisy datasets (Cohen, 1995).

3.2.16. RANDOM FOREST

Random Forest algorithm that constitutes most of the existing machine learning system can be used for both classification and regression problems. One of the biggest problems of decision trees is overfitting. In the interest of solving this problem, the random forest model selects many different sub-sets and trains these sub-sets randomly from both the dataset and the attribute set. With this method, a large number of decision trees are created, and each decision tree is individually estimated. In this manner, the algorithm creates its own forest. If the problem is related to the regression, the average of the estimates of the decision trees is chosen, and if the problem is related to the classification, trees with the most votes are selected.

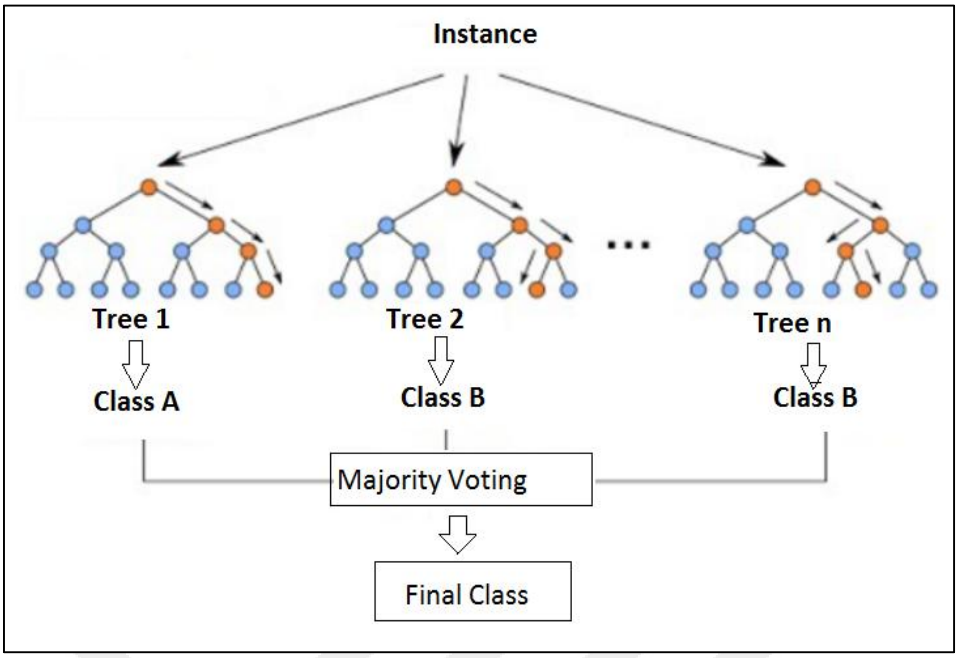

Figure 3 Random Forest Logic

3.2.17. RANDOM TREE

Random Tree is an ensemble learning algorithm and generates a random subset using the concept of bagging to develop a decision tree. The basic idea in Random Trees is to assimilate the single model trees structure with the Random Forest structure (Mishra & Ratha, 2016).

3.2.18. DECISION TABLE

The Decision Table is used for numerical estimation and consists of "if … then …" series of rules. The decision table is thought to be more understandable because it is more compressed than decision trees. This classifier builds and uses a simple decision table majority classifier (Kalmegh, 2018).

3.2.19. VOTING

In the voting technique, which is one of the basic collective learning methods, the most voted class label is assigned as the class label of the test point. In simple voting, the weights of all classifiers are equal. Decisions of each classifier on all class labels are combined and averaged. The class label with the highest rate is assigned to the test sample (Dietterich, 2000).

3.2.20. WEKADEEPLEARNING4J

WekaDeeplearning4j is designed to train and test the deep learning models implemented in Deeplearning4j in Weka, a data mining program developed in Java platform and distributed as an open source by Waikato University, which combines machine learning algorithms and data pre-processing requirements.

CHAPTER 4
IMPLEMENTATION AND EVALUATION

No matter how regular and reliable the data source is, it is necessary to pre-process the data to use it in a data mining application. The data source has been transformed into the appropriate structure for analysis used in this study. In the data preprocessing step, structuring is the most important task to clean up the data, editing the data by extracting redundant information that will not be used in the analysis. The data source used in the study was examined before the study and the analysis was erroneous, missing or unnecessary attributes and records.

To figure out what gesture recognition is, it is essential to understand how the word "gesture" is described. Gestures are representative body movements including physical motility of the fingers, hands, arms, head, face, or body with the aim to express a message or to involve in an interaction with the environment (Khan & Ibraheem, 2012).

Gesture can be divided into static like a specific posture or dynamic such as stroke phases or as in sign languages as seen in the figure below (see Figure 4).

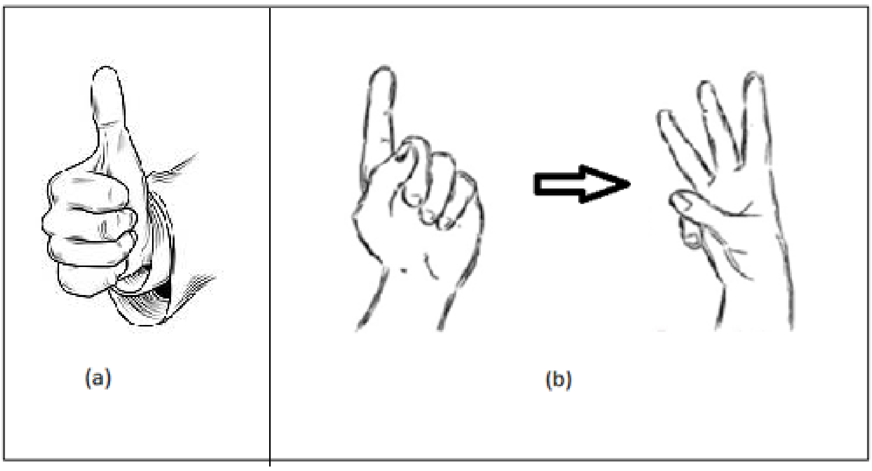

Figure 4 Static gesture(a) and dynamic gesture(b) ("Sketch Hand Gesture Vector", 2019)

Gestures can broadly be of the following types:

- hand and arm postures: hand poses recognition, sign languages, and learning applications (help children to play in virtual environments) (Nishihara, Hsu, Kaehler & Jangaard, 2017);
- head and face gestures: emotions with facial expressions, mouth movements for speaking (Moon, Lee, Ryu & Mun, 2003);
- body gestures: contribution of gait motions (Boyd & Little, 2005), etc.

Gesture recognition is called a kind of perceptual data processing interface that allows computers to perceive human movements as a command through sensors and to interpret these human movement data (Turk, 2014). The usage area of gesture recognition technology is quite wide such as the following:

- to create non-verbal communication between the operator and the system (Ho, Yamada & Umetani, 2005),
- to establish a close relationship between humans and robots (Shin, Kim & Seo, 2014),
- in the health sector and to meet military needs (Naik, Kumar & Weghorn, 2007),
- lie detection (Mitra & Acharya, 2007),

to check the level of alertness of the drivers etc.

The dataset that was studied to classify gestures by reading muscle activity was obtained from Kaggle (Kaggle, 2019), the network of data scientists, which is also a rich open dataset platform. This dataset is designed to match the user's muscle movements with certain actions of the hand prosthesis, such as on/off or wrist rotation. The data were collected with Myo armbands which is muscle activity (EMG, Electromyography) sensors connected to the arm muscles through the application (seen in Figure 5).

Figure 5 Myo armband with 8 sensors (Tatarian, Couceiro, Ribeiro & Faria, 2018)

The Myo armband has 8 sensors, each placed on the skin surface, measuring the electrical activity generated by the muscle's underneath. The dataset record has 8 consecutive readings on all 8 sensors and 11679 records. The first 64 features are raw muscle activity data and the last feature after them is a movement that occurs when recording data. The motion class label (0, 1, 2, 3) has 0 for punch gesture, 1 for scissor gesture, 2 for card gesture, a game like rock paper scissors and 3 is okay sign (seen in Figure 6). The open source dataset was used which consists of multi-classes such as; 0, 1, 2, 3. Data were recorded at 200 Hz, which means that each line is 40ms of record time. A short example of the dataset is shown as follows (seen in Figure 7):

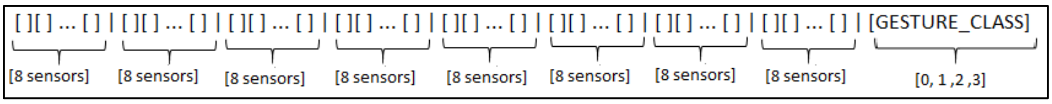

Figure 6 Structure of Records

26,4,5,8,-1,-13,-109,-66,-9,2,4,13,-18,-30,-119,17,6,-9,27,91,-26,-1,1,13,20,-62,2,2,0,-23,-1,-80,-7,-6,-27,-16,-67,-27,1,13,-8,-11,21,-28,4,8,5,-7,-59,0

69,8,0,-2,-7,2,1,2,-5,-7,-5,-5,40,65,1,-2,-2,1,-14,-4,-14,2,1,-59,-37,-4,-8,-4,-1,4,1,-1,8,2,-1,9,0,-2,-5,-7,-4,33,5,2,1,-3,-2,38,1,1

-13,-2,0,5,2,20,2,2,1,-8,2,2,-4,10,1,-1,8,-2,-3,-2,-6,14,-2,3,-15,1,-24,-7,-2,-1,-23,-3,-2,10,-2,2,-6,3,-2,2,9,11,-1,4,8,0,-1,0,1,2

1,-2,0,2,-13,3,7,12,2,-2,0,-1,8,-13,4,8,2,-4,-13,-1,3,-1,1,9,17,1,2,1,0,-11,-2,6,-7,-2,-4,7,-3,-7,-1,0,13,16,-5,4,-2,-2,-3,-26,-5,3

Figure 7 Example of datasets

The attributes of the dataset are as follows:

- reading1sensor1 - numerical
- reading1sensor2 - numerical
- reading1sensor3 - numerical
- reading1sensor4 - numerical
- reading1sensor5 - numerical
- reading1sensor6 - numerical
- reading1sensor7 - numerical
- reading1sensor8 - numerical
- reading2sensor1 - numerical
- reading2sensor2 - numerical
- reading2sensor3 - numerical
- reading2sensor4 - numerical
- reading2sensor5 - numerical
- reading2sensor6 - numerical
- reading2sensor7 - numerical
- reading2sensor8 - numerical
- reading3sensor1 - numerical
- reading3sensor2 - numerical
- reading3sensor3 - numerical
- reading3sensor4 - numerical
- reading3sensor5 - numerical
- reading3sensor6 - numerical
- reading3sensor7 - numerical
- reading3sensor8 - numerical
- reading4sensor1 - numerical
- reading4sensor2 - numerical
- reading4sensor3 - numerical
- reading4sensor4 - numerical
- reading4sensor5 - numerical
- reading4sensor6 - numerical
- reading4sensor7 - numerical
- reading4sensor8 - numerical
- reading5sensor1 - numerical
- reading5sensor2 - numerical
- reading5sensor3 - numerical
- reading5sensor4 - numerical
- reading5sensor5 - numerical
- reading5sensor6 - numerical
- reading5sensor7 - numerical
- reading5sensor8 - numerical
- reading6sensor1 - numerical
- reading6sensor2 - numerical
- reading6sensor3 - numerical
- reading6sensor4 - numerical
- reading6sensor5 - numerical
- reading6sensor6 - numerical
- reading6sensor7 - numerical
- reading6sensor8 - numerical
- reading7sensor1 - numerical
- reading7sensor2 - numerical
- reading7sensor3 - numerical
- reading7sensor4 - numerical
- reading7sensor5 - numerical
- reading7sensor6 - numerical
- reading7sensor7 - numerical
- reading7sensor8 - numerical
- reading8sensor1 - numerical
- reading8sensor2 - numerical
- reading8sensor3 - numerical
- reading8sensor4 - numerical
- reading8sensor5 - numerical
- reading8sensor6 - numerical
- reading8sensor7 - numerical
- reading8sensor8 - numerical
- GESTURE_CLASS {0, 1, 2, 3}

The experiment in this thesis was carried out with the algorithms written on the NetBeans platform using WEKA's background structure and the classifiers explained in Chapter 4.

NetBeans is a Java development environment (IDE) developed by Oracle and distributed free of charge. It is especially preferred because of the convenience it provides in user interface design. Working with the Netbeans IDE, Java, C / C ++, PHP and HTML5 language includes all the tools needed for desktop, enterprise, Web and Mobile applications ("NetBeans IDE 8.2 Release Information", n.d., "NetBeans Platform Description", n.d.). WEKA is a visual combination of algorithms and tools for data analysis and estimator modeling, which is used in academic research, educational and industrial application areas. The main advantages of the software are that it has extensive data preprocessing and modeling techniques is easy to use thanks to its graphical user interface and is portable to any platform where it is implemented with the Java programming language (Witten et al., 2011).

All the algorithms are evaluated using their WEKA implementations with default parameter settings in the official site of WEKA. The reason using the default parameter settings was to understand how well they would perform without any a priori information. In order to determine which features will be used in the classification in the experiments, it is easy to observe which features create more nodes on the trees created by the J48 and JRIP algorithms or which are used more in the created model. It should also be noted here that during the evaluations, first, the number of uses of the features used of the data set in the nodes of trees created by the J48 and JRip algorithms were determined (seen in Table 4.1).

Table 4.1 Proportion Rate of Features

<= 10		
v1-j48	**v2-jrip**	
r2s8	r1s1	r5s1
r3s6	r1s3	r5s3
r3s8	r1s4	r5s4
r4s3	r1s5	r5s5
r4s8	r1s6	r5s6
r5s1	r1s8	r5s8
r5s8	r2s1	r6s1
r6s3	r2s3	r6s2
r6s5	r2s4	r6s3
r6s8	r2s5	r6s4
r7s3	r2s6	r6s5
r8s1	r2s8	r6s6
r8s5	r3s1	r6s8
r8s8	r3s3	r7s1
r3s3	r3s4	r7s3
	r3s5	r7s4
	r3s6	r7s5
	r3s8	r7s6
	r4s1	r7s8
	r4s3	r8s1
	r4s4	r8s3
	r4s5	r8s4
	r4s6	r8s5
	r4s8	r8s6
		r8s8

<= 5		
v3-j48	**v4-jrip**	
r6s3	r1s1	r5s1
r6s8	r1s3	r5s3
	r1s4	r5s4
	r1s5	r5s5
	r1s6	r5s8
	r1s8	r6s1
	r2s1	r6s3
	r2s3	r6s4
	r2s4	r6s5
	r2s5	r6s8
	r2s8	r7s1
	r3s1	r7s3
	r3s3	r7s4
	r3s4	r7s5
	r3s5	r7s8
	r3s8	r8s1
	r4s1	r8s3
	r4s3	r8s4
	r4s4	r8s5
	r4s5	r8s8
	r4s8	

<= 3		
v5-j48	**v6-jrip**	
r6s3	r1s1	r5s1
	r1s3	r5s3
	r1s4	r5s4
	r1s5	r5s5
	r1s8	r5s8
	r2s1	r6s3
	r2s3	r6s4
	r2s4	r6s5
	r2s8	r6s8
	r3s1	r7s1
	r3s4	r7s3
	r3s5	r7s4
	r3s8	r7s8
	r4s3	r8s1
	r4s4	r8s3
	r4s5	r8s4
	r4s8	r8s5
		r8s8

In this study, experimentally, according to the determined numbers, the features in the data set in which their numbers in the nodes of the trees are respectively equal to or less than ten, equal to or less than five and equal to or less than three have been removed to get more successful results. Then, seven approaches are used; according to proportion rate of features, experiments were performed with features removed from 15 out of 64 features for version 1, 49 out of 64 features for version 2, 2 out of the 64 features for version 3, 41 out of the 64 features for version 4, 1 out of the 64 features for version 5, and 35 out of the 64 features for version 6 respectively (seen in Table 4.2). In doing so, the goal is to understand whether these attributes could bias/effect the results of the experiment in one way or another. Given the dataset, to minimize any statistical biases, 10-fold cross-validation was employed to evaluate the 20 different machine learning algorithms used for multi-class classification and these are listed as follows:

- Naive Bayes
- Logistic Regression
- BayesNet
- SMO
- Bagging
- IbK
- LWL
- AdaBoost
- Classification Via Regression
- LogitBoost
- ZeroR
- Hoeffding Tree
- Multilayer Perceptron
- Decision Tree(J48)
- RIPPER (JRip)
- Random Forest
- Random Tree
- Decision Table
- Voting
- Wekadeeplearning4j

Table 4.2 Number of Removed Features

Versions	Number of Removed Features	Algorithms
V0	All features	-
V1	15	J48 <= 10
V2	49	JRIP <= 10
V3	2	J48 <= 5
V4	41	JRIP <= 5
V5	1	J48 <= 3
V6	35	JRIP <= 3

In the following tables (Table 4.3, Table 4.4, Table 4.5, Table 4.6, Table 4.7, Table 4.8), the 10-fold cross-validation results for balancing total accuracy, true positive, true negative, ROC, precision, and sensitivity of the proposed machine learning based system are reported.

- Total accuracy (TA): is the ratio of accurate estimation to all estimations in the system as in (1).

$Accuracy = {TP + TN}/{TN + TP + FN + FP}$	(Joshi, 2016) (1)

- True positive (TP): is a positive estimation of a condition that actually exists in the prediction process.
- True negative (TN): is the prediction of a situation that actually exists as negative in the prediction process.
- ROC: ROC curve and provides a complete sensitivity/specificity report. In the ROC curve, the true positive ratio (Sensitivity) is plotted in the function of the false positive ratio (100-Specificity) for different cut-off points of a parameter. The degrees of ROC according to the traditional academic score system are listed below (Mehdi, Bashardoost & Ahmadi, 2011).
    - .90-1 = excellent (A)
    - .80-.90 = good (B)
    - 70-.80 = fair (C)
    - .60-.70 = poor (D)

- .50-.60 = fail (F)

- Precision: is the case that indicates success in a situation that is estimated to be positive as in (2).

$Precision = {}^{TP}/_{(TP + FP)}$	(Joshi, 2016) (2)

- Recall: refers to calculate the distribution of certain positives that are correctly classified as in (3).

$Recall = {}^{TP}/_{TP + FN}$	(Joshi, 2016) (3)

Between the 20 algorithms used in this thesis, Random Forest algorithm reaches the best results in third version of dataset with 92.8926% total accuracy and 0.9735 for class 0, 0.9328 for class 1, 0.9419 for class 2, 0.8676 for class 3 of TP (seen in Table 4.9).

Table 4.3 Results with all features and 20 algorithms

emg_4_v0

	NaiveBayes				BayesNet				Logistic				Multilayer Perceptron				SMO			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	88.4912%				90.4008%				34.7748%				88.8765%				32.3514%			
TP	0.9175	0.9528	0.9045	0.7652	0.9375	0.9476	0.9185	0.8128	0.4567	0.3142	0.2881	0.3326	0.9299	0.9070	0.8607	0.8580	0.2385	0.4364	0.3136	0.3060
TN	0.9722	0.9390	0.9842	0.9512	0.9723	0.9654	0.9747	0.9597	0.8461	0.7875	0.7351	0.7614	0.9740	0.9700	0.9665	0.9412	0.9970	0.6473	0.7025	0.7510
ROC	0.9871	0.9883	0.9886	0.9482	0.9941	0.9924	0.9888	0.9604	0.5916	0.5492	0.5207	0.5461	0.9831	0.9862	0.9643	0.9513	0.6474	0.5390	0.5127	0.5210
Precision	0.9163	0.8379	0.9507	0.8397	0.9182	0.9005	0.9244	0.8706	0.4963	0.3284	0.2682	0.3175	0.9223	0.9092	0.8963	0.8296	0.9639	0.2905	0.2621	0.2908
RECALL	0.9175	0.9528	0.9045	0.7652	0.9375	0.9476	0.9185	0.8128	0.4567	0.3142	0.2881	0.3326	0.9299	0.9070	0.8607	0.8580	0.2385	0.4364	0.3136	0.3060

	Bagging				Ibk				LWL				AdaBoostM1				ClassificationViaRegression			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	89.2704%				65.7133%				44.7594%				34.0298%				82.3001%			
TP	0.9543	0.8756	0.8977	0.8433	0.8041	0.8446	0.2518	0.7327	0.7997	0.9990	0	0	0.3698	0.8987	0.0982	0	0.8835	0.8801	0.7853	0.7440
TN	0.9790	0.9736	0.9603	0.9440	0.9921	0.7874	0.9639	0.7997	0.9479	0.3169	1	1	0.9784	0.2316	0.9116	1	0.9629	0.9357	0.9365	0.9288
ROC	0.9957	0.9835	0.9794	0.9637	0.8980	0.8138	0.6073	0.7661	0.8944	0.7196	0.7384	0.6394	0.6785	0.5746	0.5537	0.5441	0.9715	0.9623	0.9336	0.9034
Precision	0.9379	0.9164	0.8839	0.8341	0.9714	0.5679	0.7017	0.5497	0.8358	0.3261	0	0	0.8506	0.2790	0.2724	0	0.8878	0.8192	0.8064	0.7773
RECALL	0.9543	0.8756	0.8977	0.8433	0.8041	0.8446	0.2518	0.7327	0.7997	0.9990	0	0	0.3698	0.8987	0.0982	0	0.8835	0.8801	0.7853	0.7440

	LogitBoost				Jrip				ZeroR				HoeffdingTree				J48			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	81.7777%				82.8995%				25.2012%				88.5063%				80.5104%			
TP	0.8605	0.8829	0.7880	0.7406	0.9402	0.8378	0.7632	0.7758	0	0	1	0	0.9179	0.9501	0.9086	0.7642	0.8756	0.8319	0.7866	0.7269
TN	0.9745	0.9161	0.9459	0.9206	0.9774	0.9325	0.9376	0.9244	1	1	0	1	0.9722	0.9394	0.9843	0.9510	0.9602	0.9341	0.9335	0.9123
ROC	0.9815	0.9660	0.9540	0.9213	0.9739	0.9279	0.9034	0.8919	0.4999	0.4995	0.4995	0.4996	0.9871	0.9878	0.9884	0.9478	0.9260	0.8863	0.8441	0.8003
Precision	0.9179	0.7769	0.8306	0.7569	0.9325	0.8042	0.8047	0.7740	0	0	0.2520	0	0.9163	0.8383	0.9513	0.8388	0.8795	0.8069	0.7994	0.7344
RECALL	0.8605	0.8829	0.7880	0.7406	0.9402	0.8378	0.7632	0.7758	0	0	1	0	0.9179	0.9501	0.9086	0.7642	0.8756	0.8319	0.7866	0.7269

	RandomForest				RandomTree				DecisionTable				WekaDeepLearning4J				Vote			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	92.5758%				75.1755%				66.4326%				33.8842%				25.2012%			
TP	0.9732	0.9263	0.9385	0.8652	0.8137	0.7899	0.7210	0.6831	0.7196	0.8312	0.5647	0.5438	0.427	0.300	0.325	0.304	0	0	1	0
TN	0.9813	0.9798	0.9721	0.9678	0.9494	0.9271	0.9098	0.8827	0.9235	0.8581	0.8948	0.8761	0.1678	0.2122	0.2606	0.2412	1	1	0	1
ROC	0.9983	0.9934	0.9880	0.9833	0.8816	0.8585	0.8154	0.7829	0.9249	0.9037	0.8291	0.7989	0.584	0.555	0.530	0.546	0.4999	0.4995	0.4995	0.4996
Precision	0.9453	0.9382	0.9188	0.8996	0.8421	0.7818	0.7292	0.6603	0.7573	0.6597	0.6439	0.5942	0.458	0.318	0.296	0.296	0	0	0.2520	0
RECALL	0.9732	0.9263	0.9385	0.8652	0.8137	0.7899	0.7210	0.6831	0.7196	0.8312	0.5647	0.5438	0.427	0.300	0.325	0.304	0	0	1	0

Table 4.4 Results with deducted 15 features and 20 algorithms

emg_4_v1

	NaiveBayes				BayesNet				Logistic				Multilayer Perceptron				SMO			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	88.4227				90.8717				34.0726				87.9945				31.4951			
TP	0.9230	0.9459	0.9096	0.7587	0.9495	0.9501	0.9215	0.8142	0.4646	0.2935	0.2915	0.3138	0.9275	0.8984	0.8760	0.8183	0.2330	0.4034	0.3337	0.2899
TN	0.9718	0.9377	0.9850	0.9512	0.9742	0.9632	0.9779	0.9630	0.8539	0.7863	0.7309	0.7497	0.9733	0.9594	0.9630	0.9442	0.9975	0.6598	0.6853	0.7437
ROC	0.9900	0.9872	0.9890	0.9507	0.9954	0.9919	0.9900	0.9645	0.5879	0.5376	0.5164	0.5397	0.9818	0.9814	0.9627	0.9444	0.6419	0.5281	0.5107	0.5178
Precision	0.9158	0.8339	0.9533	0.8385	0.9244	0.8952	0.9336	0.8801	0.5135	0.3124	0.2674	0.2950	0.9202	0.8799	0.8887	0.8302	0.9686	0.2818	0.2632	0.2740
RECALL	0.9230	0.9459	0.9096	0.7587	0.9495	0.9501	0.9215	0.8142	0.4646	0.2935	0.2915	0.3138	0.9275	0.8984	0.8760	0.8183	0.2330	0.4034	0.3337	0.2899

	Bagging				Ibk				LWL				AdaBoostM1				ClassificationViaRegression			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	89.2533				68.0253				45.3502				34.0298				82.2658			
TP	0.9546	0.8708	0.9052	0.8395	0.8361	0.8350	0.3082	0.7461	0.8234	0.9990	0	0	0.3698	0.8987	0.0982	0.0000	0.8900	0.8787	0.7808	0.7420
TN	0.9786	0.9738	0.9604	0.9439	0.9933	0.8058	0.9700	0.8048	0.9440	0.3287	1	1	0.9784	0.2316	0.9116	1.0000	0.9621	0.9360	0.9392	0.9262
ROC	0.9960	0.9836	0.9796	0.9628	0.9172	0.8207	0.6411	0.7762	0.9076	0.7388	0.7372	0.6558	0.6785	0.5746	0.5537	0.5441	0.9721	0.9623	0.9345	0.9003
Precision	0.9366	0.9166	0.8850	0.8332	0.9763	0.5872	0.7759	0.5606	0.8299	0.3299	0	0	0.8506	0.2790	0.2724	0.0000	0.8864	0.8195	0.8123	0.7704
RECALL	0.9546	0.8708	0.9052	0.8395	0.8361	0.8350	0.3082	0.7461	0.8234	0.9990	0	0	0.3698	0.8987	0.0982	0.0000	0.8900	0.8787	0.7808	0.7420

	LogitBoost				Jrip				ZeroR				HoeffdingTree				J48			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	81.8120				82.4114				25.2012				88.4398				81.3067			
TP	0.8588	0.8812	0.7907	0.7426	0.9385	0.8319	0.7693	0.7577	0	0	1	0	0.9244	0.9463	0.9110	0.7563	0.8784	0.8367	0.8012	0.7365
TN	0.9743	0.9169	0.9456	0.9206	0.9757	0.9323	0.9353	0.9221	1	1	0	1	0.9713	0.9374	0.9850	0.9523	0.9624	0.9365	0.9376	0.9142
ROC	0.9812	0.9663	0.9543	0.9220	0.9766	0.9253	0.9104	0.8849	0.4999	0.4995	0.4995	0.4996	0.9899	0.9868	0.9889	0.9503	0.9258	0.8888	0.8572	0.8071
Precision	0.9174	0.7782	0.8305	0.7574	0.9276	0.8026	0.8003	0.7645	0	0	0.2520	0	0.9143	0.8334	0.9534	0.8409	0.8857	0.8135	0.8123	0.7413
RECALL	0.8588	0.8812	0.7907	0.7426	0.9385	0.8319	0.7693	0.7577	0	0	1	0	0.9244	0.9463	0.9110	0.7563	0.8784	0.8367	0.8012	0.7365

	RandomForest				RandomTree				DecisionTable				WekaDeepLearning4J				Vote			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	92.6357				75.9633				66.4326				33.7044				25.2012			
TP	0.9729	0.9232	0.9392	0.8703	0.8474	0.7888	0.7217	0.6814	0.7196	0.8312	0.5647	0.5438	0.4370	0.3030	0.3030	0.3050	0	0	1	0
TN	0.9822	0.9810	0.9725	0.9661	0.9514	0.9274	0.9139	0.8867	0.9235	0.8581	0.8948	0.8761	0.8174	0.7731	0.7612	0.7628	1	1	0	1
ROC	0.9982	0.9926	0.9884	0.9826	0.8994	0.8581	0.8178	0.7840	0.9249	0.9037	0.8291	0.7989	0.5810	0.5500	0.5240	0.5420	0.4999	0.4995	0.4995	0.4996
Precision	0.9478	0.9413	0.9201	0.8954	0.8527	0.7824	0.7385	0.6674	0.7573	0.6597	0.6439	0.5942	0.4700	0.3190	0.2890	0.2860	0	0	0.2520	0
RECALL	0.9729	0.9232	0.9392	0.8703	0.8474	0.7888	0.7217	0.6814	0.7196	0.8312	0.5647	0.5438	0.4370	0.3030	0.3030	0.3050	0	0	1.0000	0

Table 4.5 Results with deducted 49 features and 20 algorithms

emg_4_v2	NaiveBayes				BayesNet				Logistic				Multilayer Perceptron				SMO			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	88.6111				89.2533				30.3220				81.4866				31.7263			
TP	0.9096	0.9507	0.8964	0.7882	0.9144	0.9390	0.9117	0.8053	0.5024	0.0558	0.4288	0.2242	0.8436	0.8977	0.8525	0.6660	0.2323	0.4068	0.4234	0.2060
TN	0.9760	0.9428	0.9813	0.9480	0.9765	0.9573	0.9718	0.9511	0.8471	0.8883	0.5510	0.7837	0.9624	0.9096	0.9438	0.9374	0.9986	0.6838	0.5991	0.8077
ROC	0.9851	0.9860	0.9841	0.9505	0.9894	0.9876	0.9862	0.9564	0.5902	0.5371	0.5160	0.5256	0.9683	0.9524	0.9382	0.8893	0.6297	0.4813	0.5051	0.5179
Precision	0.9265	0.8461	0.9418	0.8350	0.9281	0.8791	0.9160	0.8461	0.5216	0.1419	0.2434	0.2570	0.8815	0.7667	0.8363	0.7803	0.9826	0.2985	0.2624	0.2633
RECALL	0.9096	0.9507	0.8964	0.7882	0.9144	0.9390	0.9117	0.8053	0.5024	0.0558	0.4288	0.2242	0.8436	0.8977	0.8525	0.6660	0.2323	0.4068	0.4234	0.2060

emg_4_v2	Bagging				Ibk				LWL				AdaBoostM1				ClassificationViaRegression			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	88.2600				83.0879				47.7993				34.0298				82.1973			
TP	0.9210	0.8843	0.9038	0.8214	0.8313	0.9270	0.7781	0.7882	0.9237	0.9969	0	0	0.3698	0.8987	0.0982	0.0000	0.8364	0.8694	0.8226	0.7598
TN	0.9747	0.9698	0.9611	0.9379	0.9852	0.9018	0.9731	0.9146	0.8847	0.4205	1	1	0.9784	0.2316	0.9116	1.0000	0.9693	0.9378	0.9447	0.9108
ROC	0.9892	0.9820	0.9792	0.9526	0.9111	0.9118	0.8743	0.8500	0.9598	0.8914	0.8827	0.6198	0.6785	0.5746	0.5537	0.5441	0.9660	0.9637	0.9462	0.9101
Precision	0.9235	0.9064	0.8867	0.8152	0.9490	0.7574	0.9069	0.7548	0.7267	0.3627	0	0	0.8506	0.2790	0.2724	0.0000	0.9005	0.8221	0.8337	0.7398
RECALL	0.9210	0.8843	0.9038	0.8214	0.8313	0.9270	0.7781	0.7882	0.9237	0.9969	0	0	0.3698	0.8987	0.0982	0.0000	0.8364	0.8694	0.8226	0.7598

emg_4_v2	LogitBoost				Jrip				ZeroR				HoeffdingTree				J48			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	81.5122				82.7025				25.2012				88.6282				81.7863			
TP	0.8395	0.9104	0.7931	0.7183	0.9052	0.8426	0.8158	0.7450	0	0	1	0	0.9113	0.9511	0.8987	0.7844	0.8704	0.8581	0.8090	0.7344
TN	0.9711	0.9137	0.9457	0.9229	0.9748	0.9397	0.9293	0.9255	1	1	0	1	0.9749	0.9426	0.9817	0.9493	0.9611	0.9432	0.9433	0.9094
ROC	0.9765	0.9685	0.9575	0.9151	0.9648	0.9325	0.9189	0.8818	0.4999	0.4995	0.4995	0.4996	0.9851	0.9858	0.9839	0.9504	0.9313	0.9102	0.8741	0.8197
Precision	0.9062	0.7774	0.8312	0.7567	0.9226	0.8222	0.7953	0.7695	0	0	0.2520	0	0.9234	0.8456	0.9430	0.8377	0.8814	0.8334	0.8279	0.7302
RECALL	0.8395	0.9104	0.7931	0.7183	0.9052	0.8426	0.8158	0.7450	0	0	1	0	0.9113	0.9511	0.8987	0.7844	0.8704	0.8581	0.8090	0.7344

emg_4_v2	RandomForest				RandomTree				DecisionTable				WekaDeepLearning4J				Vote			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	90.1010				79.1231				66.4326				33.8757				25.2012			
TP	0.9354	0.9008	0.9276	0.8402	0.8533	0.8209	0.7727	0.7187	0.7196	0.8312	0.5647	0.5438	0.4840	0.1810	0.3820	0.3080	0	0	1	0
TN	0.9768	0.9736	0.9685	0.9491	0.9578	0.9393	0.9323	0.8922	0.9235	0.8581	0.8948	0.8761	0.8320	0.7594	0.7693	0.7570	1	1	0	1
ROC	0.9919	0.9869	0.9853	0.9657	0.9055	0.8801	0.8525	0.8054	0.9249	0.9037	0.8291	0.7989	0.5850	0.5470	0.5260	0.5310	0.4999	0.4995	0.4995	0.4996
Precision	0.9306	0.9185	0.9085	0.8463	0.8703	0.8172	0.7937	0.6899	0.7573	0.6597	0.6439	0.5942	0.5150	0.2920	0.2960	0.2680	0	0	0.2520	0
RECALL	0.9354	0.9008	0.9276	0.8402	0.8533	0.8209	0.7727	0.7187	0.7196	0.8312	0.5647	0.5438	0.4840	0.1810	0.3820	0.3080	0	0	1.0000	0

Table 4.6 Results with deducted 2 features and 20 algorithms

emg_4_v3	NaiveBayes				BayesNet				Logistic				Multilayer Perceptron				SMO			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	88.6025				90.7176				34.6977				88.4398				32.5141			
TP	0.9182	0.9535	0.9062	0.7666	0.9426	0.9507	0.9191	0.8166	0.4529	0.3124	0.2875	0.3357	0.9320	0.8949	0.8641	0.8470	0.2392	0.4306	0.3323	0.2988
TN	0.9726	0.9395	0.9839	0.9521	0.9723	0.9668	0.9755	0.9616	0.8479	0.7872	0.7333	0.7607	0.9762	0.9647	0.9642	0.9408	0.9974	0.6588	0.6871	0.7566
ROC	0.9877	0.9884	0.9886	0.9492	0.9944	0.9926	0.9890	0.9615	0.5883	0.5492	0.5197	0.5471	0.9851	0.9829	0.9636	0.9500	0.6478	0.5388	0.5172	0.5213
Precision	0.9176	0.8390	0.9498	0.8424	0.9186	0.9046	0.9267	0.8766	0.4970	0.3270	0.2664	0.3189	0.9284	0.8934	0.8904	0.8269	0.9680	0.2945	0.2635	0.2906
RECALL	0.9182	0.9535	0.9062	0.7666	0.9426	0.9507	0.9191	0.8166	0.4529	0.3124	0.2875	0.3357	0.9320	0.8949	0.8641	0.8470	0.2392	0.4306	0.3323	0.2988

emg_4_v3	Bagging				Ibk				LWL				AdaBoostM1				ClassificationViaRegression			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	89.2875				65.7989				45.1704				34.0298				82.3257			
TP	0.9553	0.8756	0.8987	0.8419	0.8048	0.8405	0.2555	0.7358	0.8162	0.9990	0	0	0.3698	0.8987	0.0982	0	0.8842	0.8808	0.7863	0.7426
TN	0.9789	0.9733	0.9604	0.9445	0.9922	0.7920	0.9634	0.7966	0.9462	0.3241	1	1	0.9784	0.2316	0.9116	1	0.9630	0.9352	0.9374	0.9287
ROC	0.9958	0.9836	0.9793	0.9640	0.8987	0.8138	0.6088	0.7658	0.9027	0.7285	0.7385	0.6456	0.6785	0.5746	0.5537	0.5441	0.9716	0.9626	0.9337	0.9037
Precision	0.9376	0.9157	0.8843	0.8350	0.9718	0.5721	0.7015	0.5469	0.8342	0.3284	0	0	0.8506	0.2790	0.2724	0	0.8882	0.8180	0.8088	0.7767
RECALL	0.9553	0.8756	0.8987	0.8419	0.8048	0.8405	0.2555	0.7358	0.8162	0.9990	0	0	0.3698	0.8987	0.0982	0	0.8842	0.8808	0.7863	0.7426

emg_4_v3	LogitBoost				Jrip				ZeroR				HoeffdingTree				J48			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	81.7777				82.9937				25.2012				88.5939				80.5874			
TP	0.8605	0.8829	0.7880	0.7406	0.9426	0.8340	0.7747	0.7693	0	0	1	0	0.9182	0.9518	0.9079	0.7663	0.8770	0.8312	0.7873	0.7286
TN	0.9745	0.9161	0.9459	0.9206	0.9773	0.9334	0.9369	0.9255	1	1	0	1	0.9721	0.9393	0.9841	0.9526	0.9607	0.9331	0.9349	0.9125
ROC	0.9815	0.9660	0.9540	0.9213	0.9761	0.9278	0.9119	0.8902	0.4999	0.4995	0.4995	0.4996	0.9876	0.9880	0.9884	0.9488	0.9263	0.8832	0.8450	0.8005
Precision	0.9179	0.7769	0.8306	0.7569	0.9324	0.8057	0.8054	0.7752	0	0	0.2520	0	0.9160	0.8383	0.9506	0.8436	0.8809	0.8043	0.8028	0.7354
RECALL	0.8605	0.8829	0.7880	0.7406	0.9426	0.8340	0.7747	0.7693	0	0	1	0	0.9182	0.9518	0.9079	0.7663	0.8770	0.8312	0.7873	0.7286

emg_4_v3	RandomForest				RandomTree				DecisionTable				WekaDeepLearning4J				Vote			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	92.8926				74.4905				66.4326				33.7301				25.2012			
TP	0.9735	0.9328	0.9419	0.8676	0.8079	0.7906	0.7200	0.6619	0.7196	0.8312	0.5647	0.5438	0.4300	0.2970	0.3170	0.3050	0	0	1	0
TN	0.9816	0.9813	0.9726	0.9696	0.9451	0.9274	0.9064	0.8809	0.9235	0.8581	0.8948	0.8761	0.8147	0.7719	0.7637	0.7648	1	1	0	1
ROC	0.9983	0.9933	0.9875	0.9838	0.8765	0.8590	0.8132	0.7714	0.9249	0.9037	0.8291	0.7989	0.5820	0.5540	0.5280	0.5460	0.4999	0.4995	0.4995	0.4996
Precision	0.9462	0.9429	0.9206	0.9050	0.8302	0.7827	0.7215	0.6496	0.7573	0.6597	0.6439	0.5942	0.4590	0.3150	0.2940	0.2930	0	0	0.2520	0
RECALL	0.9735	0.9328	0.9419	0.8676	0.8079	0.7906	0.7200	0.6619	0.7196	0.8312	0.5647	0.5438	0.4300	0.2970	0.3170	0.3050	0	0	1	0

Table 4.7 Results with deducted 41 features and 20 algorithms

emg_4_v4	NaiveBayes				BayesNet				Logistic				Multilayer Perceptron				SMO			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	89.4845				90.9574				33.7986				82.1117				30.9899			
TP	0.9265	0.9383	0.9110	0.8039	0.9481	0.9370	0.9235	0.8299	0.5014	0.2377	0.2559	0.3576	0.9296	0.8333	0.8012	0.7211	0.2381	0.3810	0.3099	0.3107
TN	0.9877	0.9410	0.9809	0.9503	0.9883	0.9556	0.9742	0.9614	0.8365	0.8138	0.7599	0.7069	0.9745	0.9181	0.9486	0.9204	0.9989	0.6670	0.6924	0.7213
ROC	0.9937	0.9832	0.9865	0.9614	0.9976	0.9865	0.9881	0.9710	0.5959	0.5425	0.5080	0.5431	0.9828	0.9535	0.9366	0.9118	0.6357	0.5231	0.5044	0.5297
Precision	0.9615	0.8402	0.9414	0.8438	0.9640	0.8746	0.9235	0.8777	0.5043	0.2969	0.2642	0.2894	0.9235	0.7709	0.8400	0.7514	0.9858	0.2746	0.2534	0.2712
RECALL	0.9265	0.9383	0.9110	0.8039	0.9481	0.9370	0.9235	0.8299	0.5014	0.2377	0.2559	0.3576	0.9296	0.8333	0.8012	0.7211	0.2381	0.3810	0.3099	0.3107

emg_4_v4	Bagging				Ibk				LWL				AdaBoostM1				ClassificationViaRegression			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	89.3561				72.8892				47.3454				34.0298				82.8053			
TP	0.9581	0.8712	0.9076	0.8374	0.9450	0.8591	0.5229	0.5917	0.9041	0.9983	0	0	0.3698	0.8987	0.0982	0	0.8945	0.8794	0.7924	0.7467
TN	0.9800	0.9700	0.9634	0.9446	0.9973	0.7768	0.9777	0.8872	0.9199	0.3793	1	1	0.9784	0.2316	0.9116	1	0.9669	0.9311	0.9469	0.9259
ROC	0.9965	0.9813	0.9801	0.9618	0.9723	0.8165	0.7513	0.7394	0.9298	0.8390	0.7863	0.6704	0.6785	0.5746	0.5537	0.5441	0.9759	0.9605	0.9389	0.9086
Precision	0.9409	0.9058	0.8930	0.8346	0.9913	0.5601	0.8875	0.6364	0.7894	0.3473	0	0	0.8506	0.2790	0.2724	0	0.8998	0.8084	0.8340	0.7708
RECALL	0.9581	0.8712	0.9076	0.8374	0.9450	0.8591	0.5229	0.5917	0.9041	0.9983	0	0	0.3698	0.8987	0.0982	0	0.8945	0.8794	0.7924	0.7467

emg_4_v4	LogitBoost				Jrip				ZeroR				HoeffdingTree				J48			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	82.0346				83.8157				25.2012				89.5444				82.0089			
TP	0.8660	0.8870	0.7920	0.7372	0.9522	0.8529	0.7941	0.7543	0	0	1	0	0.9282	0.9387	0.9096	0.8056	0.8907	0.8405	0.8053	0.7444
TN	0.9739	0.9160	0.9471	0.9235	0.9778	0.9365	0.9396	0.9303	1	1	0	1	0.9875	0.9411	0.9815	0.9507	0.9629	0.9407	0.9418	0.9146
ROC	0.9824	0.9653	0.9541	0.9213	0.9785	0.9398	0.9201	0.8995	0.4999	0.4995	0.4995	0.4996	0.9935	0.9831	0.9863	0.9612	0.9348	0.8941	0.8686	0.8153
Precision	0.9167	0.7775	0.8346	0.7627	0.9343	0.8164	0.8157	0.7832	0	0	0.2520	0	0.9609	0.8405	0.9429	0.8449	0.8886	0.8243	0.8235	0.7441
RECALL	0.8660	0.8870	0.7920	0.7372	0.9522	0.8529	0.7941	0.7543	0	0	1	0	0.9282	0.9387	0.9096	0.8056	0.8907	0.8405	0.8053	0.7444

emg_4_v4	RandomForest				RandomTree				DecisionTable				WekaDeepLearning4J				Vote			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	91.9764				78.9519				66.4326				34.8433				25.2012			
TP	0.9773	0.9029	0.9324	0.8665	0.8780	0.8085	0.7666	0.7057	0.7196	0.8312	0.5647	0.5438	0.4730	0.2410	0.3300	0.3490	0	0	1	0
TN	0.9848	0.9756	0.9715	0.9611	0.9629	0.9287	0.9272	0.9005	0.9235	0.8581	0.8948	0.8761	0.8295	0.7635	0.7685	0.7685	1	1	0	1
ROC	0.9987	0.9881	0.9862	0.9772	0.9205	0.8686	0.8469	0.8031	0.9249	0.9037	0.8291	0.7989	0.5910	0.5500	0.5240	0.5480	0.4999	0.4995	0.4995	0.4996
Precision	0.9553	0.9245	0.9168	0.8813	0.8872	0.7894	0.7801	0.7030	0.7573	0.6597	0.6439	0.5942	0.5120	0.2960	0.3070	0.2950	0	0	0.2520	0
RECALL	0.9773	0.9029	0.9324	0.8665	0.8780	0.8085	0.7666	0.7057	0.7196	0.8312	0.5647	0.5438	0.4730	0.2410	0.3300	0.3490	0	0	1	0

Table 4.8 Results with deducted 1 feature and 20 algorithms

emg_4_v5	NaiveBayes				BayesNet				Logistic				Multilayer Perceptron				SMO			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	88.5682				90.5635				34.7748				88.5939				32.2315			
TP	0.9179	0.9518	0.9062	0.7673	0.9392	0.9497	0.9191	0.8149	0.4584	0.3159	0.2844	0.3330	0.9206	0.9101	0.8658	0.8477	0.2392	0.4302	0.3184	0.3018
TN	0.9727	0.9391	0.9842	0.9516	0.9722	0.9656	0.9750	0.9614	0.8463	0.7879	0.7350	0.7610	0.9778	0.9600	0.9653	0.9448	0.9977	0.6548	0.6912	0.7524
ROC	0.9874	0.9884	0.9885	0.9486	0.9942	0.9925	0.9889	0.9608	0.5917	0.5491	0.5191	0.5467	0.9834	0.9835	0.9634	0.9539	0.6480	0.5375	0.5128	0.5208
Precision	0.9179	0.8380	0.9508	0.8410	0.9180	0.9013	0.9254	0.8757	0.4974	0.3301	0.2655	0.3174	0.9322	0.8827	0.8937	0.8368	0.9721	0.2920	0.2578	0.2892
RECALL	0.9179	0.9518	0.9062	0.7673	0.9392	0.9497	0.9191	0.8149	0.4584	0.3159	0.2844	0.3330	0.9206	0.9101	0.8658	0.8477	0.2392	0.4302	0.3184	0.3018

emg_4_v5	Bagging				Ibk				LWL				AdaBoostM1				ClassificationViaRegression			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	89.2790				65.5506				45.0762				34.0298				82.2487			
TP	0.9543	0.8750	0.8991	0.8429	0.8007	0.8422	0.2552	0.7286	0.8124	0.9990	0	0	0.3698	0.8987	0.0982	0	0.8835	0.8805	0.7836	0.7433
TN	0.9788	0.9738	0.9599	0.9445	0.9920	0.7888	0.9624	0.7976	0.9462	0.3228	1	1	0.9784	0.2316	0.9116	1	0.9632	0.9349	0.9363	0.9288
ROC	0.9957	0.9835	0.9792	0.9638	0.8967	0.8133	0.6081	0.7628	0.9005	0.7282	0.7371	0.6449	0.6785	0.5746	0.5537	0.5441	0.9715	0.9623	0.9337	0.9041
Precision	0.9372	0.9170	0.8832	0.8352	0.9708	0.5689	0.6960	0.5458	0.8336	0.3280	0	0	0.8506	0.2790	0.2724	0	0.8884	0.8174	0.8057	0.7771
RECALL	0.9543	0.8750	0.8991	0.8429	0.8007	0.8422	0.2552	0.7286	0.8124	0.9990	0	0	0.3698	0.8987	0.0982	0	0.8835	0.8805	0.7836	0.7433

emg_4_v5	LogitBoost				Jrip				ZeroR				HoeffdingTree				J48			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	81.7777				83.6445				25.2012				88.5340				80.5617			
TP	0.8605	0.8829	0.7880	0.7406	0.9457	0.8508	0.7870	0.7632	0	0	1	0	0.9182	0.9497	0.9083	0.7656	0.8759	0.8316	0.7876	0.7279
TN	0.9745	0.9161	0.9459	0.9206	0.9783	0.9363	0.9376	0.9296	1	1	0	1	0.9722	0.9391	0.9845	0.9513	0.9603	0.9331	0.9346	0.9127
ROC	0.9815	0.9660	0.9540	0.9213	0.9759	0.9310	0.9156	0.8937	0.4999	0.4995	0.4995	0.4996	0.9873	0.9879	0.9884	0.9482	0.9255	0.8836	0.8444	0.7999
Precision	0.9179	0.7769	0.8306	0.7569	0.9354	0.8155	0.8095	0.7836	0	0	0.2520	0	0.9163	0.8377	0.9519	0.8400	0.8799	0.8044	0.8024	0.7357
RECALL	0.8605	0.8829	0.7880	0.7406	0.9457	0.8508	0.7870	0.7632	0	0	1	0	0.9182	0.9497	0.9083	0.7656	0.8759	0.8316	0.7876	0.7279

emg_4_v5	RandomForest				RandomTree				DecisionTable				WekaDeepLearning4J				Vote			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	92.5073				74.9443				66.4326				33.6530				25.2012			
TP	0.9722	0.9280	0.9399	0.8604	0.8326	0.7871	0.7136	0.6653	0.7196	0.8312	0.5647	0.5438	0.4290	0.2950	0.3170	0.3050	0	0	1	0
TN	0.9811	0.9809	0.9708	0.9673	0.9434	0.9273	0.9118	0.8833	0.9235	0.8581	0.8948	0.8761	0.8140	0.7715	0.7633	0.7654	1	1	0	1
ROC	0.9982	0.9933	0.9877	0.9831	0.8880	0.8572	0.8127	0.7743	0.9249	0.9037	0.8291	0.7989	0.5840	0.5540	0.5290	0.5460	0.4999	0.4995	0.4995	0.4996
Precision	0.9446	0.9413	0.9156	0.8979	0.8301	0.7817	0.7317	0.6554	0.7573	0.6597	0.6439	0.5942	0.4550	0.3150	0.2930	0.2950	0	0	0.2520	0
RECALL	0.9722	0.9280	0.9399	0.8604	0.8326	0.7871	0.7136	0.6653	0.7196	0.8312	0.5647	0.5438	0.4290	0.2950	0.3170	0.3050	0	0	1	0

Table 4.9 Results with deducted 35 features and 20 algorithms

emg_4_v6	NaiveBayes				BayesNet				Logistic				Multilayer Perceptron				SMO			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	89.9469				91.5996				34.4494				85.8537				32.4456			
TP	0.9292	0.9439	0.9083	0.8169	0.9540	0.9435	0.9259	0.8409	0.4904	0.2776	0.2429	0.3679	0.9220	0.8760	0.8617	0.7748	0.2405	0.4743	0.2987	0.2851
TN	0.9872	0.9468	0.9790	0.9529	0.9885	0.9618	0.9726	0.9651	0.8371	0.7997	0.7742	0.7148	0.9746	0.9394	0.9560	0.9414	0.9990	0.6223	0.7086	0.7692
ROC	0.9938	0.9869	0.9865	0.9645	0.9979	0.9895	0.9882	0.9736	0.5960	0.5434	0.5058	0.5479	0.9848	0.9696	0.9572	0.9328	0.6419	0.5426	0.5065	0.5223
Precision	0.9602	0.8544	0.9359	0.8528	0.9649	0.8910	0.9194	0.8893	0.4998	0.3144	0.2661	0.3010	0.9233	0.8270	0.8685	0.8153	0.9873	0.2935	0.2567	0.2919
RECALL	0.9292	0.9439	0.9083	0.8169	0.9540	0.9435	0.9259	0.8409	0.4904	0.2776	0.2429	0.3679	0.9220	0.8760	0.8617	0.7748	0.2405	0.4743	0.2987	0.2851

emg_4_v6	Bagging				Ibk				LWL				AdaBoostM1				ClassificationViaRegression			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	89.6044				71.3992				46.9858				34.0298				82.8224			
TP	0.9588	0.8750	0.9038	0.8467	0.9292	0.8577	0.4207	0.6523	0.8893	0.9986	0	0	0.3698	0.8987	0.0982	0	0.8952	0.8877	0.7937	0.7372
TN	0.9791	0.9733	0.9641	0.9448	0.9960	0.7895	0.9703	0.8631	0.9253	0.3691	1	1	0.9784	0.2316	0.9116	1	0.9611	0.9342	0.9415	0.9341
ROC	0.9966	0.9832	0.9801	0.9644	0.9644	0.8223	0.6939	0.7589	0.9208	0.7887	0.7576	0.6575	0.6785	0.5746	0.5537	0.5441	0.9742	0.9636	0.9376	0.9067
Precision	0.9384	0.9156	0.8944	0.8367	0.9872	0.5741	0.8270	0.6138	0.7980	0.3437	0	0	0.8506	0.2790	0.2724	0	0.8842	0.8171	0.8205	0.7887
RECALL	0.9588	0.8750	0.9038	0.8467	0.9292	0.8577	0.4207	0.6523	0.8893	0.9986	0	0	0.3698	0.8987	0.0982	0	0.8952	0.8877	0.7937	0.7372

emg_4_v6	LogitBoost				Jrip				ZeroR				HoeffdingTree				J48			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	81.9575				82.7796				25.2012				89.9127				81.6493			
TP	0.8643	0.8856	0.7917	0.7375	0.9502	0.8412	0.7829	0.7379	0	0	1	0	0.9302	0.9459	0.9079	0.8128	0.8890	0.8422	0.7985	0.7368
TN	0.9735	0.9169	0.9469	0.9221	0.9786	0.9339	0.9260	0.9318	1	1	0	1	0.9867	0.9460	0.9799	0.9531	0.9632	0.9369	0.9407	0.9146
ROC	0.9822	0.9659	0.9548	0.9212	0.9773	0.9298	0.9019	0.8853	0.4999	0.4995	0.4995	0.4996	0.9937	0.9867	0.9864	0.9643	0.9313	0.8991	0.8642	0.8141
Precision	0.9155	0.7791	0.8339	0.7596	0.9363	0.8081	0.7810	0.7831	0	0	0.2520	0	0.9586	0.8528	0.9382	0.8525	0.8890	0.8153	0.8194	0.7422
RECALL	0.8643	0.8856	0.7917	0.7375	0.9502	0.8412	0.7829	0.7379	0	0	1	0	0.9302	0.9459	0.9079	0.8128	0.8890	0.8422	0.7985	0.7368

emg_4_v6	RandomForest				RandomTree				DecisionTable				WekaDeepLearning4J				Vote			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
TA	92.0363				78.0527				66.4326				34.7063				25.2012			
TP	0.9766	0.9094	0.9320	0.8634	0.8625	0.8016	0.7503	0.7084	0.7196	0.8312	0.5647	0.5438	0.4690	0.2650	0.3160	0.3390	0	0	1	0
TN	0.9854	0.9766	0.9693	0.9624	0.9603	0.9315	0.9256	0.8899	0.9235	0.8581	0.8948	0.8761	0.8200	0.7940	0.7640	0.7618	1	1	0	1
ROC	0.9987	0.9902	0.9871	0.9800	0.9114	0.8665	0.8379	0.7992	0.9249	0.9037	0.8291	0.7989	0.5920	0.5520	0.5230	0.5510	0.4999	0.4995	0.4995	0.4996
Precision	0.9569	0.9279	0.9110	0.8846	0.8782	0.7947	0.7726	0.6823	0.7573	0.6597	0.6439	0.5942	0.4890	0.3110	0.2950	0.3030	0	0	0.2520	0
RECALL	0.9766	0.9094	0.9320	0.8634	0.8625	0.8016	0.7503	0.7084	0.7196	0.8312	0.5647	0.5438	0.4690	0.2650	0.3160	0.3390	0	0	1	0

Table 4.10 Results of Random Forest According to Versions

	emg_4_v1				emg_4_v2				emg_4_v3			
	0	1	2	3	0	1	2	3	0	1	2	3
TA	92.6357				90.1010				92.8926			
TP	0.9729	0.9232	0.9392	0.8703	0.9354	0.9008	0.9276	0.8402	0.9735	0.9328	0.9419	0.8676
TN	0.9822	0.9810	0.9725	0.9661	0.9768	0.9736	0.9685	0.9491	0.9816	0.9813	0.9726	0.9696
ROC	0.9982	0.9926	0.9884	0.9826	0.9919	0.9869	0.9853	0.9657	0.9983	0.9933	0.9875	0.9838
PRECISION	0.9478	0.9413	0.9201	0.8954	0.9306	0.9185	0.9085	0.8463	0.9462	0.9429	0.9206	0.9050
RECALL	0.9729	0.9232	0.9392	0.8703	0.9354	0.9008	0.9276	0.8402	0.9735	0.9328	0.9419	0.8676

	emg_4_v4				emg_4_v5				emg_4_v6			
	0	1	2	3	0	1	2	3	0	1	2	3
TA	91.9764				92.5073				92.0363			
TP	0.9773	0.9029	0.9324	0.8665	0.9722	0.9280	0.9399	0.8604	0.9766	0.9094	0.9320	0.8634
TN	0.9848	0.9756	0.9715	0.9611	0.9811	0.9809	0.9708	0.9673	0.9854	0.9766	0.9693	0.9624
ROC	0.9987	0.9881	0.9862	0.9772	0.9982	0.9933	0.9877	0.9831	0.9987	0.9902	0.9871	0.9800
PRECISION	0.9553	0.9245	0.9168	0.8813	0.9446	0.9413	0.9156	0.8979	0.9569	0.9279	0.9110	0.8846
RECALL	0.9773	0.9029	0.9324	0.8665	0.9722	0.9280	0.9399	0.8604	0.9766	0.9094	0.9320	0.8634

CHAPTER 5
CONCLUSIONS AND FUTURE RESEARCH

In everyday life, messages are transmitted using different gestures. The gestures can be a nod, shake hands, facial expressions, eye movements or any gait movement. It can be used in various applications such as gesture recognition, gait recognition, sign language recognition, computer virtual mouse, television control, human/robot manipulation.

The terms gesture and gesture recognition, human-computer interaction is increasingly encountered. Movements are physical actions that transmit meaningful information. As a new input method in human-computer interactions, systems designed to recognize gestures contribute a heuristic and more comfortable manner of interaction. If the world of computers is forgotten and the interaction between people is kept in mind for a while, it can easily be noticed that a variety of movements are used in personal expression. The significant use of gestures in daily life motivates the use of the gestural interface in the modern era. Hand and arm movement have become the perfect choice for expressing simple ideas, interpreting them by gesture recognition, and turning them into corresponding events. Different hand and arm movements classified on the basis of unique hand shapes, patterns or gestures, orientation, or finger positions have the potential to interact with the computer system to transmit reliable information for human-computer interaction.

The aim of this thesis is creating a system having a gesture recognition capability by taking advantage of human-computer interaction. The structure of the datasets consists of 11679 recordings, as quantity 8 consecutive readings from all 8 sensors, which are raw muscle activity data, and the attribute class, which are certain types of motion like gestures of rock, paper, scissors and okay sign. In spite of achieving the goal, 20 classifiers were used to make multi-class classifications using the dataset, 6 of them were chosen which gave the best results and these algorithms were selected as Naive Bayes, BayesNet, Multilayer Perceptron, Bagging, Hoeffding Tree and Random Forest.

However, Random Forest is the best selected algorithm among them with 92.8926% total accuracy and 0.9987 for 0-class, 0.9881 for 1-class, 0.9862 for 2-class, 0.9772 for 4-class of TP.

Thanks to this study, we can make motion recognition according to the data coming from EMG sensors. Thus, in the entertainment industry, it can make it easier to communicate with kinect while playing games wirelessly. As a different example, by contributing to human-computer interaction, we can communicate with robots, move them or use them as a tool of remote access. As further studies, the question of whether it can teach all movements instead of restricted movements can be investigated. This helps not only people with disabilities but also those with different disabilities. In the health sector, in robotic surgery, studies can be carried out in order to ensure comfortable communication between the doctor and the robot.

REFERENCES

Aha, D., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, *6*(1), 37-66. doi: 10.1007/bf00153759.

Basarici, Samsun M. (2019). CSE310: İnsan-Bilgisayar Etkileşimi, session 1 notes [Course Presentation]. Retrieved from https://akademik.adu.edu.tr/fakulte/muhendislik/personel/sbasarici/dersler

Bartlett, M., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., & Movellan, J. (2005). Recognizing facial expression: machine learning and application to spontaneous behavior. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (pp. 568-573). San Diego, CA, USA: IEEE.

Benalcazar, M., Jaramillo, A., Zea, J., Zea, A., Paez, A., & Andaluz, V. (2017). Hand gesture recognition using machine learning and the Myo armband. *2017 25Th European Signal Processing Conference (EUSIPCO)*. doi: 10.23919/eusipco.2017.8081366

Bhattacharya, S., Czejdo, B., & Perez, N. (2012). Gesture Classification with Machine Learning Using Kinect Sensor Data. In *Third International Conference on Emerging Applications of Information Technology* (pp. 348-351). IEEE. doi: 10.1109/EAIT.2012.6407958.

Boyd, J., & Little, J. (2005). Biometric Gait Recognition. In M. Tistarelli, J. Bigun & E. Grosso, *Advanced Studies in Biometrics* (pp. 19-42). Berlin, Heidelberg: Springer-Verlag.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123-140. doi: 10.1007/bf00058655.

Brownlee, J. (2019). Bagging and Random Forest Ensemble Algorithms for Machine Learning. Retrieved from https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/

Brownlee, J. (2019). How to Use Ensemble Machine Learning Algorithms in Weka. Retrieved from https://machinelearningmastery.com/use-ensemble-machine-learning-algorithms-weka/

Castellano, G., Villalba, S., & Camurri, A. (2007). Recognising Human Emotions from Body Movement and Gesture Dynamics. In *International Conference on Affective Computing and Intelligent Interaction* (pp. 71-82). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-540-74889-2_7.

Cohen, W. (1995). Fast Effective Rule Induction. *Machine Learning Proceedings 1995*, 115-123. doi: 10.1016/b978-1-55860-377-6.50023-2.

Data Mining: Practical Machine Learning Tools and Techniques. Retrieved from https://www.cs.waikato.ac.nz/~ml/weka/book.html

Dietterich, T. (2000). Ensemble Methods in Machine Learning. *Multiple Classifier Systems*, 1-15. doi: 10.1007/3-540-45014-9_1.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (2003). *Human-computer interaction* (3rd ed.). London: Pearson.

Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. *Proceedings of The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '00*. doi: 10.1145/347090.347107.

Frank, E. ZeroR. Retrieved from http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/ZeroR.html

Frank, E., Hall, M., & Pfahringer, B. (2002, August). Locally Weighted Naive Bayes. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence* (pp. 249-256). Morgan Kaufmann Publishers Inc..

Frank, E., & Trigg, L. ClassificationViaRegression. Retrieved from http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/ClassificationViaRegression.html

Guo, G., Li, S., & Chan, K. (2002). Face recognition by support vector machines. *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. doi: 10.1109/afgr.2000.840634

Gupta, P. (2017). Decision Trees in Machine Learning. Retrieved from https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

Ho, M., Yamada, Y., & Umetani, Y. (2005). An adaptive visual attentive tracker for human communicational behaviors using HMM-based TD learning with new State distinction capability. *IEEE Transactions on Robotics, 21*(3), 497-504. doi: 10.1109/tro.2004.840912

Joshi, R. (2016). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog. Retrieved from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

Kaggle. (2019). Classify gestures by reading muscle activity. Retrieved from https://www.kaggle.com/kyr7plus/emg-4

Kalmegh, S. (2018). Comparative Analysis of the WEKA Classifiers Rules Conjunctive Rule & Decision Table on Indian News Dataset by Using Different Test Mode. *International Journal of Engineering Science Invention (IJESI)*, *7*(2), 1-9.

Kapur, A., Kapur, A., Virji-Babul, N., Tzanetakis, G., & Driessen, P. (2005). Gesture-Based Affective Computing on Motion Capture Data. In *International Conference on Affective Computing and Intelligent Interaction* (pp. 1-7). Berlin, Heidelberg: Springer. doi: 10.1007/11573548_1

Khan, R., & Ibraheem, N. (2012). Hand Gesture Recognition: A Literature Review. *International Journal of Artificial Intelligence & Applications*, *3*(4), 161-174. doi: 10.5121/ijaia.2012.3412

Lindlöf, A., Bräutigam, M., Chawade, A., Olsson, O., & Olsson, B. (2008). Evaluation of Combining Several Statistical Methods with a Flexible Cutoff for Identifying Differentially Expressed Genes in Pairwise Comparison of EST Sets. *Bioinformatics and Biology Insights*, *2*, BBI.S431. doi: 10.4137/bbi.s431

Lu, Z., Chen, X., Zhang, X., Tong, K., & Zhou, P. (2017). Real-Time Control of an Exoskeleton Hand Robot with Myoelectric Pattern Recognition. *International Journal of Neural Systems*, *27*(05), 1750009. doi: 10.1142/s0129065717500095

Mane, S., Kambli, R., Kazi, F., & Singh, N. (2015). Hand Motion Recognition from Single Channel Surface EMG Using Wavelet & Artificial Neural Network. *Procedia Computer Science*, *49*, 58-65. doi: 10.1016/j.procs.2015.04.227

Mehdi, T., Bashardoost, N., & Ahmadi, M. (2011). Kernel Smoothing For ROC Curve And Estimation For Thyroid Stimulating Hormone. *International Journal Of Public Health Research*, (2011), 239-242.

Mishra, A. K., & Ratha, B. K. (2016). Study of Random Tree and Random Forest Data Mining Algorithms for Microarray Data Analysis. *International Journal on Advanced Electrical and Computer Engineering*, *3*(4), 5-7.

Mitra, S., & Acharya, T. (2007). Gesture Recognition: A Survey. IEEE Transactions on Systems, Man And Cybernetics, Part C (Applications And Reviews), 37(3), 311-324. doi: 10.1109/tsmcc.2007.893280

Mohri, M., Talwalkar, A., & Rostamizadeh, A. (2012). *Foundations of Machine Learning (Adaptive Computation and Machine Learning Series)* (p. 7). Cambridge, Mass: MIT Press.

Moon, I., Lee, M., Ryu, J., & Mun, M. (2003). Intelligent robotic wheelchair with EMG-, gesture-, and voice-based interfaces. *Proceedings 2003 IEEE/RSJ International Conference On Intelligent Robots And Systems (IROS 2003) (Cat. No.03CH37453)*. doi: 10.1109/iros.2003.1249690.

Naik, G., Kumar, D., & Weghorn, H. (2007). Performance comparison of ICA algorithms for Isometric Hand gesture identification using Surface EMG. In *3rd International Conference on Intelligent Sensors, Sensor Networks and Information* (pp. 613-618). Melbourne, Qld., Australia: IEEE. doi: 10.1109/ISSNIP.2007.4496913.

NetBeans IDE 8.2 Release Information. Retrieved from https://netbeans.org/community/releases/82/

NetBeans Platform Description. Retrieved from https://netbeans.org/features/platform/features.html

Nielsen, J. (2012). Usability 101: Introduction to Usability. Retrieved from https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Nishihara, H. K., Hsu, S. P., Kaehler, A., & Jangaard, L. (2017). Hand-gesture recognition method. *U.S. Patent No. US9696808B2*. Washington, DC: U.S. Patent and Trademark Office.

Palkowski, A., & Redlarski, G. (2016). Basic Hand Gestures Classification Based on Surface Electromyography. *Computational and Mathematical Methods in Medicine*, *2016*, 1-7. doi: 10.1155/2016/6481282

Plawiak, P., Sosnicki, T., Niedzwiecki, M., Tabor, Z., & Rzecki, K. (2016). Hand Body Language Gesture Recognition Based on Signals from Specialized Glove and Machine Learning Algorithms. *IEEE Transactions on Industrial Informatics*, *12*(3), 1104-1113. doi: 10.1109/tii.2016.2550528

Royalty Vectors (2019). Hand drawn hand with thumb up vector image on VectorStock. Retrieved from https://www.vectorstock.com/royalty-free-vector/hand-drawn-hand-with-thumb-up-vector-8446201

Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210-229. doi: 10.1147/rd.33.0210

Shin, S., Kim, D., & Seo, Y. (2014). Controlling Mobile Robot Using IMU and EMG Sensor-Based Gesture Recognition. In *Ninth International Conference on Broadband and Wireless Computing, Communication and Applications* (pp. 554-557). Guangdong, China: IEEE. doi:10.1109/BWCCA.2014.145.

Sketch Hand Gesture Vector. (2019). Retrieved from https://www.vecteezy.com/vector-art/168473-sketch-hand-gesture-vector

Tatarian, K., Couceiro, M., Ribeiro, E., & Faria, D. (2018). Stepping-stones to Transhumanism: An EMG-controlled Low-cost Prosthetic Hand for Academia. *2018 International Conference On Intelligent Systems (IS)*. doi: 10.1109/is.2018.8710489

Trigg, L. & Frank, E. LogitBoost. Retrieved from http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/LogitBoost.html

Trigueiros, P., Ribeiro, F., & Reis, L. (2012). A Comparison of Machine Learning Algorithms Applied to Hand Gesture Recognition. In *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)* (pp. 1-6). Madrid, Spain: IEEE.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, *LIX* (236), 433-460. doi: 10.1093/mind/lix.236.433

Turk, M. (2014). Gesture Recognition. *Computer Vision*, 346-349. doi: 10.1007/978-0-387-31439-6_376

Witten, I., Frank, E., & Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Morgan Kaufmann.

Weka 3 - Data Mining with Open Source Machine Learning Software in Java. (2019). Retrieved from https://www.cs.waikato.ac.nz/~ml/weka/

Wu, J., Tian, Z., Sun, L., Estevez, L., & Jafari, R. (2015). Real-time American Sign Language Recognition using wrist-worn motion and surface EMG sensors. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)* (pp. 1-6). Cambridge, MA, USA: IEEE. doi: 10.1109/BSN.2015.7299393.

Xu, X. (2011). Logistic. Retrieved from http://weka.sourceforge.net/doc.dev/weka/cFlassifiers/functions/Logistic.html

APPENDIX 1 – Sample Method for Classification

The following is the code written for Weka (reference).

```
public static void randomForest_Train(String pathFile) throws
Exception
{
    BufferedReader reader;
      Instances traindata;

      try (BufferedWriter bw = new BufferedWriter(new
FileWriter(FILENAME,true)))
      {
            reader = new BufferedReader(new FileReader(pathFile));
            traindata = new Instances(reader);
            reader.close();
            traindata.setClassIndex(traindata.numAttributes() - 1);

            Classifier cls = new RandomForest();
            Evaluation eval = new Evaluation(traindata);
            eval.crossValidateModel(cls, traindata, 10, new
Random(1));
            System.out.println(eval.pctCorrect()/100);
            System.out.println(eval.truePositiveRate(0));
            System.out.println(eval.trueNegativeRate(0));
            System.out.println(eval.weightedAreaUnderROC());
            String txt = "RandomForest\t" + eval.pctCorrect() + "\t"
+ eval.truePositiveRate(0) + "\t" + eval.trueNegativeRate(0) + "\t"
+ eval.areaUnderROC(0)+ "\t" + eval.precision(0)+ "\t" +
eval.recall(0)+"\n";
            bw.write(txt);
            txt = "RandomForest\t" + eval.pctCorrect() + "\t" +
eval.truePositiveRate(1) + "\t" + eval.trueNegativeRate(1) + "\t" +
eval.areaUnderROC(1)+ "\t" + eval.precision(1)+ "\t" +
eval.recall(1)+"\n";
            bw.write(txt);
            txt = "RandomForest\t" + eval.pctCorrect() + "\t" +
eval.truePositiveRate(2) + "\t" + eval.trueNegativeRate(2) + "\t" +
eval.areaUnderROC(2)+ "\t" + eval.precision(2)+ "\t" +
eval.recall(2)+"\n";
            bw.write(txt);
            txt = "RandomForest\t" + eval.pctCorrect() + "\t" +
eval.truePositiveRate(3) + "\t" + eval.trueNegativeRate(3) + "\t" +
eval.areaUnderROC(3)+ "\t" + eval.precision(3)+ "\t" +
eval.recall(3)+"\n";
            bw.write(txt+"\n");

            String txt2 = "RandomForest\t" +
eval.toSummaryString()+"\n";
            bw.write(txt2);
    }
    catch (IOException e) {e.printStackTrace();}
}
```