**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY**

# MACHINE LEARNING TECHNIQUES FOR SURFACE ELECTROMYOGRAPHY BASED HAND GESTURE RECOGNITION

**M.Sc. THESIS**

**Engin KAYA**

**Department of Control and Automation Engineering**

**Control and Automation Engineering Programme**

**JUNE 2018**

**ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY**

# MACHINE LEARNING TECHNIQUES FOR SURFACE ELECTROMYOGRAPHY BASED HAND GESTURE RECOGNITION

**M.Sc. THESIS**

**Engin KAYA**
**(504151109)**

**Department of Control and Automation Engineering**

**Control and Automation Engineering Programme**

**Thesis Advisor: Assoc. Prof. Dr. Tufan KUMBASAR**

**JUNE 2018**

**ISTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# YÜZEY ELEKTROMİYOGRAFİ TEMELLİ EL JESTİ TANIMA İÇİN MAKİNE ÖĞRENMESİ TEKNİKLERİ

**YÜKSEK LİSANS TEZİ**

**Engin KAYA**
**(504151109)**

**Kontrol ve Otomasyon Mühendisliği Anabilim Dalı**

**Kontrol ve Otomasyon Mühendisliği Programı**

**Tez Danışmanı: Doç. Dr. Tufan KUMBASAR**

**HAZİRAN 2018**

Engin KAYA, a M.Sc. student of İTU Graduate School of Science Engineering and Technology student ID 504151109, successfully defended the thesis/dissertation entitled "MACHINE LEARNING TECHNIQUES FOR SURFACE ELECTROMYOGRAPHY BASED HAND GESTURE RECOGNITION", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

**Thesis Advisor :** **Assoc. Prof. Dr. Tufan KUMBASAR**
Istanbul Technical University

**Jury Members :** **Asst. Prof. Dr. Volkan SEZER**
Istanbul Technical University

**Asst. Prof. Dr. İlker ÜSTOĞLU**
Yıldız Technical University

**Date of Submission : 4 May 2018**
**Date of Defense : 6 June 2018**

***To my family,***

# FOREWORD

I am grateful for my advisor Asst. Prof. Tufan Kumbasar, who accepted me to work with him and he has done to help me. Thanks for him supports.

I am so thankful to my colleague Aykut Beke and Fethi Candan for their technical support and motivation.

Also, I want to thank all the members of Robotics Lab in Biomedical Institute of Bogazici University, where I have gained a lot of experience in the use of biosignals.

Last, but not least, I would like to thank my mother, father and brother for supporting me whatever I do.

May 2018

Engin KAYA
Control and Automation Engineer

# TABLE OF CONTENTS

**Page**

# ABBREVIATIONS

**ANN** : Artificial Neural Network
**DT** : Decision Tree
**EMG** **:** Electromyography
**EN** **:** Elman Network
**FCRN** **:** Fully Connected Recurrent Network
**FN** : False Negative
**FP** : False Positive
**gdi** : Gini's Diversity Index
**IMU** : Internal Measurement Unit
**kNN** : k-Nearest Neighbor
**MAV** : Mean Absolute Value
**MLP** : Multi Layered Perceptron
**MSE** : Mean Square Error
**MUAP** : Motor Unit Action Potential
**OVA** : One versus All
**OVO** : One versus One
**PCA** : Principal Component Analysis
**POV** : Proportion of Variance
**RBF** : Radial Basis Function
**SDK** : Software Development Kit
**sEMG** : Surface Electromyography
**SSC** : Slope Sign Change
**SVM** : Support Vector Machine
**TN** : True Negative
**TP** : True Positive
**TSL** : Turkish Sign Language
**ZC** : Zero Crossing

# LIST OF TABLES

**Page**

# LIST OF FIGURES

**Page**

## MACHINE LEARNING TECHNIQUES FOR SURFACE ELECTROMYOGRAPHY BASED HAND GESTURE RECOGNITION

## SUMMARY

People use their hand unconsciously to perform many task in daily life. The goal of these tasks can be a physical interaction such as holding, grasping and pushing as well as communication. Hand gestures constitute one of the most popular types of non-verbal communication. People use their hands to give basic messages such as ‘stop’, ‘come’, ‘like’, ‘dislike’ and ‘okay’. Even when texting, we now use hand gesture emojis. Moreover, hearing impaired people use hand gestures as sign language, which means hand gestures can be considered a language on their own. On the other hand, the communication does not take place only between humans, but also between humans and technological devices. Input method of these devices is generally limited with mouse, keyboard, joyistick or touchpad. The human machine interaction studies focus on developing new interfaces to overcome this restriction by simulating a natural ways of communication for humans. Here, the hand gestures arise as an interface for devices. There are various methods for hand gesture detection such as cameras, gloves and electrodes. Visual based solutions provide high recognition rate; however, they face with difficulties like light conditions, skin and background color, spatial position and orientation of the hand and fingers. Also, this kind of systems suffer from range limitations of sight view. Next, glove based systems hold people back from using their hands comfortably and may also cause loss of sensation. Hence, an alternative solution has been by generated electrode based systems. When contracted to perform an action, muscle cells generate an electrical signal. The combination of all innervated muscle cells produces superposed electrical signals called Motor Unit Action Potentials. A technique of recording and studying the myoelectric signal is defined as electromyography (EMG). This thesis mainly proposes a hand gesture recognition system using the electromyography signals of the forearm.

The muscle signals can be recorded via needle electrodes or surface electrodes. While needle electrodes are providing signals from a specific muscle group, surface electrodes sense signals that reach the surface of skin. The use of non-invasive electrodes is painless and easy; therefore, they are known as convenient for daily usages and applications. In this work, a commercial product, the Myo Armband has been used as a signal source. The Myo Armband is a wearable and wireless device, which has eight electrodes located circularly around the arm. When a user wears the Myo Armband, the device transmits the measured EMG signal via Bluetooth and the signals can be easily obtained with help of software development kit of device. Hence, it is widely used in researches and applications like prosthetic control, mobile robot navigation, virtual reality projects, game control, sign to speech translation and art works. In this study, the hand gestures of numbers between 0 and 9 in Turkish Sign Language was recorded as a dataset using the Myo Armband.

To make a better gesture recognition, the proposed system includes the comparison of four different machine learning algorithms which consist of k-Nearest Neighbor, Decision Tree, Support Vector Machines and Artificial Neural Network. In machine learning, working with directly raw data increases the complexity of implementation and processing cost. Thus, first, seven time domain features were extracted from each channel of the recorded data. To make the feature matrix simple, Principal Component Analysis was employed, and then the algorithms was fed by resultant feature vector. Before the comparison of algorithms, each algorithm was investigated to find the optimal parameters by changing the values. Firstly, k-Nearest Neighbor algorithm was examined with various number of neighbors and distance metrics. The Decision Tree algorithm was tested for two splitting criteria which are gini diversity index and entropy. The Support Vector Machine algorithm was analyzed with different kernels and binary discrimination methods. Lastly, Artificial Neural Network was investigated by changing the number of neurons in hidden layer, training function and error functions. Then the outcome of this study has obtained with setting the parameters of each classifiers to optimal. According that, Decision Tree algorithim has lowest fscore ratio which was about 0.76. k-Nearest Neighbor and Artificial Neural Network algorithms increased the result to 0.82 and 0.85, respectively. The highest fscore ratio was achieved as 0.87 using the Support Vector Machine algorithm. In addition, the performance of each class has also evaluated, and it was found that the Support Vector Machine algorithm gives the highest ratio for all gestures. Also, gestures of ‘6’, ‘7’, ‘8’ and ‘9’ have higher recognition ratio; whereas, lowest ratio was obtained for class ‘2’. The achieved results are quite promising that, the usage of muscle signals could employ as hand gesture recognition and the proposed system can be used to interact with machines.

# YÜZEY ELEKTROMİYOGRAFİ TEMELLİ EL JESTİ TANIMA İÇİN MAKİNE ÖĞRENMESİ TEKNİKLERİ

## ÖZET

İnsanlar günlük hayatta ellerini farkında olmadan pek çok ihtiyaç için kullanmaktadır. Tutma, kavrama, dokunma, çekme ve itme şeklinde nesnelerle etkileşime geçmenin yanında, eller iletişim için de sıkça kullanılmaktadır. Örneğin, uçaklarda hostlar ve hostesler emniyet ve acil durum bildirimlerini yaparken sesli olarak anlatılan bilgileri ellerini kullanarak da göstermektedirler. Bu durum işitme engelliler ve anlatımın yapıldığı dili bilmeyen kişiler için anlatılanları anlamak için zorunlu olmakla birlikte dili anlayan ve işitme engeli olmayan kişiler için de anlatımın pekişmesini sağlamaktadır. El jestlerinin kullanımı arttıkça sözlü iletişimde taşınan bilgi miktarının azaldığını gösteren çalışmalar bulunmaktadır. Ayrıca, artık yazılı iletişimde bile emojiler sayesinde basit mesajları iletmek için el işaretlerine başvurmaktayız. Bunların yanında, işitme engelli kişiler de işaret diliyle iletişim kurmak için el jestlerine ihtiyaç duymaktadır. Yani el jestleri, bir dilin aracı olarak da kullanılabilmektedir. Tüm bunlar, iletişim için ellerin kullanımının önemini göstermektedir.

İletişim artık sadece insanlar arasında olan bir ihtiyaç değil, insanlar ile makineler arasında olan bir forma da sahiptir. Makinelerin ve bilgisayarların kullanımı çoğunlukla klavye, fare, kumanda ve dokunmatik ekran gibi araçlarla sınırlıdır. Fakat bu araçlar insanlara rahat kullanım sunamamaktadır. İnsan makine etkileşimi çalışmalarının odak noktalarından biri, insanların cihazları daha doğal bir şekilde kullanacakları yöntemler geliştirmektir. Örneğin, ses ile kontrol edilen cihazlar, kumanda ile kontrol edilenlere göre daha basit ve doğal kullanıma sahiptir. Bu noktada, insanların kendi arasındaki iletişimde kullandıkları el jestlerinin cihazlar ile etkileşimde kullanımı da mevcut yöntemlere alternatif olarak ortaya çıkmaktadır.

El jestlerinin makineler tarafından kullanılabilmesi için algılayıcılara ihtiyaç duyulmaktadır. Literatürde el jestlerini tanımlamak için kullanılabilecek kamera, derinlik algılayabilen kamera, eldiven ve elektrotlar gibi farklı yöntemler bulunmaktadır. Görsel tabanlı sensörler ile tanımlama yapan sistemlerle yüksek başarıma sahip sonuçlara ulaşılsa da uygulamada pek çok zorluk yaşanmaktadır. Örneğin ortamın ışığı, görüntü arka planının deseni, el ve parmakların açısı, boyutları ve rengi gibi problemler jest tanıma başarısını düşürmektedir. Ayrıca, görüntünün alınabildiği alanın sınırlı olması da bir kullanım kısıtı yaratmaktadır. Öte yandan, her bir parmağında esneklik algılayan sensörler bulunan bir eldiven tasarımıyla gerçekleştirilen çözümler ile de el jestlerinin tanımlanması mümkündür. Fakat bu eldivenin giyilmesi ellerin rahat şekilde kullanımını engellediği ve hissiyat kaybı yaşattığı için gündelik hayatta tercih edilmemektedir. Elektrotlar yardımıyla alınan ön kol kas sinyallerinin kullanımı el jestlerinin tespiti için alternatif olarak geliştirilen bir yöntemdir. El hareketlerini yapabilmemiz, sinir sisteminden kas hücrelerine gelen komutlar sayesinde olmaktadır. Hücreler kasılma işlemini gerçekleştirebilmek için

enerjiye ihtiyaç duymaktadır ve bunu da ortamdaki maddeleri kullanarak üretirler. Kasılma sırasında meydana gelen bu kimyasal tepkime aynı zamanda elektriksel bir değişime de sebep olmaktadır. Ortaya çıkan bu elektriksel sinyallerin algılanması elektrotlar ile sağlanabilir. Bu elektrotlar bir iğne şeklinde olup doğrudan belirli bir kas grubunun içerisine saplanabildiği gibi, cilt yüzeyine de yerleştirilebilmektedir. Yüzey elektrotları ile sadece yüzeye ulaşabilen sinyalleri toplamak mümkündür, ve bu sinyaller pek çok kas hücresinden gelen sinyallerin birleşimi olarak yüzeye ulaşmaktadır. Medikal bir işlem gerektirmediği için yüzey elektrotları araştırmalarda ve uygulamalarda kolaylıkla kullanılabilmektedir. Giyilebilir cihaz teknolojilerinin gelişmesi sayesinde bu elektrotlara sahip cihazlar daha ulaşılabilir ve daha farklı alanlarda kullanılabilir olmuştur.

Bu çalışmada, Myo Armband cihazı ön kol kas sinyallerini toplamak için kullanılmıştır. Thalmic Labs tarafında geliştirilen, giyilebilir ve kablosuz bir cihaz olan Myo Armband'ın üzerinde bir işlemci ve sekiz adet yüzey elektrodu vardır. Kullanıcılar bileklik şeklindeki bu cihazı koluna giydiği zaman, cihaz kolun etrafına dairesel olarak yerleşmiş sekiz farklı noktadan ölçtüğü sinyalleri Bluetooth ile doğrudan diğer cihazlara aktarabilmektedir. Yazılım geliştirme kitinin sağladığı kolaylık sayesinde Myo Armband mobil robot kontrolü, protez el kontrolü, sanal gerçeklik uygulamaları, sanat projeleri gibi el jestlerinin kullanılabildiği farklı uygulama alanlarına sahiptir.

Bu çalışmada geliştirilen el jestlerini tanıma sistemi için Türk İşaret Dili'ndeki rakamların jestleri veri seti olarak kullanılmıştır. Kullanıcının sağ ön koluna giydiği Myo Armband ile 10 adet rakamları gösterirken ve bir adet dinlenme durumunda olmak üzere toplamda 11 farklı el jesti için kas sinyalleri toplanmıştır. Toplanan sinyaller içerisinden sadece kasılı haldeki kısımlar dikkate alınmış yani geçiş sinyalleri incelemeye katılmamıştır. Bu yüzden geliştirilen sistem statik el hareketleri için uygulanmıştır. Sekiz farklı kanaldan aynı anda sinyaller toplandığı ve toplanan sinyallerin basit eşikleme yöntemiyle sınıflandırılması mümkün olmadığı için el jestlerinin tanımlanmasında makine öğrenmesi tekniklerine başvurulmuştur.

Tez kapsamında önerilen sistem makine öğrenmesi yönteminin aşamalarını takip etmektedir. Buna göre, toplanan sinyallerden ilk olarak anlamlı ve ayırt edici öznitelikler çıkartmak gerekmektedir. Doğrudan ham veri ile çalışmak uygulamanın karmaşıklığını ve hesaplama işlemlerini arttırmaktadır. Ayrıca, ham veride görülemeyen bazı paternler öznitelikler ile ortaya çıkmaktadır. Kas sinyallerinin karmaşık ve gürültülü karakteristiği nedeniyle uygun özniteliklerin çıkartılması önem taşımaktadır. Literatürde bulunan özniteliklerden yedi tanesi baskın ve bu çalışma için uygun bulunmuştur. Bunlar ortalama mutlak değer, varyans, dalga formu uzunluğu, ortalama karekök, Willison genliği, sıfır kesme sayısı ve eğim yönü değiştirme sayısıdır. Toplanan sinyal ilk olarak kayan pencereler yöntemi ile pencerelere bölünüp her bir pencereden bu yedi öznitelik çıkarılmıştır. Bu işlem sekiz elektrodun hepsinde yapıldığı için sonuçta 56 boyutlu bir öznitelik matrisi elde edilmiştir. Bu öznitelik matrisinin boyut sayısı sınıflandırmanın daha rahat yapılabilmesi için azaltılmıştır. Azaltma işlemi için literatürde sıkça kullanılan Temel Bileşenler Analizi yöntemi uygulanmıştır. Bu algoritma ile boyut sayısı 56'dan 15'e düşürülüp elde edilen öznitelik matrisi sınıflandırma algoritmalarını beslemek için kullanılmıştır. Bu çalışma kapsamında dört adet sınıflandırma algoritması ugulanmış ve el jesti tanımlama performansları karışlaştırılmıştır. Bu algoritmalar k-En Yakın Komşu, Karar Ağacı, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma algoritmalarıdır. Her bir sınıflandırma algoritması ilk olarak en uygun

parametrelerini belirlemek için çalışılmıştır. Sonrasında hepsinin en uygun değerine göre elde edilen sonuçlar birbirleriyle kıyaslanmıştır.

Uygulanan ilk algoritma olan k-En Yakın Komşu algoritması, örnek olarak alınan veriye en yakın olan *k* adet verinin hangi sınıfa ait olduğuna bakarak sınıflandırma yapmaktadır. Burada *k* ve uzaklık hesaplama yöntemleri değişken olabileceği için, bu çalışmada farklı *k* değerleri ve farklı uzaklık fonksiyonları için başarımlar araştırılmıştır. Karar ağacı algoritması, eldeki veri setini sürekli olarak iki veya daha fazla dala bölerek sınıflandırmaktadır. Daha sonra yeni gelen örneği, bölme işlemlerindeki kararlara uygun dallardan geçirip sınıfını belirlemektedir. Bu bölme işleminide kullanılan kriterler sınıflandırma performansını etkilediği için farklı kriterler bu çalışma kapsamında test edilmiştir. Destek Vektör Makinesi algoritması ikili sınıflandırıcılardan biridir ve eldeki veriyi optimal bir hiperdüzlem ile en iyi şekilde ikiye ayırmaya çalışır. Bu algoritma farklı kernel metotları için denenmiştir. Ayrıca, bu çalışmadaki problem çok sınıflı sınıflandırma problem olduğu için ikili sınıflandırıcılar ile çoklu sınıflandırma yapılmasını sağlayan yöntemler de burada denenmiştir. Son olarak, Yapay Sinir Ağları algoritması el jestlerini sınıflandırmak için uygulanmıştır. Çok katmalı olarak oluşturulan ağ, gizli katmana farklı nöron sayıları koyularak incelenmiştir. Bununla birlikte, farklı hata hesaplama fonksiyonları ve farklı eğitim fonksiyonları için de algoritmanın başarım sonuçları karşılaştırılmıştır. Önerilen algoritmaların parametreleri yapılan tüm bu denemelerin sonucunda elde edilen en uygun değerlere sabitlenmiş ve sonrasında algoritmaların başarımları birbiri ile kıyaslanmıştır.

Tüm algoritmaları birbiriyle karşılaştırırken yapılan deneyler, geçerliliği arttırmak için 10 kez tekrarlanmış ve ortalaması hesaplanmıştır. Ayrıca, her bir deneyde k katlamalı çapraz doğrulama yöntemi de uygulanmıştır. Bu yöntem veriyi rastgele olarak k adet alt veriye bölüp, her seferinde bunlardan birini test için kalanını algoritmanın eğitimi için kullanılmasına dayanan bir doğrulama yöntemidir. Bu sayede her seferinde algoritma eğitildiği veri setinden farklı bir veri seti ile test edilmiş olur. Elde edilen sonuçlara göre, Karar Ağacı algoritması 0.76 Fskor başarım oranı ile en düşük değere sahip olarak bulunmuştur. k-En Yakın Komşu ve Yapay Sinir Ağları algoritmaları başarımı sırasıyla 0.82 ve 0.85 seviyesine çıkartmıştır. Son olarak, en yüksek başarımlı sınıflandırma oranı olan 0.87'ye Destek Vektör Makinesi algoritması ile ulaşılmıştır. Algoritmalar ayrıca her bir jestin tanımlanması için ayrı olarak da analiz edilmiştir. Buna göre en yüksek başarıya sahip yani daha kolay sınıflandırılabilen jestler '6', '7', '8' ve '9' iken, en düşük oran '2' jesti olarak bulunmuştur. Ayrıca her bir jestin ayrı ayrı tanımlanması bakımından kıyaslandığında da Destek Vektör Makinesi algoritması en iyi sonuçları vermektedir. Ulaşılan sonuçlar, el jestlerini tanımlamak için ön kol kas sinyallerinin kullanılabileceğini ve önerilen sistemin makineler ile etkileşimde kullanılabileceğini göstermektedir.

# 1. INTRODUCTION

In daily life, people regularly use their hands to communicate with other people and interact with objects. Although speaking is a sufficient and rapid way of communication, people naturally emphasize the speech using their hand. Kendon mentions that the information amount which is sent by the human voice decreases, when usage of the hand gestures increases [1]. Also, hand gestures facilitate to understand the speech. For example a host or hostes in the airplane explain the safety guide for passengers using the hand gestures [2]. It makes the guide understanable for hear impaired passengers or foreign passengers who unable to understand the language of host/hostes. On the other hand, interaction covers not only touching the surfaces but also doing some gestures like in communication to give a command in Human Computer Interaction studies [3]. Hence, hand gestures have important role to improve understanding by machines and humans.

Humans interact with technological devices using mouse, keyboard, joystick or touchpad. These input methods restrict the people, for that reason Human Computer Interaction researches focus on creating more natural interfaces between human and computer [4]. Here, the hand gestures arise as an interface for devices. The hand gestures can be static or dynamic [2]. The computational complexity of the static hand gestures is lower than the dynamic gestures. Also, the complexity of the system is determined by the time and shape variability of the gesture [3]. The gestures are user dependent; therefore, recognition of the same gesture can be different for different users.

The gesture recognition problem is not limited to any particular research domain. Researchers from various fields come up with different approaches and use different tools [5]. The approximations of the solutions like electrodes, gloves and cameras to detect the hand gestures have required completly diverse fundamentals. Also, there are various easy-to-use commercial products available in the market [6][7][8][9]. Thalmic Lab’s Myo Armband which is one of the most known wearable devices is

invented as a solution for hand gesture recognition. The Myo Armband is designed to wear on forearm and has eight surface electrodes to provide the signals from forearm muscle. While humans are doing a hand gesture, the skeletal muscles contract with chemical reactions because they require energy. The chemical reactions in the muscle cells generate the voltage difference which is called electromyographic (EMG) signal. The Myo Armband detects the voltage difference via electrodes. The EMG signals collected from forearm have various hand gesture application such as human computer interfaces [10], automatic sign language interpretations [11], remote controls of electronic appliances [12], and smart robotics [13].

Furthermore, hand gestures are often used by deaf people because they express their thoughts and feelings using the sign languages. According to The World Federation of the Deaf, there are 70 million deaf people and 300+ sign languages around the world [14]. However most of the hearing people do not know the sign language; therefore, communication becomes difficult between deaf people and hearing people. To overcome that issue, researchers have worked on various projects which make translation between sign language and speech [15]. Hand gesture recognition is important for correct translation. This thesis contributes the hand gesture recognition by considering the hand gestures of numbers between 0 and 9 in Turkish Sign Language.

A basic EMG based hand gesture recognition problem can be solved with thresholding [16]. For example, if a signal collected with one electrode will be used for on/off control, it can be classified by creating a threshold to the amplitude of the signal. However, when the problem becomes complex such as in case of multiple electrodes and various gestures, more generic solution is necessary. Therefore, machine learning can be used to recognize the hand gestures because the algorithms can automatically infer the underlying pattern of the signal. The learned pattern is compared with incoming signal to predict a gesture.

Machine learning consist of three fundamental step [16]. First, related data should be collected from process. According to problem, it can be labelled/unlabeled and continuous/discrete. Then, a proper learning algorithm is selected to train the system with data. There are assorted machine learning algorithims in the literature. Prediction is the last step of the machine learning. After the algorithm complete to learn, it can proceed with providing information about the new instances. The

machine learning has various application area. Hand gesture recognition is also one of the application field for the machine learning studies.

In this work, the machine learning algorithims are applied to recognize the hand gestures using the EMG signals. The following paragraphs will explain the structure of the thesis.

In Chapter 2, first, the definition of gesture recognition with literature review will be given. The various devices that have used in previous gesture recognition work will be explained with examples from literature. Second, the fundamentals of the muscle physiology will be introduced to understand how the muscle signals are generated by activity of the muscle. The Myo Armband that was used in this work as a source to capture the EMG signal will be presented in third subsection. Also, the proposed data set which consists of the representation of the numbers between 0 and 9 in the TSL and data collection process will be presented in the last subsection.

Chapter 3 begins with the fundamentals and general applications of the machine learining algorithms. According that, the feature extraction methdology will be informed firstly. Extraction of seven features related with the EMG signal will be defined as sub-parts of feature extraction step. Then, the dimension reduction with Principal Component Analysis and the supervised machine learning classification algorithims k Nearest Neighbor, Decision Tree, Support Vector Machines and Artificial Neural Networks will be explained with mathematical expressions. Last, the evaluation metrics accuracy, precision, recall and F score that were used for performance measurement will introduce.

In Chapter 4, the results of the proposed methods will be given. Each classification algorithm first evaluated by different parameters. After the determination of best parameters, the results of four classification algorithm will be compared. Also, recognition performance of each class will discussed in this section.

Last, the conclusion of proposed method will be remarked and the future works will be suggested in Chapter 5.

# 2. GESTURE RECOGNITION WITH ELECTROMYOGRAPHY

Gesture recognition have been getting popular as a part of Human Machine Interaction studies [17]. While humans are communicating, they use speech and gestures naturally; however, the communication between humans and machines required some interfaces. If the interfaces are complicated, people have face difficulty use of devices. The problem can be overcome with usage of the hand gestures which provides people can interact with machines in more natural way.

In this chapter, related works with hand gesture recognition, background of muscle physiology and usage of the electromygraphic signal in gesture recognition will be explained. Last, the proposed gesture set and data collection method will be given.

## 2.1 Hand Gesture Recognition

The word “gesture” can be defined as nonverbal expression of the body which is commonly face and hand [18]. Gesture recognition is the ability of detection and interpretation of human expressions via computational algorithms. Depth sensing cameras, stereo cameras, gloves and different sensors like electrodes are mostly used in gesture recognition studies. All of the devices utilizes different methods to provide input data; therefore, the algorithms are specialized according to solution. The scope of this work covers only the hand gestures, hence examples of devices and studies from the literature related with hand gesture recognition are introduced in this section.

Flex sensors are the one of the well known technique used in hand gesture recognition. In that method, a glove is designed with array flex sensors which are attached to each finger of glove. When it is bended, the resistance of the flex sensor increases. The flex sensor measures the amount of change in resistance; therefore, the location of each finger is estimated using sensor data. For example, in Taiwanese Sign Language recognition project, Liang and Ouhyoung used the DataGlove to

collect the flexion data of 10 finger joints as a posture feature [19]. They also used the 3D tracker to gather orientation, position and motion features. The lexicon of 250 words in Taiwanese Sign Language was executed with 51 postures, 6 orientations and 8 motion in the data set of project. These words were classified by hidden markov models in real-time and 80.4% was achieved as the average recognition rate.


**Figure 2.1:** Glove examples: (a) is used in [20] (b) is used in [21].

Verma et. all contributed the literature with their sign to speech system which is mainly designed to help deaf and dumb people [20]. They created a glove called as Smart Gloves to detect the hand gestures. The system recognizes the sign language gestures and then convert them to voice. They claimed that the system can be achieved recognition ratio of nearly 99% in real time.

Further, hand gestures can be recognized using single camera and colored gloves. The gloves are designed with different colors for each fingers, then the color information makes the finger recognition possible. In [21], Lamberti and Camastra used wool glove where the fingers and palm are colored and rest of glove is black. They segmented the colors with a thresholding method. The location and orientation of the colored part of the hand was used as features and hand gestures were classified with Learning Vector Quantization. They selected 13 gestures and collected 1541 image to validate the algorithm. The algorithm has recognition rate of 97.79%.

Maraqa and Abu-Zaiter aimed to create a Arabic Sign Language hand gesture recognition system with the color gloves [22]. They used 900 image as a training set which consists of 30 different hand gestures. Thirty features were extracted from each image to feed the classifier. The images were classified with two different type

of neural network architecture which are Elman Network (EN) and Fully Connected Recurrent Network (FCRN). The recognition rate of 95.11% and 89.66% were achieved for EN and FCRN respectively.

Also in [23], a glove was designed with locating the different shape of reflectors for fingers, palm and back of the hand. By the help of coaxial infrared illumination, the difference between reflectors and background was maximized. The binarization and erosion process were applied to images. The shape of the patterns was detected using the polygon approximation function. Hu moments and convex hull were used as hand gesture descriptor. Although there were not indicated accuracy rate, they claimed that the recognition rate of the selected hand gestures is high.

In hand gesture recognition works, usage of the depth cameras has become more popular in recent years because they provide information about spatial location and orientation of hand. Number of researches have been increasing with availability of the commercial products such as Kinect which provides depth sensor data. For example, Du and To used the Kinect to recognize the hand gestures which were the numbers from 0 to 5 [24]. In their method, first, pre-processing was applied to scale and eliminate the back-ground. Then, the algorithm detected the hand contour and extracted the number of convex and concave points related with the hand gesture. The classification was realized based on the number of concave and convex points. According to results, the algorithm has correct classification rate of 94.4%.

Moreover, Dinh et all. presented a hand number gesture recognition method using depth images [25]. The algorithm extracted the depth silhouette of the hand parts by removing the background. After, open or close states of each finger was taken to create rules and features. The hand number sing gestures were classified by random forests algorithm. The experimental results showed that the system has recognition rate of 97.8%.

Leap Motion is another depth sensor; however, unlike Kinect, it provides the depth data frames including finger position, hand position, rotation, scaling data and so on. In [26], Lu, Thong and Chu proposed a method to recognize the dynamic hand gestures using Leap Motion.The features extracted from data frames were based on palm direction, palm center position, pal normal and fingertips positions. Hidden conditional neural field algorithm was used as classifier. Two different hand gesture

datasets were collected to measure the performance of algorithm: MSRGesture3D which consists of 12 American Sign Language gestures and Handicraft-Gesture which included 10 pottery skills gestures. The accuracy were 89.5% and 95.0% for MSRGesture3D and Handicraft respectively.

Although visual based hand recognition systems provide high accuracy, they confront some challenges. In [2], the challenges are listed as five main problem which are. variation of illumination conditions, rotation problem, background problem, scale problem and translation problem. Therefore, the light conditions, hand and fingers position with respect to camera and properties of the hand like skin color and size affects the recognition of the hand.

Another impressing method to hand gesture recognition was suggested in [27]. Wen et al. used the smartwatches which has already integrated the rotation, gravity, accelerometer, gyroscope and linear accelerometer sensors. The system uses the data from only three of them (the accelerometer, gyroscope and linear accelerometer sensors) and calculates the 7 statistical features. The Support Vector Machine (SVM), the Naive Bayes classifier, the Logistic Regression and the k-NearestNeighbors methods were chosen as classifier. The system tries the all classifier and then selected the best fitted one. Five gestures (Pinch, Tap, Rub fingers, Squeeze, and Wave) were performed in the work for 10 different participants. According to results, the Support Vector Machine and Logistic Regression were obtained as best classifiers and the mean f1-score across all participants is achieved as 87%.

### 2.2 Muscle Physiology

Electromyography (EMG) is a technique of recording and study of myoelectric signal that creates the muscle tension [28]. That myoelectric signals are produced by electrical and chemical activity of skeletal muscle. At the rest position, a normal muscle is known as electrically silent and generates signal with low amplitude [29]. However, when the contraction starts with nerve stimulations, the electrical signal becomes detectable both under isometric and isotonic conditions. EMG is an abbreviation for both electromyography and electromyogram which is the graphical representation of electromyographic signals [36].

Skeletal muscle consists of many muscle fibres (myofiber) which is also known as a muscle cell [38]. Muscle fibres (or fibers) is excited by alpha motoneuron that conducts the signals from spinal cord to multiple dendrites of the neuron [30]. That smallest functional unit of neural controlled muscular contraction is called Motor Unit [34]. In Figure 2.2, myofibers and motor neuron can be seen.

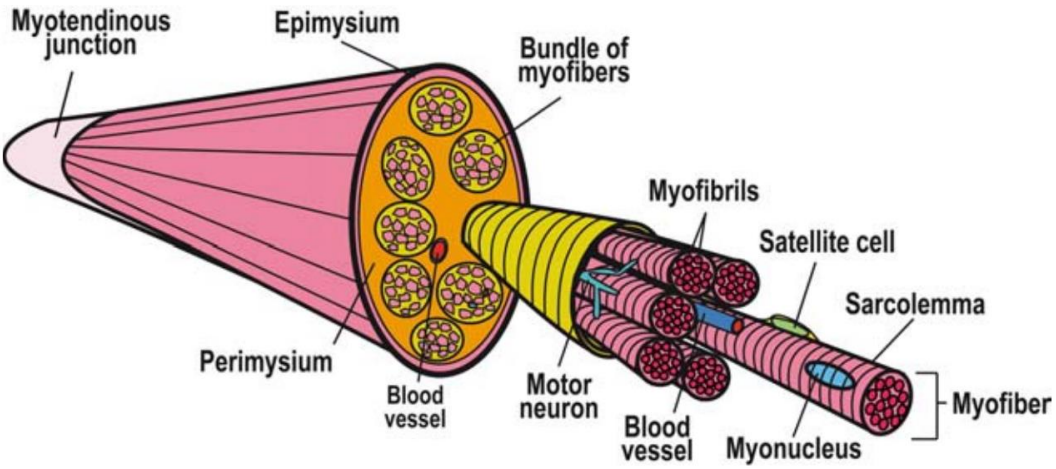



**Figure 2.2:** Structure of skeletal muscle [37].

Each myofiber which includes many myofibrils are covered by sarcolemma. Sarcolemma is a thin and elastic plasma membrane of muscle cell. Semi-permeable property of sarcolemma adjusts the ionic equilibrium between inner and outer space of the cell [31]. That adjustment is called as ion pump which is realized by transmission of the sodium and potassium elements. The ionic difference results in an electrical potential. That potential is generated even in resting position and it is about ~-80 mV which means the outer cell charge is higher than inner [31]. When the motor endplates in motor unit are activated, Na+ ions flows into the cell. If a certain threshold level is exceeded within the Na+ influx, the membrane is depolarised. Depolarisation produces about +30 mV electrical Action Potential. Further the depolarisation, inside the muscle fiber, spreads along the muscle fiber in both direction through a tubular system [32]. This potential helps the flowing of calcium ions to intra-cellular space. The contraction of muscle is finally realized with chemical process using that calcium ions. If the excitation with Na+ can not exceed the threshold, that does not result with contraction. Repolarization occurs thanks to ion pump when the contraction ends up and results with Afterhyperpolarization. At the end, it goes back to rest position without any signal requirement.

The EMG signals are generated by the action potentials at the muscle fiber membrane [32]. The depolarization - repolarization cycle moves and creates an electrical dipole along the muscle fiber. Motor endplates of alpha motoneuron is connected many muscle fibers with different locations; therefore, electrical cycle of each muscle fiber has distinct initial time and magnitude. However, the obtained signal from motor unit includes the combination of all innervated fibers within the motor unit. It is called a Motor unit Action Potential (MUAP) which means, in other words, the superposed signal of the activated muscle fibers by a motor unit [33]. Generation of MUAP is shown in Figure 2.3.

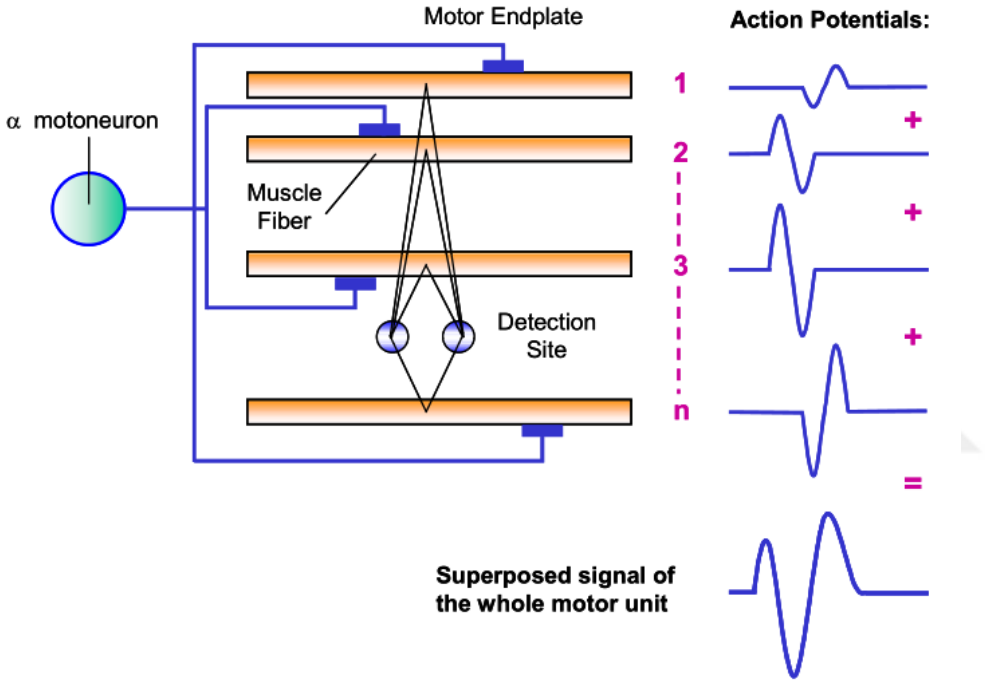



**Figure 2.3:** Generation of motor unit action potential [31].

Monitored EMG signals with surface electrode composed of multiple MUAP which are also superposed and has symmetric distribution of positive and negative amplitudes [31]. The shape of EMG signal are affected by two important factors: Recruitment and Firing Frequency of MUAPs [34]. These factors also manage the contraction process and the force of the muscle. Although EMG signal can not prove the real firing and amplitude characteristics because of the skin layer and connective tissue of human, it is basically considered as reflection of the recruitment and firing characteristics of muscle.

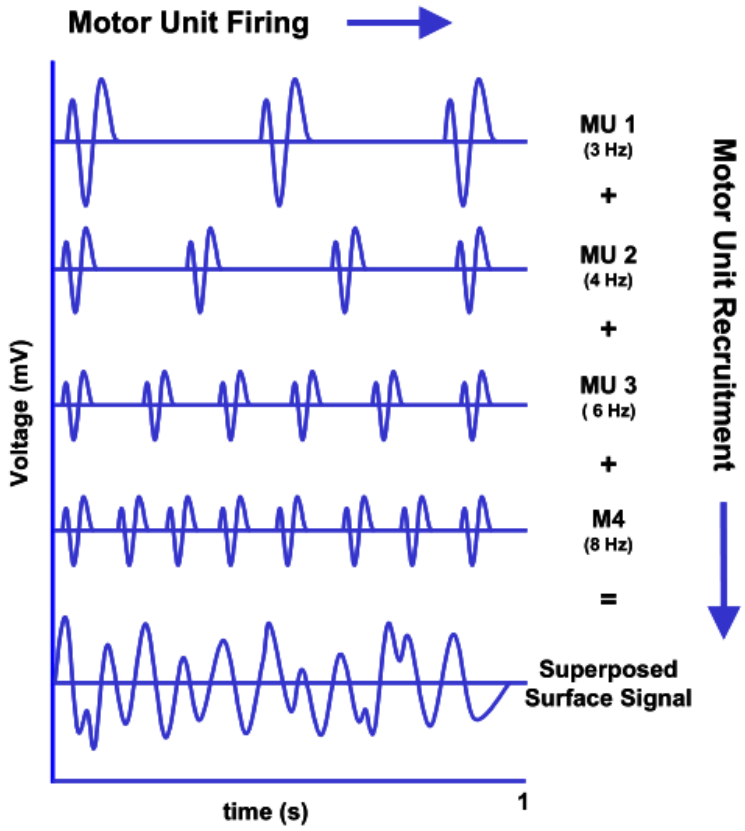



**Figure 2.4:** Generation of superposed surface signal [31].

The superposed and unprocessed signal of the muscle is called as raw EMG signal. The scope of the thesis covers only the muscle of forearm the part of the upper limb between the wrist and the elbow. Figure 2.4 shows the EMG signal from the extensor digitorum muscle. According that, the EMG signal has not significant activity during the relaxed position of muscle. It has a noisy characteristics and provides a baseline to detect the muscle activity. During the active contraction phase, raw EMG signals has random shape spikes and can not be reproducible in exact shape [30]. Because that spikes are generated by superposition of multiple motor units signals.

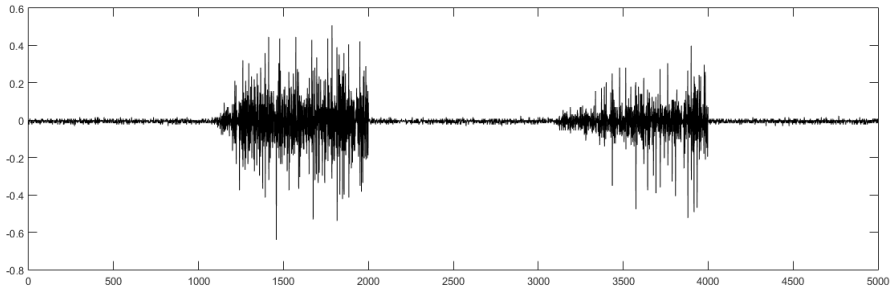



**Figure 2.5:** Raw EMG signal from two times contraction of the extensor digitorum muscle.

Generally, there are two types of electrodes that are used in EMG process: fine wire and skin surface [33]. Fine wire electrodes are used in invasive method to obtain samples from deeper muscle layers. Thin and flexible fine wire electrodes are inserted into the muscle by help of a needle and provide the signal from more specific space [34]. On the other hand, skin surface electrode is non-invasive and more useful for various application because of its easy to implement characteristics. These electrodes sense only the signals that can reach the skin surface. Also it provides from wider area of the muscle insead exact muscle fibers. Surface electrodes can also be monopolar and bipolar configuration [35]. In monopolar configuration, one electrode is located on the center of muscle and one electrode is placed on the reference point of the body. In bipolar configuration, as a more common, two electrodes are placed on the center of muscle within 1-2 cm from each other and one electrode on the reference point of the body. Difference between two electrodes on the muscle is used to process and this configuration provides more signal to noise ratio by eliminating the common noise in two electrodes. In the literature, the surface EMG signals is abbreviated as sEMG. The sensor used in this work is a type of bipolar surface electrode; therefore, the signals are surface EMG signals. The terms EMG signals and sEMG signals are used interchangeably because there is no invasive technique in the scope of the work.

The amplitude range of raw sEMG signal are fractions of a microvolt to thousand microvolts [32]. The raw sEMG signal has also constant frequency range between 5 and 500 Hz; however, mostly seen 20 and 150 Hz band. Although it can be influenced by noise from different sources such as the equipment, the unfixable positions of the electrodes on skin surface and the randomness of the motor unit firing patterns, naturally the EMG signal has nearly zero mean voltage level [40].

### 2.3 The Myo Armband

The Myo Armband is launched by Thalmic Labs on 2013 [9]. It is a wearable and bluetooth based wireless device that contains eight EMG sensor and nine axis Internal Measurement Unit (IMU) including gyroscope, accelerometer and magnetometer. The device can provide feedback to user via LED lights and vibrations. Also there is a ARM Cortex M4 Processor in the core of the device [41].


**Figure 2.6:** The Myo Armband. (a) number of channels (b) worn on forearm.

Users wear the Myo Armband on the wrist as shown in Figure 2.6 and gather the data from the forearm muscle activity. The data measured by the Myo Armband can be obtained using Software Development Kit (SDK) of device via bluetooth. The Myo SDK written in C++ has a easy-to-use library, libmyo, to interact with the applications.

The Myo Armband fundamentally provides Spatial and Gestural data. Spatial Data, related with the position, orientation and movement of the device, is collected by the IMU that has 50 Hz sampling rate. The EMG signals are measured by the surface electrodes to provide gestural data. The Myo Armband can recognize 5 built-in hand gestures; on the other hand, it gives the raw EMG data. The EMG data is measured from 8 channel and the locations of the electrodes can be seen in Figure 2.6. The EMG data is collected with sampling rate of 200 Hz. There is not any information about the unit of the data in the Myo SDK documentation [42]; however, it is known that the data is converted to 8-bit values within range between -128 and 128.

## 2.4 Literature Review about Myo Armband

sEMG signals are created by activity of the muscle and collected via surface electrodes. Although EMG is mainly used in medicine, there are various field of application such as ergonomy, sports, art [43] and entertainment [44]. In this chapter, some of the sEMG studies will be introduced.

First, Sueaseenak et all aimed to compare the performance of the Myo Armband and BIOPAC which is a conventional EMG system [45]. The collected data from both devices were investigated in terms of quality of EMG signal and class-separability of

EMG feature. Signal-to-noise raito, total harmonic distortion and power density spectrum were used to measure the quality of EMG signal. Davies-bouldin and scattering criterions were parameters to evaluate the performance. Six hand gestures were performed in experiment. The results showed that the Myo Armband is more robust in class-separability and has provides ideal shape in frequency spectrum. Also, they mentioned that the design with dry electrodes facilitates the usage of Myo Armband in researches.

The Myo Armband is capable to recognize default 5 hand gestures which are fist, open hand, wave in, wave out and pinch. In [46], Benalcazar et. all measured the accuracy of the default system and that was found as 83%. After, they proposed a recognition system. According that, first the signal was rectified and then low-pass Butterworth filter was applied. Obtained signal was classified with dynamic time wrapping and k-nearest neighbor algorithims. The system was tested with same hand gestures and 86% recognition rate was achieved.

Prosthetics designed for the amputees helps to accomplish the daily requirements of the people. Although, they are passive in early age, nowadays people uses the controllable prosthetics thanks to the contribution of biomedical studies. sEMG is one of the input method to activate the prosthetics. In [47], an anthropomorphic prosthesis hand was actuated by sEMG. Seven hand gestures for grasping were classified by artificial neural networks. The wavelet packet transforms were selected as features. The experiments were realized for two different data set which were collected from one person and eighteen persons. The results showed that classification is more accurate when the system is trained for one subject instead to generalizate.

Gestures of non-dominant hand of classical music orchestra conductor were classified in [48]. Moving averaging window, full wave rectification of interelement difference and circular harmonic coefficients were chosen as features and Artificial Neural Network (ANN) was used as classifier. The system was trained using fourteen gestures but during performance the six of them including rest position were performed by conductor. The accuracy of the six gestures was 96.9% and it was 100% for gesture of rest.

In [49], recognition of 20 stationary letter gestures from Brazilian Sign Language alphabet with surface EMG was performed. They collected signals and then applied fullwave rectification. Feature vector was created with computing the mean of each 50 data points. SVM algorithm with RBF kernel was used to classify the gestures. For each class, it was achieved more than 96% success rate in the training; however, when they apply the classifier in real time, the performanceof each class varied between 4% and 95%. They suggested that if the gestures are performed same as training set, the classification become correct; however, slight changes in position fails the result.

Another application of sEMG signal is CoMBaT which is a training system for badminton players [50]. The system was designed to evaluate the physical characteristics of muscular effort and arms' swing while the player is shooting. Also haptic feedback is given to player in real time. The Myo Armband provides EMG and IMU data for the system and Dynamic Time Warping algorithm was used to compare of experts shot and trainees shot. They also give a visual feedback of the performance.

Moreover, detection of the poses in the game Rock-Paper-Scissors is one of the entertainment application using sEMG [51]. They sampled the recorded signal with different sampling rate and used these points as feature. The decision tree algorithm was used to classify the received signal and the correct rate of the system was around 78%.

Another game based on hand gestures is Hand Cricket which is a common game in south Asia. The game is played with two people and require to perform pre-defined five hand gestures. Krishnan et. all offered a system to recognize that hand gestures with Myo Armband [52]. The collected signal pre-processed with DC Offset removal and notch filter. Then the features which were simple square integral, maximum, minimum, mean frequency and mean power were extracted. The linear SVM algorithm was used as classifier and hold-out method was applied for validation. At the end, accuracies of the recognition were obtained as 92.4% and 84.27% for player 1 and player 2 respectively.

In [53], Lun et. all. used the Myo Armband to navigate a three-wheeled omni-directional mobile robot. They propsed two control method using EMG and using

both EMG and IMU values. In first method, ”stop”, ”go forward”, ”rotate clckwise” and “rotate counterclockwise” commands were given with the four basic hand gestures which were fist, spread fingers, wave left and wave right that have 98.17%, 95.68%,99.45% and 99.82% gesture recognition rates, respectively.

### 2.5 Handled Data Set

Sign language is one the way to communicate especially used in community of deaf people. Like the spoken language, the sign language also varies in countries. It is known that there are more than 300 different sign language around the world [14]. In order to contribute to gesture recognition works using EMG signals, the gestures of the numbers in Turkish Sign Language (TSL) was selected as a dataset. The gestures of the numbers between 0 and 9 are obtained from [54] and can be seen in Figure 2.7.

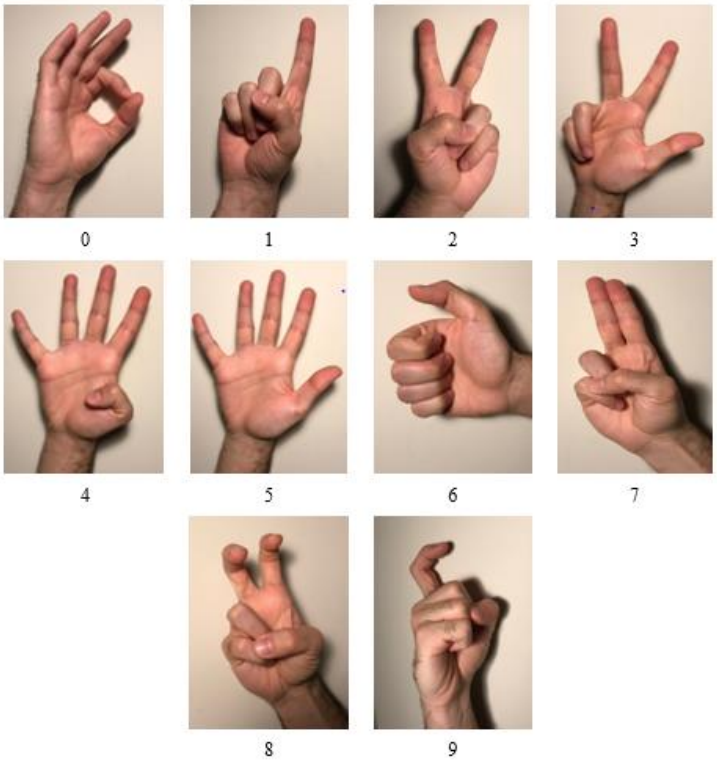



**Figure 2.7:** Handled gesture set which are numbers between 0 and 9 in TSL.

The EMG signals were collected from forearm with the Myo Armband. Subjects can easily wear the device but there is a crucial point that the device has 8 same electrodes located circularly; thus, the provided data depends on how you wear it. Hence, subjects should wear the device by considering the direction and orientation.

A guide in [55] is used as a reference while wearing the Myo Armband to have same calibration.

The Myo Armband allows to interact with the SDK that is written in C++. However, the Matlab scripts were used in that project; therefore, it was required to conversion from C++ to Matlab. Thanks to Mark Tomaszewski who developed Matlab Mex Wrapper using the C++ bindings of official Myo SDK, the device is available to interact within Matlab [56]. The only difference of the mex wrapper is that the data normalized [-1 1] range instead [-128 +128].

The Myo Armbands has sampling frequency of 200 Hz for 8 EMG channel. The data collected from one subject includes 10 gestures related with different numbers in TSL and 1 gesture for rest position; thus, total 11 gestures for each experiment. While collecting data, each gesture was performed for 15 seconds which consists of first 5 second for resting, 5 seconds for holding in gesture and last 5 seconds for resting again. The resting was added to before and after the movement to prevent the muscle fatigue. However, the considered part of the raw signal contains the part of while user holding in performed gesture, which was 5 seconds in the middle. The experiment was repeated 5 times to improve the data validation. Consequently, the collected and labelled data has 55000 data points including 5 seconds, 200 Hz, 11 gesture and 5 times was collected from one subject.

$$V = [\text{rawData}]_{55000\text{x}8} \tag{2.1}$$

# 3. HAND GESTURE RECOGNITION SYSTEM

Machine learning, is also called automated learning, is a subfield of Artificial Intelligence that focuses on methods that makes available computers to learn [57]. Learning is method of obtaining expertise or knowledge from experience [58]. This experience is the input of the algorithms and is known as training data. The output is the expertise. In other words, machine learning can be defined as computational methods using available past information to provide better performance or efficient and accurate predictions [59]. The prediction is required for many tasks; therefore, learning algorithms have various applications such as credit scoring, recommendation systems, stock trading, fraud detection, drug design [69]. The application areas spread to new fields day by day. Each application has different learning algorithm and there are diverse categorization techniques. For example, the problems of machine learning can be categorized as classification, regression, ranking and clustering [59]. In classification, each instance is assigned a category; whereas, regression produces a real value for each instance. In ranking, instances are ordered according to some criterion and in clustering, instances are seperated into homogeneous regions.

Another general categorization of the learning algorithm is the feedback availability of the past information such as supervised, unspuervised, semi-supervised and reinforcement learning [60]. Supervised learner is trained using set of labelled instances, in contrast unsupervised learner uses set of unlabelled instances. In semi-supervised learning, set of both labelled and unlabelled instances are used. Reinforcement learner is fed by past good actions.

The subject of this thesis is consist of classification in the supervised machine learning; therefore, the terms machine learning and supervised machine learning were used interchangeably.

A machine learning algorithm starts with obtaining feature matrix from the raw data to make a good discriminant between categories. If the dimension of feature matrix is

large, it can be diminished with dimension reduction algorithms. After, the learning algorithm is trained using the feature vector. When the learning is completed, the system can predict the labels for new instances. The process can be seen in Figure 3.1.

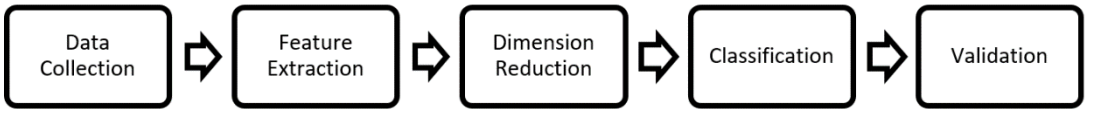



**Figure 3.1:** Classification algorithm pipeline.

In the following subsections, the stages of the machine learning will be explained in details. The stages consist of feature extraction, dimens reduction, classification and validation.

### 3.1 Feature Extraction

Feature extraction, the first and most important step of gesture recognition, is a process of getting meaningful and distinctive information from the raw data [61]. It is often used in machine learning because working with directly raw data increase the complexity of implementation and processing cost. Also, some hidden patterns which can not be visible in a single data point can be obtained in features. EMG signal has very low amplitude and noisy characteristics; therefore, feature extraction is required to find the patterns in the signal [62]. Processing of the raw data is generally based on mathematical functions. In the literature, there are lots of different EMG feature extraction methods that are categorized in three domains: time domain, frequency domain and time-frequency domain [63]. Time domain features has low complexity and low computational cost, so it is used widely for muscle activity detection. Frequency domain features is generally used to detect the muscle fatigue and neural abnormalities [64]. Features in time-frequency domain are analysis of the variation of signal frequency over time such as wavelet packet transformation and discrete wavelet transformation [65]. Although there are lots of features in literature, seven time-domain features are found as dominant for this work. Mathematical expressions of the seleceted features will be explained in following subsections. Also, in this work, sliding window approach was applied before the feature extraction and, features were obtained from each window.

### 3.1.1 Sliding window approach

To catch and extract the features from the meaningful part of the signal, sliding window approach was applied to raw data. According to [66], $M$ sequential data point from raw signal was selected as a window, where features were obtained from. Then, the window is slided $N$ data points and features are obtained for new window. This process continues recursively until last data point is included. Each window is called as segment and the process can be seen in Figure 3.2.

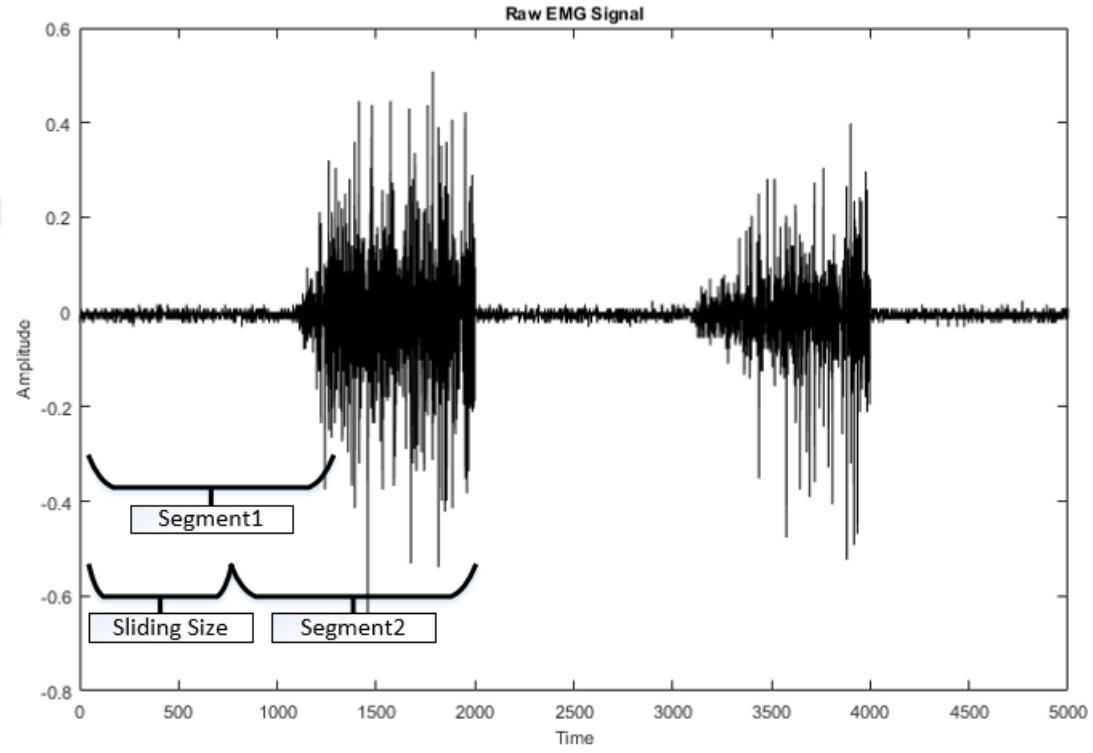



**Figure 3.2:** Example of sliding window approach

The length of a segment and sliding are critical values to make successful classification from recorded data. The sliding size is related with how much you frequently want to take segment instance from recorded data. If $N$ is lower, the feature vector will includes too many redundant values. On the other hand, higher values of $N$ cause failure to catch the related part of the signal. According to experimental results in this work, $N$ was selected as 20. Further, the length of segment, $M$ determines the response time in the real time application. To select an optimal value, the sizes between 10 and 200 were tested with consideration of time and accuracy of classification. Results show that the higher values of $M$ cause latency but lower values decrease the performance of classification. Therefore, $M$ was selected 40 as a optimal value. When the sliding window process was completed

for one class data, the length of the raw data was decreased to the number of segments, $s$, which can be calculated as,

$$s = \frac{L}{N} - 1 \tag{3.1}$$

where $L$ is the length of raw data of one class. Our data set included 11 gesture class; therefore, the length of feature matrix will be 2739.

Each feature in the following subsections was considered as a dimension of the feature vector. Also, the features were extracted for each separate channel. Hence, the feature vector, $F$, will have 56 dimensions (8 channel x 7 features) at the end of the feature extraction process.

$$F = [\text{featureValues}]_{2739\text{x}56} \tag{3.2}$$

### 3.1.2 Mean absolute value

First, one of the most popular feature is selected. Mean Absolute Value (MAV) is calculated by averaging the absolute values of the EMG signal amplitudes. The definition of MAV is,

$$Y = \frac{1}{N}\sum_{i=1}^{N}|x_i| \tag{3.3}$$

### 3.1.3 Variance

The variance is a measure of the changing of the signal amplitude, that is defined as,

$$Y = \frac{1}{N-1}\sum_{i=1}^{N}({x_i}^2) \tag{3.4}$$

### 3.1.4 Waveform length

The Waveform Length is a measure of the cumulative variations and can be expressed as,

$$Y = \sum_{i=1}^{N-1} (|x_{i+1} - x_i|) \tag{3.5}$$

### 3.1.5 Root mean square

The Root Mean Square is another well-known feature that is the square root of the sum of second power of each amplitude. The mathematically defined as,

$$Y = \frac{1}{N} \sqrt{\sum_{i=1}^{N} {x_i}^2} \tag{3.6}$$

### 3.1.6 Willison amplitude

This feature counts how many times the contiguous time instances in signal exceeds the predetermined threshold. Definition of Willison Amplitude is shown in equation 3.7.

$$Y = \sum_{i=1}^{N-1} f(|x_{i+1} - x_i|) \tag{3.7}$$

where,

$$f(x) = \begin{cases} 1 & if\ x > thr \\ 0 & otherwise \end{cases} \tag{3.8}$$

The threshold is chosen by looking the signal levels in contraction and at rest positions. It was experimentally obtained as 0.2 for our case.

### 3.1.7 Zero crossing

Zero Crossing (ZC) counts that how many times the signal crosses zero. A threshold (thr) needed to minimize the noise is applied to contiguous time instances. The ZC is calculated as,

$$Y = \sum_{i=1}^{N-1} f(x_i, x_{i+1}) \tag{3.9}$$

where,

$$f(x_i, x_{i+1}) = \begin{cases} 1, & if\ (x_i > thr\ AND\ x_{i+1} < thr) \\ & \quad OR\ (x_i < thr\ AND\ x_{i+1} > thr) \\ 0, & x \geq otherwise \end{cases} \tag{3.10}$$

The threshold was experimentally found as 0.3.

### 3.1.8 Slope sign change

The Slope Sign Change (SSC) is a measure of changing the sign of the slope of the signal. The mathematical expression of SSC is,

$$Y = \sum_{i=2}^{N-1} f(x_{i-1}, x_i, x_{i+1}) \tag{3.11}$$

where,

$$f(x_{i-1}, x_i, x_{i+1}) = \begin{cases} 1, & if\ (x_i < x_{i+1}\ AND\ x_i < x_{i-1}) \\ & \quad OR\ (x_i > x_{i+1}\ AND\ x_i > x_{i-1}) \\ 0, & otherwise \end{cases} \tag{3.12}$$

## 3.2 Dimension Reduction

Dimension reduction is a process of reducing the number of variables in the feature vector set. In the pattern recognition, choosing the number of inputs has great importance to determine the time and space complexity of the classifier [60]. The classifier can also capable to select the dominant features; thus, this is an optional step; however, there are some benefits of the feature reduction such as reducing memory and computation, saving cost, extracting knowledge which means getting clear idea about data and helping generation of more robust simpler models [67]. These reasons make the dimension reduction necessary as a separate step. Although there are various methods such as Linear Discriminant Analysis, Factor Analysis and Multidimensional Scaling, in this study Principal Component Analysis (PCA) was applied; therefore, the background of PCA will be explained in the following.

PCA is well-known and very common unsupervised dimension reduction technique which does not use output labels. PCA is the name of the transformations from the

possibly correlated inputs to new uncorrelated and orthogonal linear subspace vectors called principal components [68]. The projection of input matrix $(x)$ on a direction of principal component vectors $(w)$ can be defined as,

$$z = w^T x \tag{3.13}$$

where each column of the $w$ shows a principal component vector ($w = [w1\ w2\ ...\ wn]$). PCA aims to find maximized variance in the input matrix; therefore, the direction of the principal component is critical. Maximum variance can be found via the first principal component which is denoted by $w_1$ vector. To obtain a unique solution it is chosen a unit vector $\|w_1\| = 1$. If covariance is defined as $cov(x) = E$, then the variance for first principal is calculated [67].

$$var(z_1) = w_1^T E\ w_1 \tag{3.14}$$

The constraint variance maximization problem can be defined as a Lagrange problem like in equation 3.15,

$$\max_{w_1} w_1^T E w_1 - \alpha(w_1^T w_1 - 1) \tag{3.15}$$

If it is assumed that the derivative of the equation 3.15 with respect to $w_1$ equals to 0, then the solution gives that $w_1$ is the eigenvector of $E$ and $\alpha$ is the corresponding eigenvalue.

$$2Ew_1 - 2\alpha w_1 = 0 \tag{3.16}$$

Therefore, the maximum variance is obtained by selecting the eigenvector with the largest eigenvalue $\lambda_1 = \alpha$. Then, $d$ dominant eigenvectors can be found by the similar way; however, there is one more constraint that is orthogonality. According that each principal component should be orthogonal. If the constraint is added to function aimed to maximize [60],

$$\max_{w_d} w_d^T E w_d - \alpha(w_d^T w_d - 1) - \beta(w_d^T w_1 - 0) \tag{3.17}$$

Again, if the derivative of equation 3.17 with respect to $w_d$ is setted equal to 0, then it is obtained that $w_d$ should be eigenvector of $E$. Therefore, $d$ principal components

which provide maximum variance directions can be calculated with first $d$ eigenvectors that are sorted the descending order eigenvalue [67].

In other words, some of the eigenvectors have little contribution to variance so they can be eliminated. Then the rest $d$ dimension explains more the data. The eigenvalues are sorted in descending order and proportion of variance ($pov$) can be applied to determine the $d$ value. This can be formulated as [60],

$$pov(d) = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_{\mathrm{d}}}{\lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_k} \tag{3.18}$$

where $k$ is the original number of dimension.

In this study, at the end of the feature extraction, the feature matrix $F$ had 56 dimensions; hence, PCA was applied to reduce the dimension. To determine the $d$ value, eigenvalues and eigenvectors of $F$ were calculated. $pov$ for each $d$ can be seen in Figure 3.3.

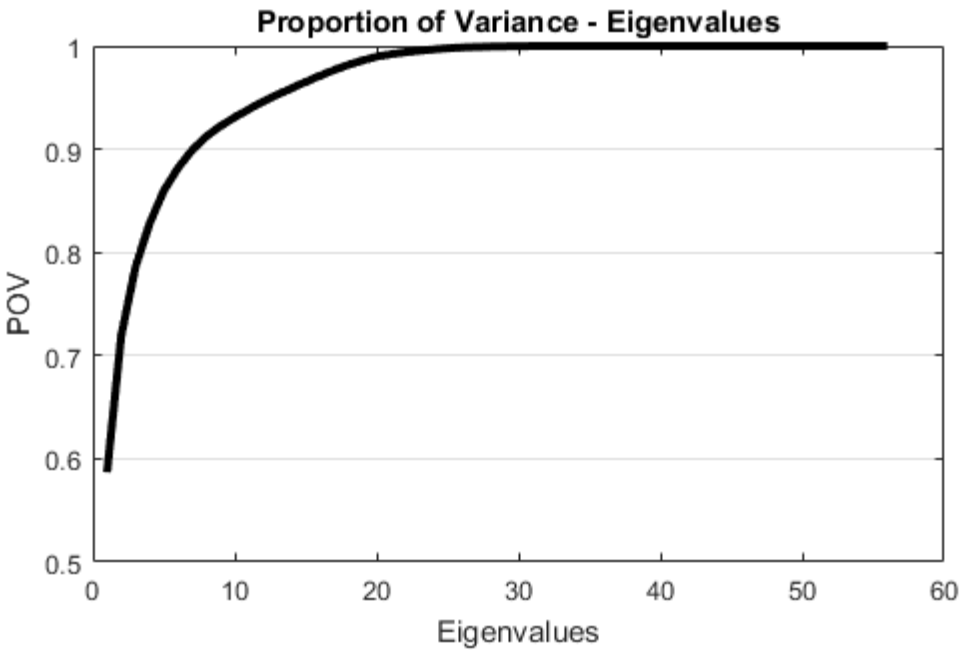



**Figure 3.3:** POV-Eigenvalues relationship to reduce dimension.

A threshold was chosen to determine the $d$ value. According that, the first 15 eigenvalue explain about 96 percent of the variance hence d was chosen as 15. The dimension of the feature vector was reduced from 56 to 15 at the end of the process.

$$F = [F]_{2739\mathrm{x}15} \tag{3.19}$$

### 3.3 Classification

In machine learning, classification is the process of discriminating the data in categories [69]. Feature matrix or raw data can be used as an input of the algorithms. A classification method is chosen to learn how to classify the data. There are 11 hand gesture in our problem; therefore, it is called as multiclass classification problem.

According to scope of the thesis, four of the supervised classification algorithms, k-Nearest Neighbor Algorithm, Decision Tree Algorithm, Support Vector Machine Algorithm and Artificial Neural Network Algorithm will be explained in the following subsections.

#### 3.3.1 k-Nearest Neighbor Algorithm

k-Nearest Neighbor (kNN) algorithm is one of the oldest and simplest method for classification [70]. It is non-parametric instance-based lazy learning algorithm. Each labelled instance in the training set is stored in the memory. $k$ is the number of the neighbors for a point and much smaller than number of sample size, $N$. When a query point $x_0$ arrives, the algorithm finds $k$ training points that are closest to $x_0$ [60]. Then, the majority voted class from these neighbors is chosen as an output class.

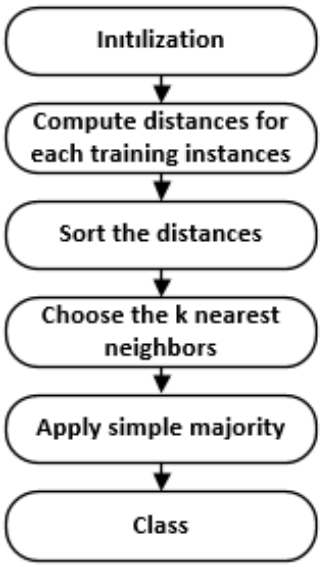



**Figure 3.4:** Flowchart of kNN.

Distance measurement method are critical variables of the kNN algorithm because this method determines which points are selected as neighbors [71]. There are various method to calculate the distance and some of the distance metrics are listed in below [72]:

- Euclidean Distance

$$D(x, x_i) = \sqrt{\sum_{i=1}^{k} (x - x_i)^2} \tag{3.20}$$

- City Block Distance

$$D(x, x_i) = \sum_{i=1}^{k} |x - x_i| \tag{3.21}$$

- Minkowski Distance

$$D(x, x_i) = \sqrt[p]{\sum_{i=1}^{n} |x - x_i|^p} \tag{3.22}$$

- Chebyshev Distance

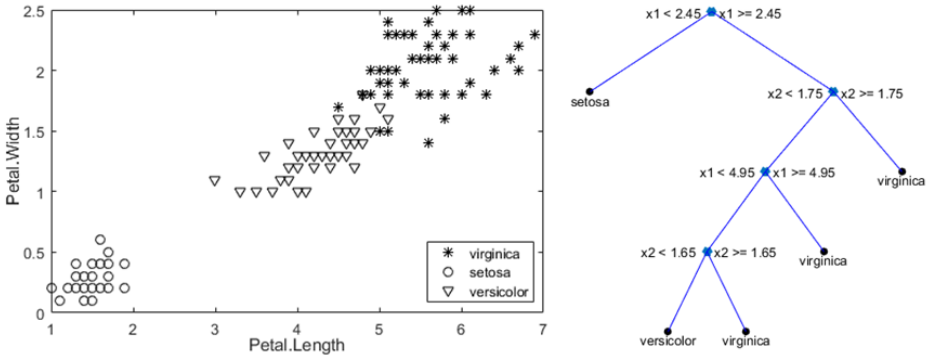
$$D(x, x_i) = \max_i \{|x - x_i|\} \tag{3.23}$$

Also, selection of *k* value is the another important factor in the algorithm. If the *k* equals 1, the algorithm is simply called nearest neighbor classifier. Higher values of *k* has higher bias and less precise; on the other hand, lower values of *k* cause the overfitting problem and become more sensitive to noise. Therefore, *k* determines the degree of smoothing. Generally, *k* is chosen as odd number [60].

In this work, the kNN algorithm will be tested for Eucledian, City Block, Minkowski and Chebyshev distance metrics and increasing *k* numbers from 1 to 10. The results will be given in section 4.1.

### 3.3.2 Decision Tree Algorithm

Decision Tree (DT), a nonparametric learning algorithm, is tree formed and sequential rule based classifier [77]. It basically consists of three different node type. Rooted tree is the first type of node and it initializes the tree. The result of tree which refers the class or probability is called as leaf node. All other nodes between rooted node and leaf node are internal decision nodes that have one incoming edge. Internal nodes of decision tree divide the input data two or more sub-data using some attributes of input data. If the algorithm uses only one input attribute (dimension) to make a decision, it is called as univariate tree; otherwise, it becomes multivariate tree [60]. The separation remains as a recursive manner and each internal node navigate the sub-data from top to down of the tree until a stopping condition is satisfied. For example, using the petal values of Irish Data Set from the UCI machine learning repository [73], a decision tree can be generated as in Figure 3.5




(a) Labelled Data (b) Tree View

**Figure 3.5:** An example of decision tree algorithm

Internal nodes split the data using some top to down algorithms such as Iterative Dichotomiser 3 [74], C4.5 [75], CART [76]. The aim of algorithms is minimizing the classification error with choosing the best decisions. One of the goodness measure of decisions can be calculated with purinity measure. After division, if all the sub-data belong the one class, it is known as pure. There are some methods to calculate the impurity such as Gini's Diversity Index ($gdi$) and *entropy*.

Entropy [77],

$$entropy = -\sum_{i=1}^{C} p(i) \log_2 p(i) \tag{3.24}$$

Gini's Diversity Index [78],

$$gdi = 1 - \sum_{i=1}^{C} p^2(i) \tag{3.25}$$

where $p(i)$ refers the probability of the class $i$ and $C$ is the number of classes. Entropy and gini are expression of homogeneity or purity of the data. Each node in the decision tree splits the data to all attributes and calculate the impurity gain ($\Delta I$) for each attribute. The impurity gain ($\Delta I$) is calculated as [79],

$$\Delta I = P(T)i_t - P(T_L)i_{tL} - P(T_R)i_{tR} \tag{3.26}$$

where, $P(T)$ denotes the probabilty with related the number of samples and $i_t$ is the impurity of the node $t$. $L$ and $R$ indices refer the left and right child nodes respectively. It is required to choose only one of the attribute to create a univariate tree. The attribute with maximum impurity gain should be chosen for the current splitting criteria. The algorithm recursively chooses the candidate with largest impurity gain until the stop criteria is achieved.

In this study, performance of the decision tree algorithm was evaluated for Gini's Diversity Index and Entropy impurity criterions.

### 3.3.3 Support Vector Machine Algorithm

One of the most popular learning algorithm is Support Vector Machines (SVM) which basically separates the data two categories by finding an optimal hyperplane [58]. The optimal hyperplane is constructed using the support vectors which are the closest points in the two classes each other. The gap between the hyperplane and support vectors is called margin and SVM aims to find the maximum margin for better separation [80].

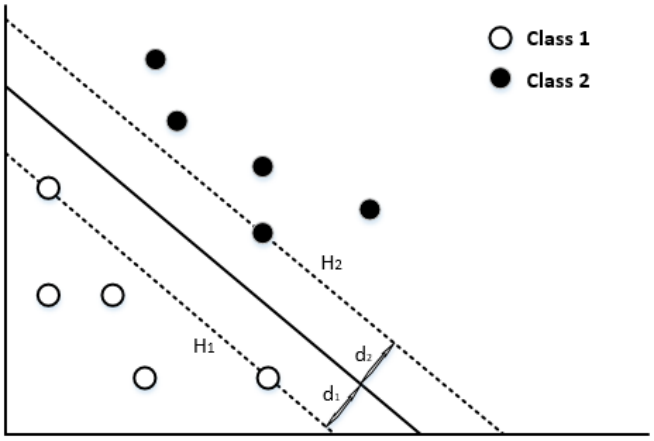



**Figure 3.6:** An example of SVM algorithm

Figure 3.6 shows the support vector points , decision boundary $(H)$ and margin $(d)$. The points on the training set are defined as [60],

$$x_i w + b \geq +1, \qquad for\ y_i = +1 \tag{3.27}$$

$$x_i w + b \leq -1\,, \qquad for\ y_i = -1 \tag{3.28}$$

where $y$ refers the classes. Combining the equation 3.27 and 3.28,

$$y_i(x_i w + b) - 1 \geq 0\,, \qquad for\ all\ i \tag{3.29}$$

The hyperplane can be described as $xw + b = 0$ where $w$ is normal vector of the $H$ and $b$ is bias. Therefore, hyperplanes that intersect with support vector points can be described as [81],

$$x_i w + b = +1, \qquad for\ H_1 \tag{3.30}$$

$$x_i w + b = -1\,, \qquad for\ H_2 \tag{3.31}$$

The margin also can be described as $1/w$ thanks to vector dot production. Minimizing $w$ is maximize the margin that is the goal of SVM. Also, minimizing $w$ has same result with minimizing $\frac{1}{2}\|w\|^2$ and also that consideration makes easy the solution. At the end, the objective function can be described as,

$$\min_w \frac{1}{2}\|w\|^2 \ such\ that\ \ y_i(x_i w + b) - 1 \geq 0\,, \forall_i \tag{3.32}$$

Lagrange multipliers $\alpha$ is used to solve the constraint optimization problem,

$$L_P \equiv \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i [y_i(x_i w + b) - 1] \tag{3.33}$$

where $\alpha_i > 0$. The objective function can be minimized by taking derivatives with respect to *w* and *b* and then setting them to zero,

$$\frac{\partial L_P}{\partial w} = 0 \rightarrow \; w = \sum_{i=1}^{L} \alpha_i y_i x_i \tag{3.34}$$

$$\frac{\partial L_P}{\partial b} = 0 \rightarrow \; \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.35}$$

Equation 3.33 can be rewritten using 3.34 and 3.35,

$$L_D \equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i,\mathrm{j}}^{L} \alpha_i \alpha_j y_i y_j x_i x_j] \;\; \text{s.t. } \alpha_i \geq 0 \; \forall_i, \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.36}$$

$$L_D \equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \alpha^T H \alpha \;\; \text{s.t. } \alpha_i \geq 0 \; \forall_i, \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.37}$$

where $H_{ij} = y_i y_j x_i x_j$. The new objective function in terms of $\alpha_i$ is obtained as in equation 3.38.

$$\max_{\alpha} [\sum_{i=1}^{L} \alpha_i - \frac{1}{2} \alpha^T H \alpha] \;\; \text{s.t. } \alpha_i \geq 0 \; \forall_i, \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.38}$$

The problem is converted from minimizing $L_p$ to maximizing $L_D$ [81]. It is a quadratic optimization problem and quadratic programming solver can give $\alpha$ as a solution. Using equation 3.38 the *w* and *b* can be calculated.

SVM is originally a linear classification method; however, nonlinear kernel transforms can be applied to obtain non-linear classification [60]. According that method, the data is transformed to higher dimensional space that allows linear separation. $H$ in the objective function related with the kernel *(k)* and can be rewritten as $H_{ij} = y_i y_j k(x_i x_j)$. There are various kernel methods in the literature such as linear

kernel and radial basis function (RBF) [81]. Definition of them are shown in equation 3.39 and equation 3.40.

Linear Kernel

$$k(x_i, x_j) = x_i \cdot x_j \tag{3.39}$$

RBF

$$k(x_i, x_j) = \exp(-\frac{|x_i - x_j|^2}{2\sigma^2}) \tag{3.40}$$

where $\sigma$ is a free parameter which defines the radius [60].

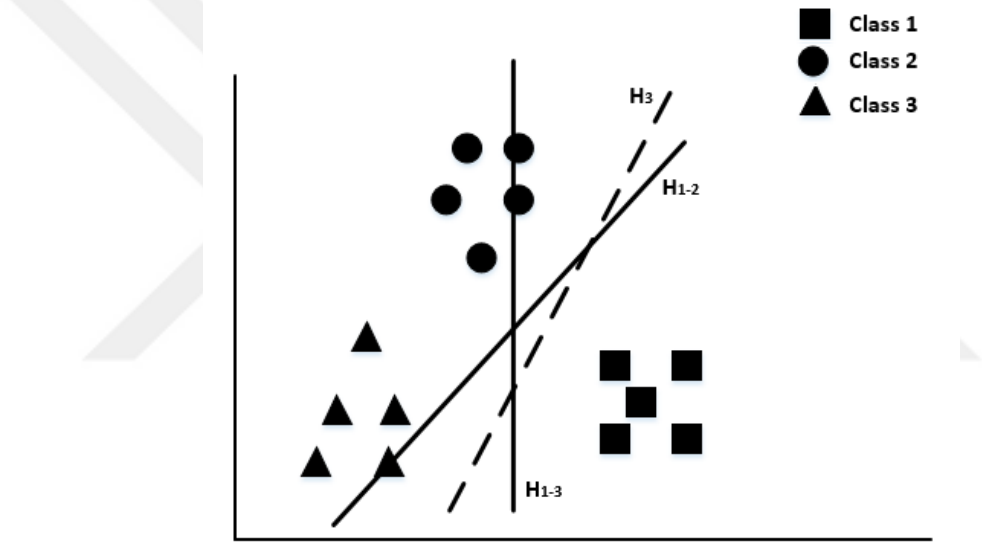



**Figure 3.7:** OVO and OVA classification.

SVM is a maximum margin classifier which seperate the data with optimal hyperplane; therefore, the results become binary classified. However, the dataset in our case, $F$, is multiclass. Hence, in literature, there are binary classification methods such as the one versus one (OVO) and one versus all (OVA) which can be used for multiclass classification [82]. OVO trains a seperate classifier for each the classes pair. For example, if there are $C$ classes in dataset, it is generated $C - 1$ classsifer for each class. Therefore, OVO leads $C(C-1)/2$ classifiers in total. OVA trains one classifier for each class by considering pairs as the selected class and rest of all. For instance, if class i is seleceted, it is assumed i labels as positive and the rest in the dataset as negative. Thus, OVA leads $C$ classifiers. In Figure 3.7, *Class 1* in the data

set which includes three classes is seperated with OVO ($H_{1-2}\ and\ H_{1-3}$) and OVA ($H_3$) methods.

In this work, SVM algorithm was applied using linear and RBF kernels. Also the OVO and OVA methods were tested.

### 3.3.4 Artificial Neural Networks Algorithm

The study of artificial neural network (ANN) is inspired by observations of biological learning system, brain, that consist of complex and interconnected neurons [58]. The human brain contains approximately $10^{11}$ neurons and each neuron is connected to about $10^4$ other neurons, hence it operates highly parallel process [77]. Basic neuron activities are inhibition or excitement the connected neurons through synapses. ANN simulates that huge network as a mathematical model.

Construction of an artificial neural network starts with creating a single neuron. First artifical neuron model was proposed by McCulloch and Pitts [83]. Basically, the neuron takes the input, and then directly feeds the output through weighted connections and a transfer function [84]. It is considered as the simplest model of feed-forward neural network which can be shown in Figure 3.8.

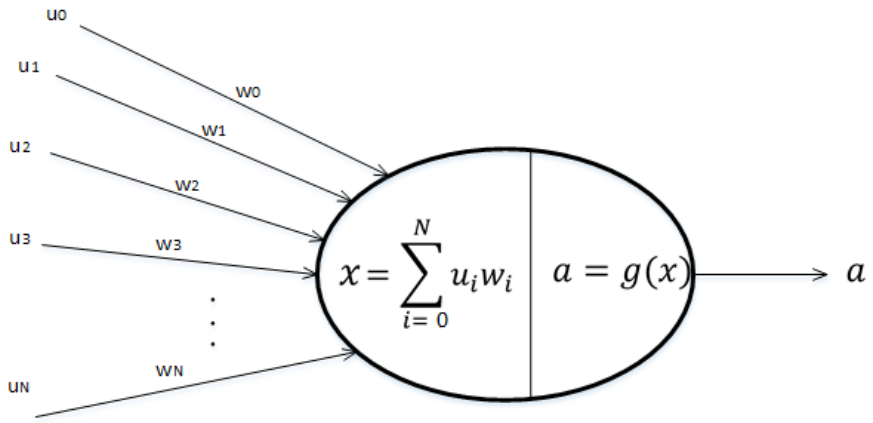



**Figure 3.8:** A model of single neuron.

where, $u_1$, $u_2$,..., $u_N$ and $w_1$,$w_2$, ...$w_N$ refer the input and weights, respectively. $u_0$ known as bias is usually selected as 1 and $w_0$ is weight of bias. An adder sums the weighted inputs and bias to obtain a numerical value [84]. Then, the activation function calculates the output of the neuron. Mathematical model of single neuron can be expressed equation 3.41.

$$a = g\left(\sum_{i=0}^{N} u_i w_i\right) \ldots \tag{3.41}$$

where $a$ is output of the neuron. Although there are linear activation function, nonlinear activation functions such as *sigmoid*, *hyperbolic tangent sigmoid* and *softmax*.are often used [86] [87].

Sigmoid,

$$g(x) = \frac{1}{1 + e^{-x}} \tag{3.42}$$

Hyperbolic tangent sigmoid,

$$g(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{3.43}$$

Softmax,

$$g(x_i) = \frac{e^{x_i}}{\sum_{c=1}^{C} e^{x_c}} \quad for\ c = 1, \ldots, C \tag{3.44}$$

The simplest architecture of ANN is formed only one layer that includes one neuron, hence that structure provides limited model [84]. However, most of the application uses complex property of ANN using multi-layer network topology which consist of input, hidden and output layers. Input layer takes input values and feeds the network, output layer is the last layer in the network and all the other layers are hidden layers. It is possible to create different network architecture with changing the number of hidden layer and the number of neurons in a layer, as well as, the connection type between layers and neurons. The multi layered perceptron (MLP) model is used in this work as shown in Figure 3.9. In MLP, each layer is fully connected to next layer [87]. Output value of a neuron in each layer except output layer is the input value of neurons in the next layer. To calculate the output value of the network $o_i$, the equation 3.41 is applied to all neurons from input to output. This process is known as forward propagation.

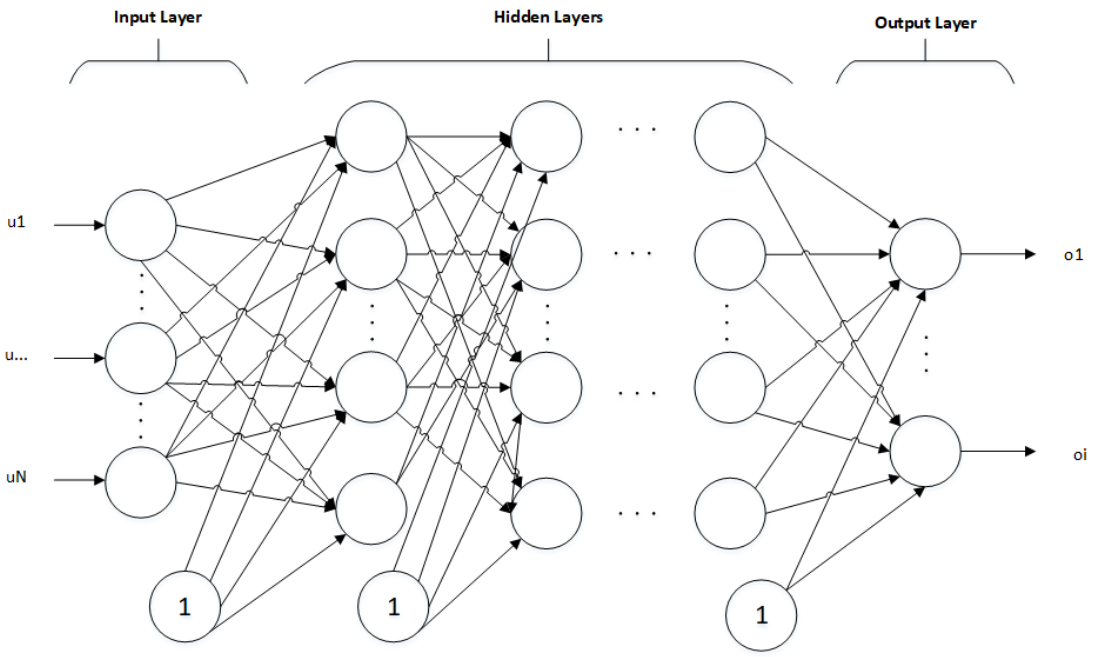



**Figure 3.9:** Structure of multilayered perceptron.

In addition to structure, one of the most important part of the neural network is backpropagation which is used to train multilayer ANN. Training the ANN means adjusting the weights of the neurons considering the error in the prediction [88]. The error between output $o_i$ and target $t_i$ can be calculated by different methods such as Mean Square Error (MSE) and Cross Entropy [89].

MSE

$$E = \frac{1}{N}\sum_{i=1}^{N}(o_i - t_i)^2 \tag{3.45}$$

Cross Entropy

$$E = -\sum_{i=1}^{N} t_i \log o_i \tag{3.46}$$

The backpropogation backwardly propogates the calculated error through the network. The gradient is used to update the weights according to training algorithm, whose goal is reduce the error [89]. There are many training function in literature. for instance gradient descent backpropagation, Levenberg-Marquardt backpropagation, Scaled Conjugate Gradient and Resilient Backpropagation [91].

Gradient descent backpropagation is most known backpropagation algorithim [90]. Consider that, $w_i$ is one of the weight of output neuron. The aim of the backpropagation is to find that how much a change in $w_i$ affects the total error $E$. The error value is propagated backward through the network with the help of the chain rule. The calculated affects are multiplayed by a learning rate α and subtracted from current weights to obtain the new weights [77].

Basically, the gradient descent calculates the derivative of $E$ with respect to $w_i$ to find the affect of change in weight [86],

$$\delta_i = \frac{\partial}{\partial w_i} E \tag{3.47}$$

Chain rule can be applied to find the gradient of the error [86]:

$$\frac{\partial E}{\partial w_{ij}^k} = \frac{\partial E}{\partial a_j^k} \frac{\partial a_j^k}{\partial x_j^k} \frac{\partial x_j^k}{\partial w_{ij}^k} \tag{3.48}$$

where $w_{ij}^k$ refers the weight between node *j* in layer *k* and node *i* in layer *k-1*. $x_j^k$ is weighted summation of and $a_j^k$ is output of the node *j* in layer *k*. For updating the weights, the proportion of the the gradient is subtracted from the current weight to decrease the error [92] [93].

$$w_{ij}^k = w_{ij}^k - \alpha \frac{\partial E(X, \theta)}{\partial w_{ij}^k} \tag{3.49}$$

where α is the learning rate. The weights continue to update until the stop criterion is satisfied.

Resilient backpropogation is another training algorithm, which varies from gradient descent in the update step [94]. The algorithm basically elememinates the effects of the partial derivative magnitudes. The sign of the derivative is used to find the direction of gradient and the magnitude is selected as seperate value [95]. Pseudo code of the algorithm is shown in Table 3.1. According that, the magnitude value is increased by a factor if the gradient has same sign for two consecutive iterations. The magnitude value is decreased by a factor when the derivative with respect to weight

changes sign from previous iteration. Otherwise, if derivative is zero, the value remains same [96].

**Table 3.1:** Resilient backpropogation algorithm [95].

Repeat
  Compute Gradient $\frac{\partial E}{\partial w}(t)$
  For all weights and biases {
    If ($\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) > 0$) then {
      $\Delta_{ij}(t) = \text{minimum}\left(\Delta_{ij}(t-1) * \eta^{+}, \Delta_{max}\right)$
      $\Delta w_{ij}(t) = -sign\left(\frac{\partial E}{\partial w_{ij}}(t)\right) * \Delta_{ij}(t)$
      $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
      $\frac{\partial E}{\partial w_{ij}}(t-1) = \frac{\partial E}{\partial w_{ij}}(t)$
    }
    else if ( $\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) < 0$) then {
      $\Delta_{ij}(t) = \text{maximum}\left(\Delta_{ij}(t-1) * \eta^{\pm}, \Delta_{min}\right)$
      $\frac{\partial E}{\partial w_{ij}}(t-1) = 0$
    }
    else if ($\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) = 0$) then {
      $\Delta w_{ij}(t) = -\text{sign}\left(\frac{\partial E}{\partial w_{ij}}(t)\right) * \Delta_{ij}(t)$
      $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
      $\frac{\partial E}{\partial w_{ij}}(t-1) = \frac{\partial E}{\partial w_{ij}}(t)$
    }
}

Levenberg Marquardt algorithm is used for nonlinear least squares problems. It is an iterative technique that is used the approximation of the Hessian matrix if the performance function has the form of a sum of squares [97]. The Hessian matrix which is the second-order partial derivatives of a scalar-valued function can be obtained as,

$$H = J^{T}J \tag{3.50}$$

and then gradient can be computed as,

$$g = J^T e \tag{3.51}$$

where $e$ is the vector of network errors and $J$ is the Jacobian matrix, that includes first order derivatives of the errors with respect to the weights and biases. Then the weights are updated as,

$$w_{k+1} = w_k - [J^T J + \mu I]^{-1} J^T e \tag{3.52}$$

where $\mu$ is the step size. $\mu$ is decreased after each succesful step and, in contrast, $\mu$ is increased if error becomes larger [98]. In this way, the error is getting smaller at each iteration. Levenberg Marquardt algorithm is known as fastest training algorithm for networks of moderate size; however, it requires more memory and computation [99].

In the conjugate gradient algorithims, the adjustment of the weight is realized by considering the conjugate directions rather than the negative of the gradient [99]. Scaled conjugate gradient uses the step size scaling mechanism; therefore, does not require the line search at each iteration like the standart conjugate gradient algorithims. The basic methodology of the scaled conjugate gradient algorithim is defined in Table 3.2 [100]. According the pseudo code, the design parameters are updated as user independently. It increases the number of iterations but reduces the computations in each iteration [88].

In this study, MLP is employed with a structure of one for input layer, one for hidden layer and one for output layer. Number of neurons in the hidden layer is examined from 5 to 55. Also gradient descent backpropagation, Resilient Backpropagation, Levenberg-Marquardt backpropagation and Scaled Conjugate Gradient methods are applied as training functions.

**Table 3.2:** Scaled conjugate gradient algorithm [100].

1. Choose initial weight vector $w_1$.
   Set $p_1 = r_1 = -E'(w_1), k = 1.$

2. Calculate second order information:
   $$s_k = E''(w_k)p_k,$$
   $$\delta_k = p_k^T s_k.$$

3. Calculate step size:
   $$\mu_k = p_k^T r_k,$$
   $$\propto_k = \frac{\mu_k}{\delta_k}.$$

4. Update weight vector:
   $$w_{k+1} = w_k + \propto_k p_k,$$
   $$r_{k+1} = -E'(w_{k+1}).$$

5. If k mod N=0 then restart algorithm: $p_{k+1} = r_{k+1}$
   else create new conjugate direction:
   $$\beta_k = \frac{|r_{k+1}|^2 - r_{k+1}r_k}{\mu_k}$$
   $$p_{k+1} = r_{k+1} + \beta_k p_k.$$

6. If the steepest descent direction $r_k \neq 0$ then set $k = k + 1$ and go to 2
   Else terminate and return $w_{k+1}$ as the desired minimum.

### 3.4 Evaluation Metrics

In machine learning, the quality of the algorithm is measured with evaluation metrics. The classifiers performances is compaired by calculating various evaluation metrics which are basically depends on number of class in the task [101]. In this work, multi-class classification was performed; therefore, the evaluation metrics were required for each class and over-all performance.

The performance measures of multi-class classification are similar with the binary classification. First, the confusion matrix is obtained using the known target values

and the outputs values of the classifier [102]. There are four terms in the confusion matrix which are,

- True Positive (TP): number of true estimated instances which are correct
- False Positive (FP): number of true estimated instances which are actually false
- True Negative (TN): number of false estimated instances which are correct
- False Negative (FN): number of false estimated instances which are actually true

The accuracy, the first evaluation metrics, is calculated as the portion of the true positives and total number of instances [102].

$$\text{Accuracy} = \frac{tp + tn}{tp + fn + fp + tn} \tag{3.53}$$

The accuracy focuses on the overall effectiveness of the classifier. However, it is not enough for some cases. For example, if larger number of instances for one of the class is tested, the effect of that class in the result will be dominant. Hence, the other metrics are required like precision, recall and F score [101]. The precision is the ratio between number of correctly classified positive instances and outputs of the classifier which are labelled as positive. Another metric is the recall which is the ratio between number of correctly classified positive instances and number of the positive classes in target. Last, the combination of the precision and the recall generates a metric, F score. In multiclass classification, these mectics are calculated for micro-averaging ($\mu$) and macro-averaging ($M$) criteria. The micro-averaging takes into account the frequency of each class; on the other hand, the macro-averaging normalizes the metrics using all class; therefore, the number of samples in each class is not considered. The mathematical expression of these metrics for micro averaging and macro averaging assessment is given in Table 3.3.

In addition to overall classification performance, success rate of each class is important factor to determine the quality of classifier. For this purpose, precision and recall values of each class were examined.

**Table 3.3 :** Multiclass performance measures [101].

| Measure | Formula |
|---|---|
| Micro Precision | $\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i + fp_i)}$ |
| Micro Recall | $\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i + fn_i)}$ |
| Micro Fscore | $\frac{(\beta^2 + 1)precision_\mu \, recall_\mu}{\beta^2 precision_\mu + recall_\mu}$ |
| Macro Precision | $\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}$ |
| Macro Recall | $\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l}$ |
| Macro Fscore | $\frac{(\beta^2 + 1)precision_M \, recall_M}{\beta^2 precision_M + recall_M}$ |

All the algorithms were feeded by the feature vector with reduced dimension $F$. To improve the validation, k-fold method was applied. According that, the input dataset is divided to $t$ subdata set and labelled by randomly for 1 to $t$ [60]. The algorithm is runned for $t$ times and each time one of the subdata is chosen for test and the rest is used for training. At the end of $t$ times, the average is considered as output for metrics. Number of fold was selected 10 for this work.

# 4. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this chapter, the classification performances of kNN, DT, SVM and ANN algorithims will be investigated in terms of evaluation metrics defined in section 3.4. Statistics and Machine Learning Toolbox™ and Neural Network Toolbox in Matlab was used as a tool for classification. Input of the all algorithims given in equation 3.19 as $F$ which is the output of the dimension reduction process.

First, each algorithm will run with different parameters to find an optimal one and to evaluate that cases, only the accuracy will be compared. Then, the algorithms with the best performance parameters will be examined by other evaluation metrics. At the end, all proposed algorithims will be compared.

As it was mentioned in section 3.4, 10-fold cross validation was applied for all classification results.

### 4.1 Classification Results of k Nearest Neighbor Algorithm

kNN is the first algorithm investigated in this thesis. The details of the algorithm was given in section 3.3.1 . *fitcknn()* function in the toolbox was used as kNN classifier. Mainly, there are two parameters in the kNN classifier which are distance metrics and number of neighbor, *k*. The distance metrics was applied for Euclidean, Cityblock, Minkowski and Chebyshev distance along with the changing values of *k* from 1 to 10. The results are shown in Figure 4.1.

Results indicate that increasing the value of *k* improves the accuracy for all the distance metrics; however, that may cause overfitting. Hence, the value of *k* was chosen as 6 to obtain the optimal solution. The distance metrics of Euclidean and Minkowski gave almost same accuracy, but Chebychev distance has low accuracy. Cityblock distance metrics was shown as better performance than others; hence, it was selected as best distance metrics.

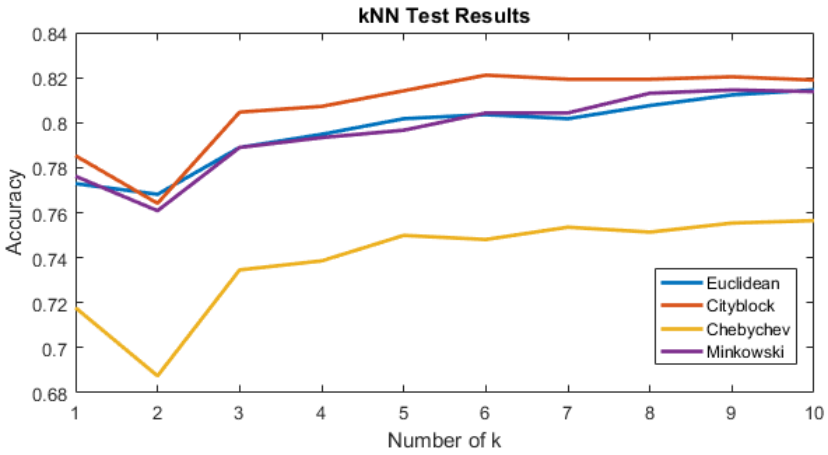



**Figure 4.1:** Accuracy of kNN algorithim with 4 different distance metrics and increasing value of *k.*

The performance of the kNN algorithm with the Cityblock distance metrics and 6 nearest neighbor were examined in detail. The experiments were repeated 10 times using same data set for one subject.

**Table 4.1 :** kNN algorithm results.

| Experiments | $Fscore_{\mu}$ | $Fscore_{M}$ |
|---|---|---|
| Experiment1 | 0.8167 | 0.8261 |
| Experiment2 | 0.8168 | 0.8280 |
| Experiment3 | 0.8167 | 0.8276 |
| Experiment4 | 0.8196 | 0.8290 |
| Experiment5 | 0.8185 | 0.8286 |
| Experiment6 | 0.8157 | 0.8265 |
| Experiment7 | 0.8120 | 0.8219 |
| Experiment8 | 0.8189 | 0.8280 |
| Experiment9 | 0.8167 | 0.8265 |
| Experiment10 | 0.8156 | 0.8156 |
| Average | 0.8167 | 0.8268 |
| Variance ($*10^{-5}$) | 0.4539 | 0.4163 |

At the end of 10 experiments, the average of $Fscore_{\mu}$ and $Fscore_{M}$ were achieved as 0.8167 and 0.8268 respectively. Next, the performance of each class with using the results of same experiments was evaluated in terms of recall, precision and Fscore that are shown in Figure 4.2.

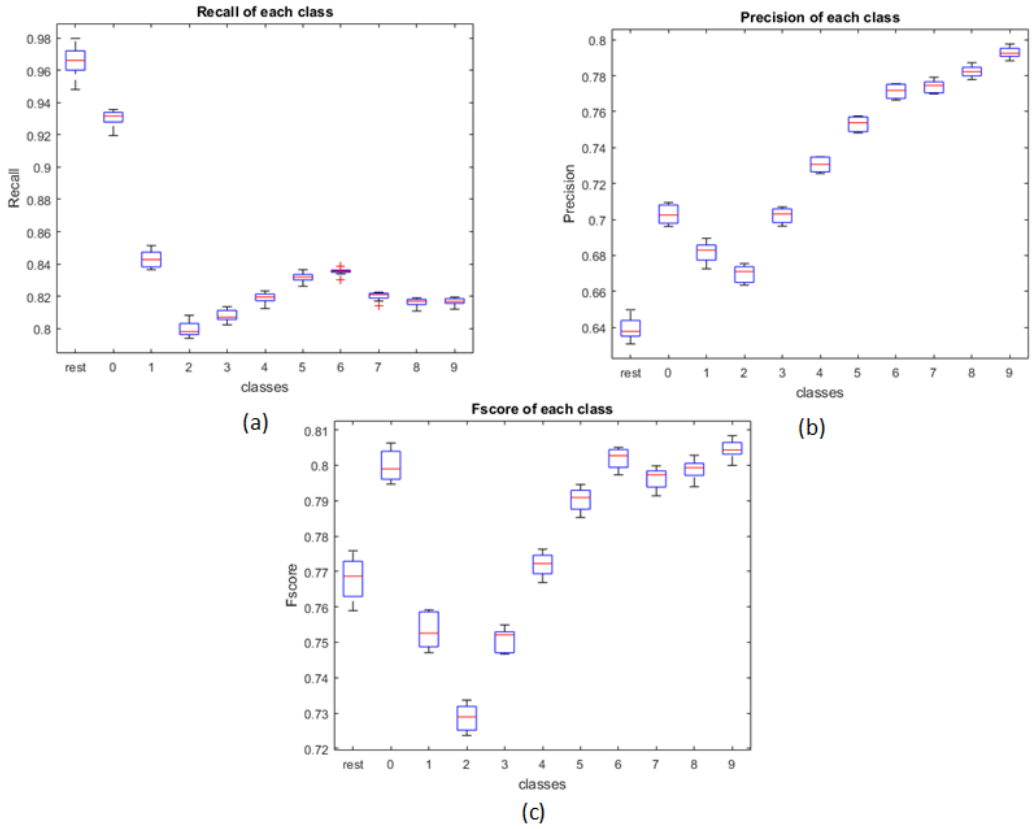



**Figure 4.2:** Performance of each class using kNN (a) Recall (b) Precision (c) Fsocre.

The recall shows how well the positive classes are estimated. Class ‘rest’ and ‘0’ have significantly higher recall ratio; however, they have also higher variance. Minimum recall was achieved in class ‘2’. The precision evaluation gives how many of the classes correctly guessed by the classifier are actually true. The precision value of class ‘rest’ was the lowest and class ‘9’ was the highest. All the classes have almost same variance about precision values. Fscore gives the haromonic mean of recall and precision. Class ‘0’, class ‘6’ and class ‘9’ have higher fscore values; whereas, class ‘2’ has the lowest. Fscore ratio of class ‘1’ and ‘3’ was about 0.75 and all the others had more than 0.77. In summary, the classifier is able to classify correctly most of class ‘rest’, but not all the predicted class ‘rest’ is correct. The lowest classification performance was achieved for class ‘2’.

### 4.2 Classification Results of Decision Tree Algorithm

Second, DT algorithm was studied using *fitctree()* function in the toolbox. According that, the algorithm recursively splits the data into two categories until the stop condition is satisfied. There are four stop criteria in the toolbox. First two of them, if

the node is pure or the maximum number of split value is achieved, the algorithm stops. Next, the minimum parent size, which is defined as minimum number of branch node observations, is another limitation to stop. Last criterion is the minimum leaf size that means minimum number of leaf node observations. The default values were chosen as 1, 10 and 1 for the maximum number of split, the minimum parent size and minimum leaf size respectively. *gdi* and *entropy* split criterions mentioned in section 3.3.2 were tested for 10 experiments. The results can be seen in Figure 4.3.

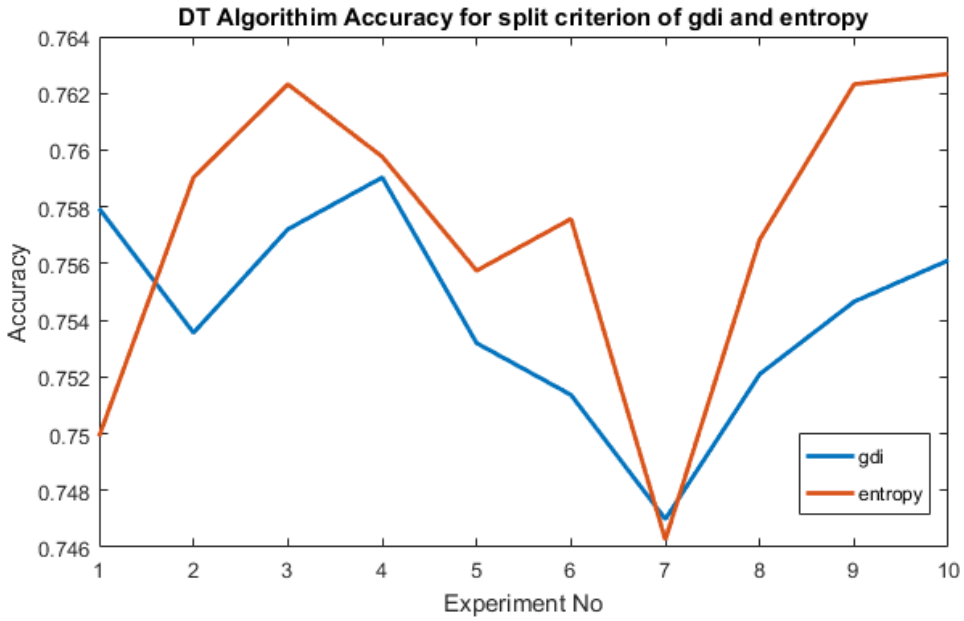



**Figure 4.3:** Results of *gdi* and *entropy* split criterions.

The result shows that split criterion accuracy of the *entropy* is better than the *gdi*. Thus, the other evaluation metrics were analysed for *entropy*.

**Table 4.2 :** DT algorithm results.

| Experiments | $Fscore_{\mu}$ | $Fscore_{M}$ |
|---|---|---|
| Experiment1 | 0.7510 | 0.7545 |
| Experiment2 | 0.7675 | 0.7709 |
| Experiment3 | 0.7502 | 0.7537 |
| Experiment4 | 0.7576 | 0.7625 |
| Experiment5 | 0.7678 | 0.7721 |
| Experiment6 | 0.7539 | 0.7577 |
| Experiment7 | 0.7514 | 0.7551 |
| Experiment8 | 0.7466 | 0.7497 |
| Experiment9 | 0.7729 | 0.7756 |
| Experiment10 | 0.7576 | 0.7601 |
| Average | 0.7576 | 0.7612 |
| Variance ($*10^{-4}$) | 0.7855 | 0.7828 |

As an average of 10 experiments, it was obtained 0.7576 and 0.7612 for $Fscore_{\mu}$ and $Fscore_{M,}$ respectively. In addition, the performance of each class with using the results of same experiments was evaluated in terms of recall, precision and Fscore that are shown in Figure 4.4.

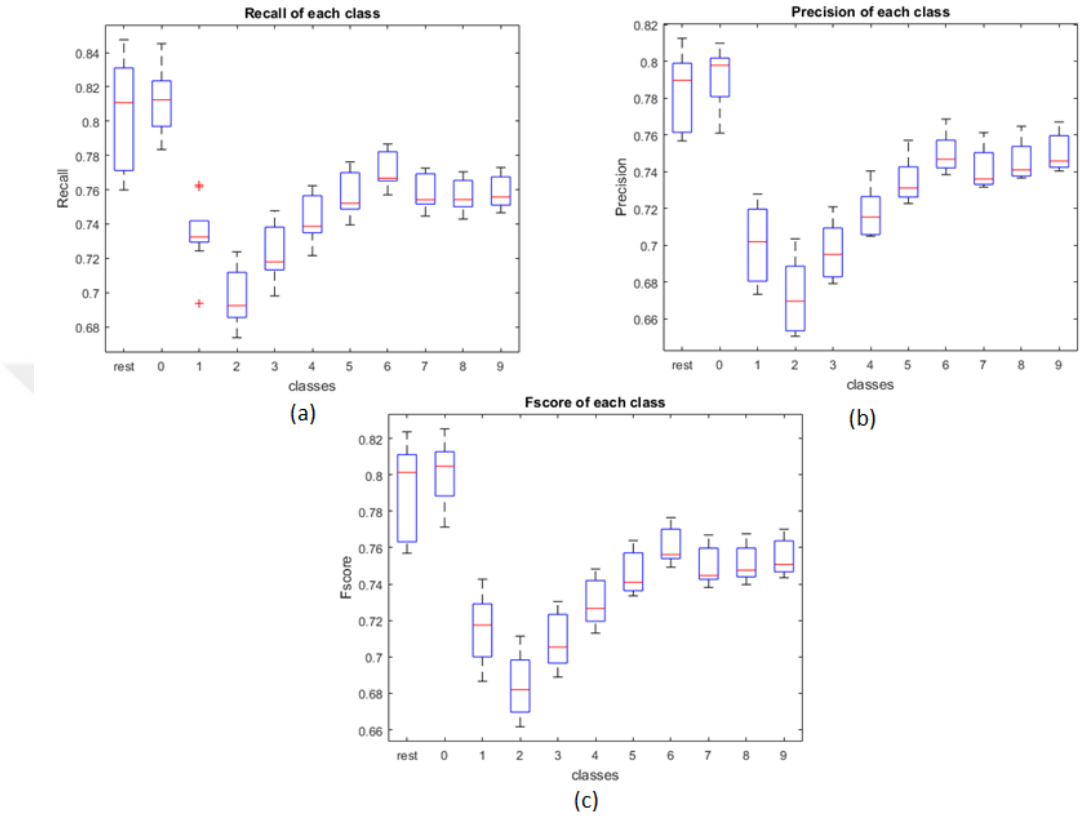



**Figure 4.4:** Performance of each class using DT (a) Recall (b) Precision (c) Fsocre.

Based on the results obtained, class 'rest' and '0' have higher recall and precision ratio; therefore, they have higher fscore ratio. On the other hand, class '2' has lowest recall and precision. The recall and precision ratios of all the other classes were interrelated. In short, the DT classifier can detect class 'rest' and '0' correctly, whereas class '2' confusing. In addition, the correct classification performance of class '5', '6', '7', '8' and '9' were about the same, while class '1', '3' and '4' was lower.

### 4.3 Classification Results of Support Vector Machine Algorithm

The SVM algorithm defined in section 3.3.3 was investigated. The learner was created by *templateSVM()* function and trained with *fitcecoc()* function in the toolbox. First, the effect of kernel was tested for *linear* and *rbf* kernels. It was

applied with setting the *'KernelFunction'* to *'linear'* and *'rbf'*. In *rbf* kernel case, *'KernelScale'* parameter was setted to *'auto'*. Obtained accuracies were shown in Figure 4.5.

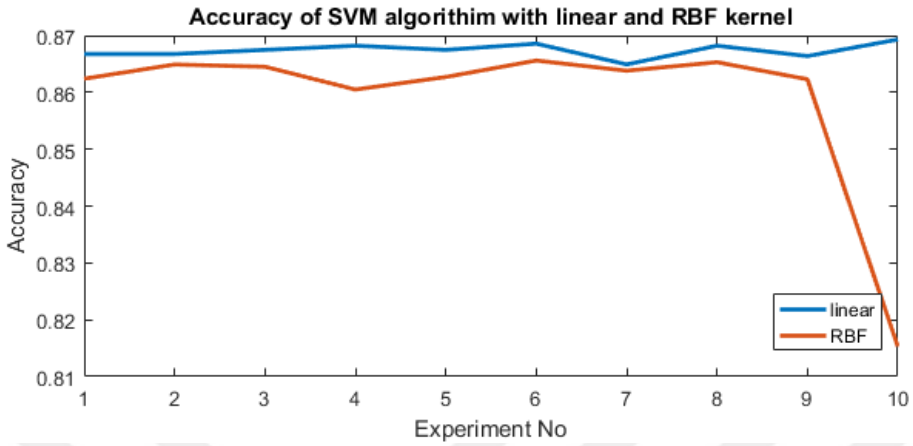



**Figure 4.5:** Accuracy of SVM classifier with linear and RBF kernel.

The result of testing showed that accuracy of the linear SVM was better than the RBF kernel. Second, OVO and OVA methods were applied to algorithm. For testing purpose, *'KernelFunction'* was fixed to *'linear'* in *templateSVM* and *'Coding'* parameter in *fitcecoc()* was setted to *'onevsone'* and *'onevsall'*. The result of the test is shown in Figure 4.6.

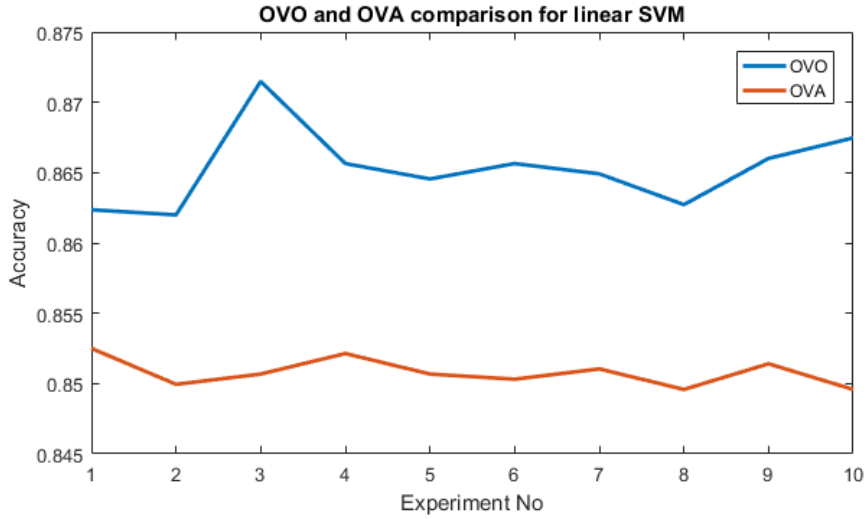



**Figure 4.6:** Accuracy of OVO and OVA methods for linear SVM.

As a result, the OVO method for the linear SVM was obtained as a better parameter for our case; therefore, following experiments were performed using these parameters. The SVM algorithim was tested for 10 times and the results were obtained as in Table 4.3.

**Table 4.3 :** SVM algorithm results.

| Experiments | $Fscore_{\mu}$ | $Fscore_{M}$ |
|---|---|---|
| Experiment1 | 0.8675 | 0.8728 |
| Experiment2 | 0.8661 | 0.8715 |
| Experiment3 | 0.8645 | 0.8699 |
| Experiment4 | 0.8668 | 0.8723 |
| Experiment5 | 0.8689 | 0.8743 |
| Experiment6 | 0.8693 | 0.8744 |
| Experiment7 | 0.8620 | 0.8677 |
| Experiment8 | 0.8660 | 0.8713 |
| Experiment9 | 0.8645 | 0.8695 |
| Experiment10 | 0.8657 | 0.8710 |
| Average | 0.8661 | 0.8715 |
| Variance ($*10^{-5}$) | 0.4702 | 0.4439 |

The results showed that the average of $Fscore_{\mu}$ and $Fscore_{M}$ were 0.8661 and 0.8715 respectively. Further, analysis of each class was resultant as shown in Figure 4.7.

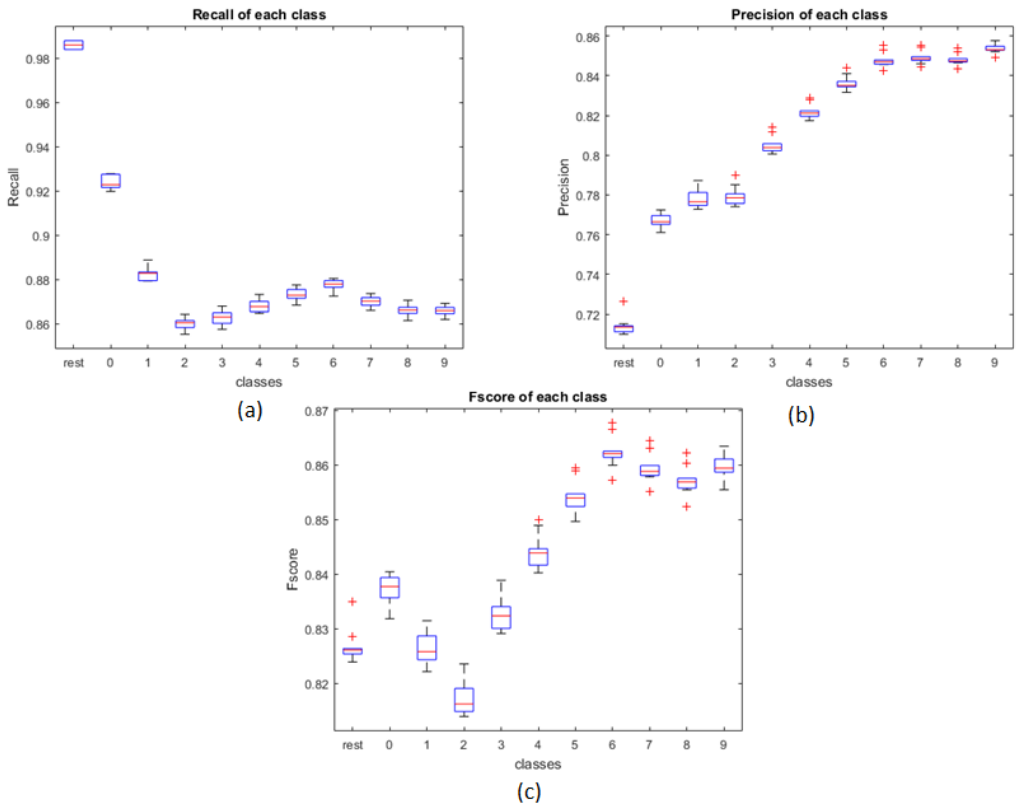



**Figure 4.7:** Performance of each class using SVM (a) Recall (b) Precision (c) Fsocre.

Class 'rest' was found as significantly highest recall and lowest precision ratio. Although between classes '2' and '9' have lower than 0.88, the minimum recall ratio has achieved for class '2'. The maximum precision ratio has obtained in class '9'.

Precision ratio of all classes excep class ‘rest’ are higher than 0.76. Class ‘6’ and ‘2’ have highest and lowest Fscore ratio, respectively. Variance of Fscore was relatively higher for classes ‘0’, ‘1’, ‘2’, ‘3’ and ‘4’.

### 4.4 Classification Results of Artificial Neural Network Algorithm

The last algorithm used in this work was ANN which was introduced in section 3.3.4 . According that, the structure of the network, number of neurons and layers are important parameters that have to be setted. The feedforward neural network was generated using *patternnet()* function in toolbox.

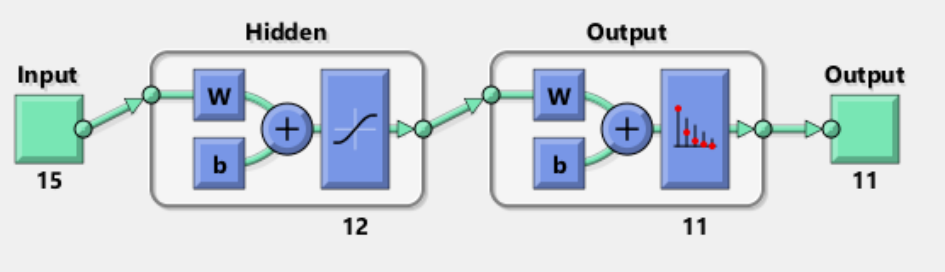



**Figure 4.8:** Structure of created ANN.

The network was configured, as shown in Figure 4.8, including one input, one hidden and one output layer. *tansig* activation function was used for hidden layer and *softmax* for output layer. First, error function was setted to *crossentropy*. Effect of neuron size in the hidden layer was tested from 5 to 55 with the *transcg*, *trainrp* and *traingd* functions in the training. *trainlm* training function requires square error; thus, could not have used with *crossentropy*. The results are shown in Figure 4.9.

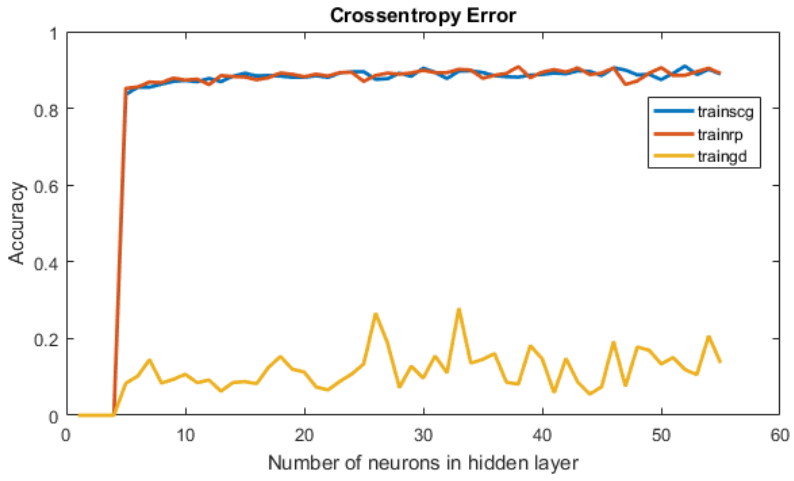



**Figure 4.9:** Accuracy of the transcg, trainrp and traingd training functions with crossentropy error.

The network with same configuration was also tested for *MSE* error function. In that case, additionaly, *trainlm* training function was applied to network. The results are illustrated in Figure 4.10.

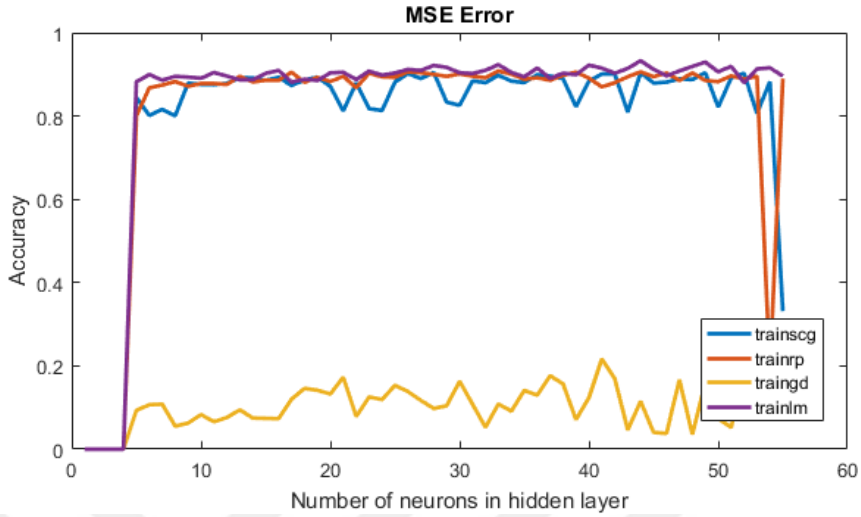



**Figure 4.10:** Accuracy of the *transcg*, *trainrp*, *traingd* and *trainml* training functions with *MSE* error.

The accuracy of the *trainlm* looks better than the others; however, there were significant difference in the training time. Considering the effect of the parameters, the optimal ones were found as 12 for neuron size, *trainrp* function for training and *crossentropy* for error function. The detailed performance metrics were calulated using optimal parameters. $Fscore_{\mu}$ and $Fscore_{M}$ ratios of 10 experiments are given in Table 4.4.

**Table 4.4 :** ANN algorithm results.

| Experiments | $Fscore_{\mu}$ | $Fscore_{M}$ |
|---|---|---|
| Experiment1 | 0.8536 | 0.8581 |
| Experiment2 | 0.8540 | 0.8591 |
| Experiment3 | 0.8529 | 0.8578 |
| Experiment4 | 0.8533 | 0.8580 |
| Experiment5 | 0.8503 | 0.8548 |
| Experiment6 | 0.8547 | 0.8596 |
| Experiment7 | 0.8503 | 0.8551 |
| Experiment8 | 0.8576 | 0.8636 |
| Experiment9 | 0.8459 | 0.8511 |
| Experiment10 | 0.8485 | 0.8532 |
| Average | 0.8521 | 0.8570 |
| Variance ($*10^{-4}$) | 0.1142 | 0.1293 |

The average of $Fscore_{\mu}$ and $Fscore_{M}$ were achieved as 0.8521 and 0.8570, respectively. Besides, recall, precision and Fscore ratios of each class is obtained using the same experiments that are shown in Figure 4.11.

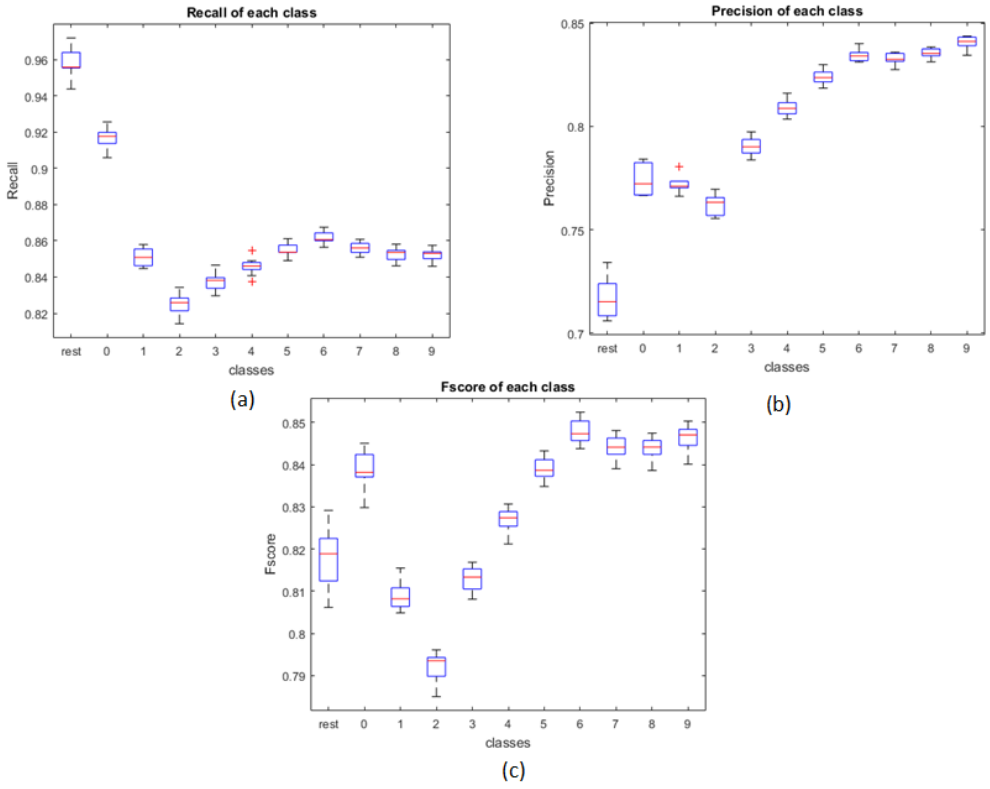



**Figure 4.11:** Performance of each class using ANN (a) Recall (b) Precision (c) Fsocre.

According to results, for class '2', the recall ratio was lowest as similar with the other classifiers. The maximum recall was obtained for class '0'. Precision of class '9' was the highest, but it did not have major difference than classes '6', '7' and '8'. The lowest precision ratio was obtained for class 'rest'. The variance of precision was higher for class 'rest' and '0'. Minimum and maximum Fscores were obtained for class '2' and class '6', respectively. Also, the highest variance of Fscore was achieved for class 'rest'.

### 4.5 Overall Performance Comparison

In this section, the recognition rates, which were obtained using best parameters, were compared. Table 4.5 shows the average $Fscore_{\mu}$ and $Fscore_{M}$ of implemented

algorithms. DT with entropy splitting criterion has 0.7576 and 0.7612 average of $Fscore_{\mu}$ and $Fscore_{M}$, respectively; therefore, it was the worst result in the system. kNN with 6 neighbor and Cityblock distance metrics enhanced the performance to 0.8167 and 0.8268 average ratio of $Fscore_{\mu}$ and $Fscore_{M}$, respectively. To improve the results, ANN was developed with 12 neurons in hidden layer, *trainrp* training function and *crossentropy* error function. Results shows that the $Fscore_{\mu}$ and $Fscore_{M}$ is improved to 0.8521 and 0.8570, respectively. Lastly, the linear SVM with OVO method raised the results to 0.8661 and 0.8715.

**Table 4.5 :** Overall Results.

| Experiments | Average $Fscore_{\mu}$ | Average $Fscore_{M}$ |
|---|---|---|
| kNN | 0.8167 | 0.8268 |
| DT | 0.7576 | 0.7612 |
| SVM | 0.8661 | 0.8715 |
| ANN | 0.8521 | 0.8570 |

Besides, the performance of each class were given in Table 4.6,

Table 4.7 and Table 4.8. The class ‘rest’ has highest recall ratio for kNN, SVM and ANN algorithms. Class ‘2’ has lowest recall ratio for all classifiers. The highest precision ratio of kNN, SVM and ANN classifiers were achieved for class ‘9’; whereas, SVM and ANN gave also equal ratio with class '9' for classes ‘6’, ’7’ and ‘8’. The lowest precision ratio was obtained in class ‘rest’ for kNN, SVM and ANN. Overall, Fscore ratio of class ‘6’, ‘7’, ‘8’ and ‘9’ were higher than the other classes using kNN, SVM and ANN algorithms. The DT classifiers gave highest Fscores for class ‘0’, while class ‘2’ has lowest Fscore ratio for all classifiers. As a result, by looking the Fscore ratios, the SVM classifier has highest recognition rate for all classes.

**Table 4.6 :** Recall evaluation of each class.

| Classifier | rest | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| kNN | 0.97 | 0.93 | 0.84 | 0.80 | 0.80 | 0.82 | 0.83 | 0.84 | 0.82 | 0.82 | 0.82 |
| DT | 0.80 | 0.81 | 0.73 | 0.70 | 0.72 | 0.74 | 0.76 | 0.77 | 0.76 | 0.76 | 0.76 |
| SVM | 0.99 | 0.92 | 0.88 | 0.86 | 0.87 | 0.87 | 0.87 | 0.88 | 0.87 | 0.87 | 0.87 |
| ANN | 0.95 | 0.91 | 0.85 | 0.82 | 0.84 | 0.85 | 0.85 | 0.86 | 0.85 | 0.85 | 0.85 |

**Table 4.7 :** Precision evaluation of each class.

| Classifier | rest | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| kNN | 0.64 | 0.70 | 0.68 | 0.67 | 0.70 | 0.73 | 0.75 | 0.77 | 0.77 | 0.78 | 0.79 |
| DT | 0.79 | 0.79 | 0.70 | 0.67 | 0.70 | 0.72 | 0.74 | 0.75 | 0.74 | 0.75 | 0.75 |
| SVM | 0.71 | 0.77 | 0.78 | 0.78 | 0.81 | 0.82 | 0.84 | 0.85 | 0.85 | 0.85 | 0.85 |
| ANN | 0.72 | 0.77 | 0.77 | 0.76 | 0.79 | 0.80 | 0.82 | 0.83 | 0.83 | 0.83 | 0.83 |

**Table 4.8 :** Fsocre evaluation of each class.

| Classifier | rest | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| kNN | 0.77 | 0.80 | 0.75 | 0.73 | 0.75 | 0.77 | 0.79 | 0.80 | 0.80 | 0.80 | 0.80 |
| DT | 0.79 | 0.80 | 0.72 | 0.68 | 0.71 | 0.73 | 0.75 | 0.76 | 0.75 | 0.75 | 0.75 |
| SVM | 0.83 | 0.84 | 0.83 | 0.82 | 0.83 | 0.84 | 0.85 | 0.86 | 0.86 | 0.86 | 0.86 |
| ANN | 0.82 | 0.83 | 0.80 | 0.78 | 0.81 | 0.81 | 0.83 | 0.84 | 0.84 | 0.84 | 0.84 |

# 5. CONCLUSIONS AND FUTURE WORK

In this thesis, a hand gesture recognition system was proposed. Basically, there are three methods to recognize the hand gestures. Two of them are visual based and glove based solutions. Previous work shows that they have also high recognition rate; however, there are some restrictions in usage of that methods. Thus, EMG based methods are proposed as an alternative solution. In this work, the EMG signals collected from the forearm muscles using the wearable Myo Armband. The signals were obtained during the contraction state; for this reason, the system is convenient for static hand gestures. Seven time domain features extracted from raw EMG data using sliding window approach. Then, PCA algorithm was applied to the resultant feature matrix to reduce the number of dimension. Obtained feature matrix was used to train the proposed machine learning algorithms.

Hand gesture data set which was collected for this study consists of 10 gestures from numbers between 0 and 9 in the TSL and 1 gesture from rest position. The data set was used to compare the four classifiers. The results were examined in terms of accuracy and Fscore. First, the classifiers were tested to find the optimal parameters. According that, Cityblock distance metrics with six nearest neighbor was found as optimal kNN algorithm parameters. DT algorithm was tested for splitting criteria and *crossentropy* was obtained as better. In the SVM algorithm, linear kernel with OVO method was chosen as optimal parameters. The optimal ANN was configured as chosing 12 neurons in hidden layer, *trainrp* in training function and *crossentropy* for error function. As a result, the lowest accuracy and Fscore ratio was obtained using DT classifier. kNN and ANN were improved the the performance, yet the highest results was obtained for SVM. The performances were also investigated for each class. The SVM classifier gave the highest success rate for all classes. As a result of the SVM classifier, the class ‘rest’ has high recall rate, but low precision rate. Also class ‘2’ has the lowest and class ‘6’, ‘7’, ‘8’ and ‘9’ have the highest Fscore rates.

The ultimate aim is to recognize the hand gestures. The Myo Armband located in center of this work as an EMG signal acquisition device. It is a wearable and wireless

device, which can transmit the EMG signal through Bluetooth to computer. Thus, the proposed system could be employed in various areas such as Virtual Reality, Augmented Reality and video games, that are require independence from control devices to alter the reality. In addition, the home automation could be another application area of proposed system, which leaves no need to install cameras in every room and device to control them. People can control the domestic devices like television, lights, jalousie and coffee machines simply by moving a hand. Moreover, a sign-to-speech device could be developed for the hearing impaired people using presented system.

In future work, extending the dataset with various users would be useful to validate the system. Further, number of features may be increased to have higher success rate. In addition, the system was developed for static hand gestures; therefore, the gesture should be completed before the detection. Dynamic hand gesture recognition system using the transition signal between movements could be developed by considering the methods in proposed work.

# REFERENCES


[1] **Kendon, A.** (1988). How gestures can become like words. *Cross-cultural perspectives in nonverbal communication*, *1*, 131-141.

[2] **Khan, R. Z., & Ibraheem, N. A.** (2012). Comparative study of hand gesture recognition system. In *Proc. of International Conference of Advanced Computer Science & Information Technology in Computer Science & Information Technology (CS & IT)* (Vol. 2, No. 3, pp. 203-213).

[3] **Bansal, B.** (2016). Gesture recognition: a survey. *International Journal of Computer Applications*, *139*(2).

[4] **Pavlovic, V. I., Sharma, R., & Huang, T. S.** (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on pattern analysis and machine intelligence*, *19*(7), 677-695.

[5] **LaViola, J.** (1999). A survey of hand posture and gesture recognition techniques and technology. *Brown University, Providence, RI*, *29*.

[6] **Url-1** <https://developer.microsoft.com/en-us/windows/kinect>, date retrieved 15.04.2018

[7] **Url-2** <http://www.cyberglovesystems.com/>, date retrieved 15.04.2018

[8] **Url-3** <https://www.leapmotion.com/>, date retrieved 15.04.2018

[9] **Url-4** <https://www.myo.com/> date retrieved 10.01.2018

[10] **Barreto, A. B., Scargle, S. D., & Adjouadi, M.** (2000). A practical EMG-based human-computer interface for users with motor disabilities. *Journal of rehabilitation research and development*, *37*(1), 53.

[11] **Kosmidou, V. E., Hadjileontiadis, L. J., & Panas, S. M.** (2006, August). Evaluation of surface EMG features for the recognition of American Sign Language gestures. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE* (pp. 6197-6200). IEEE.

[12] **Katz, D., Hafsia, L. B., Salem, O., & Mehaoua, A.** (2015). Intelligent remote control of smart home devices using physiological parameters. In *E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on* (pp. 280-285). IEEE.

[13] **Doswald, A.** (2013). *Using biosignals to control the Nao robot*. (Master dissertation)

[14] **Url-5** <https://wfdeaf.org/our-work/> date retrieved 21.04.2018

[15] **Taylor, B.** (2016). Towards the Automatic Translation of American Sign Language. *Human-Computer Interaction*. Thesis proposal

[16] **Url-6** <http://www.nickgillian.com/archive/teaching/workshops/mitiap2013/IAP_Session1.pdf> date retrieved 21.04.2018

[17] **Li, X.** (2003). Gesture recognition based on fuzzy C-Means clustering algorithm. *Department Of Computer Science The University Of Tennessee Knoxville*.

[18] **Santos, B. S., Cardoso, J., Ferreira, B. Q., Ferreira, C. and Dias, P.** (2016). Developing 3D Freehand Gesture-Based Interaction Methods for Virtual Walkthroughs: Using an Iterative Approach. In *Handbook of Research on Human-Computer Interfaces, Developments, and Applications* (pp. 52-72). IGI Global.

[19] **Liang, R. H., & Ouhyoung, M.** (1998). A real-time continuous gesture recognition system for sign language. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on* (pp. 558-567). IEEE.

[20] **Verma, M. P., SL, M. S., & Chatterji, S.** (2014). Design of Smart Gloves. *International Journal of Engineering Research and Technology (IJERT) ISSN*, 2278-0181.

[21] **Lamberti, L., & Camastra, F.** (2011, September). Real-time hand gesture recognition using a color glove. In *International Conference on Image Analysis and Processing* (pp. 365-373). Springer, Berlin, Heidelberg.

[22] **Maraqa, M., & Abu-Zaiter, R.** (2008, August). Recognition of Arabic Sign Language (ArSL) using recurrent neural networks. In *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the* (pp. 478-481). IEEE.

[23] **Diaz, C. and Payandeh, S.** (2014). Preliminary experimental study of marker-based hand gesture recognition system. *Journal of Automation and Control Engineering Vol*, *2*(3).

[24] **Du, H. and To, T.** (2011). Hand gesture recognition using Kinect. *Techical Report, Boston University*.

[25] **Dinh, D. L., Lee, S. and Kim, T. S.** (2016). Hand number gesture recognition using recognized hand parts in depth images. *Multimedia Tools and Applications*, *75*(2), 1333-1348.

[26] **Lu, W., Tong, Z., & Chu, J.** (2016). Dynamic hand gesture recognition with leap motion controller. *IEEE Signal Processing Letters*, *23*(9), 1188-1192.

[27] **Wen, H., Ramos Rojas, J., & Dey, A. K.** (2016, May). Serendipity: Finger gesture recognition using an off-the-shelf smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 3847-3851). ACM.

[28] **Mills, K. R.** (2005). The basics of electromyography. *Journal of Neurology, Neurosurgery & Psychiatry*, *76*(suppl 2), ii32-ii35.

[29] **Bigland, B., & Lippold, O. C. J.** (1954). Motor unit activity in the voluntary contraction of human muscle. *The Journal of Physiology*, *125*(2), 322-335.

[30] **Freund, H. J.** (1983). Motor unit and muscle activity in voluntary motor control. *Physiological Reviews*, *63*(2), 387-436.

[31] **Konrad, P.** (2005). The abc of emg. *A practical introduction to kinesiological electromyography*, *1*, 30-35.

[32] **Tassinary, L. G., Cacioppo, J. T. and Vanman, E. J.** (2007). The skeletomotor system: Surface electromyography, *Handbook of psychophysiology*, Second edition.

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

[34] **De Luca, C. J., Adam, A., Wotiz, R., Gilmore, L. D. and Nawab, S. H.** (2006). Decomposition of surface EMG signals. *Journal of neurophysiology*, *96*(3), 1646-1657.

[35] **De Luca, C. J.** (1979). Physiology and mathematics of myoelectric signals. *IEEE Transactions on Biomedical Engineering*, (6), 313-325.

[36] EMG. (n.d.) *Mosby's Medical Dictionary, 8th edition*. (2009), date retrieved 21.04.2018 from https://medical-dictionary.thefreedictionary.com

[37] **Scimè, Anthony & Z Caron, Annabelle & Grenier, Guillaume.** (2009). Advances in myogenic cell transplantation and skeletal muscle tissue engineering. *Frontiers in bioscience*: 14, 3012-3023

[38] myofiber. (n.d.) *Medical Dictionary*. (2009). date retrieved 18.02.2018 from https://medical-dictionary.thefreedictionary.com/myofiber

[39] **Merletti, R., & Di Torino, P.** (1999). Standards for reporting EMG data. *J Electromyogr Kinesiol*, *9*(1), 3-4.

[40] **Url-7** <https://www1.udel.edu/biology/rosewc/kaap686/notes/EMG% 20analysis.pdf> date retrieved 10.02.2018

[41] **Url-8** <https://www.myo.com/techspecs> date retrieved 10.01.2018

[42] **Url-9**<https://developer.thalmic.com/docs/api_reference/platform/ index.html> date retrieved 10.01.2018

[43] **Côté-Allard, U., St-Onge, D., Giguère, P., Laviolette, F. and Gosselin, B.** (2017). Towards the use of consumer-grade electromyographic armbands for interactive, artistic robotics performances. In *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on* (pp. 1030-1036). IEEE.

[44] **Subbu, R., Weiler, R., & Whyte, G.** (2015). The practical use of surface electromyography during running: does the evidence support the hype? A narrative review. *BMJ open sport & exercise medicine*, *1*(1), e000026.

[45] **Sueaseenak, D., Khawdee, C., Pakornsirikul, N. and Sukjamsri, C.** (2017). A performance of modern gesture control device with application in pattern classification. In *Control, Automation and Robotics (ICCAR), 2017 3rd International Conference on* (pp. 428-431). IEEE.

[46] **Benalcázar, M. E., Jaramillo, A. G., Zea, A., Páez, A. and Andaluz, V. H.** (2017). Hand gesture recognition using machine learning and the Myo armband. In *Signal Processing Conference (EUSIPCO), 2017 25th European* (pp. 1040-1044). IEEE.

[47] **Atasoy, A., Kaya, E., Toptas, E., Kuchimov, S., Kaplanoglu, E. and Ozkan, M.** (2016). 24 DOF EMG controlled hybrid actuated prosthetic hand. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the* (pp. 5059-5062). IEEE.

[48] **Demirel U.** (2017) Creating A Generic Hand and Finger Gesture Recognizer by Using Forearm Muscle Activity Signals, (Master dissertation)

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

[50] **Raina, A., Lakshmi, T. G., & Murthy, S.** (2017). CoMBaT: Wearable Technology Based Training System for Novice Badminton Players.

In *Advanced Learning Technologies (ICALT), 2017 IEEE 17th International Conference on* (pp. 153-157). IEEE.

[51] **Ploengpit, Y., & Phienthrakul, T.** (2016). Rock-paper-scissors with Myo Armband pose detection. In *Computer Science and Engineering Conference (ICSEC), 2016 International* (pp. 1-5). IEEE.

[52] **Krishnan, K. S., Saha, A., Ramachandran, S., & Kumar, S.** (2017). Recognition of human arm gestures using Myo armband for the game of hand cricket. In *Robotics and Intelligent Sensors (IRIS), 2017 IEEE International Symposium on* (pp. 389-394). IEEE.

[53] **Luh, G. C., Lin, H. A., Ma, Y. H., & Yen, C. J.** (2015). Intuitive muscle-gesture based robot navigation control using wearable gesture armband. In *Machine Learning and Cybernetics (ICMLC), 2015 International Conference on* (Vol. 1, pp. 389-395). IEEE.

[54] **Güngör C., Bağrıaçık T., Demirdöğen İ., Günaydın A. and Karahan V.** (2015) *Türk İşaret Dili Sözlüğü*, Ankara

[55] **Url-10** <https://support.getmyo.com/hc/en-us/articles/201169525-How-to-wear-the-Myo-armband> date retrieved 10.01.2018

[56] **Url-11** <https://www.mathworks.com/matlabcentral/fileexchange/55817-myo-sdk-matlab-mex-wrapper> date retrieved 10.01.2018

[57] **Katsaounidou, A. N. and Dimoulas, C. A.** (2018). Integrating Content Authentication Support in Media Services. In *Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 2908-2919). IGI Global.

[58] **Shalev-Shwartz, S. and Ben-David, S.** (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

[59] **Mohri, M., Rostamizadeh, A. and Talwalkar, A.** (2012). *Foundations of machine learning*. MIT press.

[60] **Alpaydin, E.** (2014). *Introduction to machine learning*. MIT press.

[61] **Kılıç, E. and Doğan, E.** (2016, May). Real-time feature extraction from EMG signals. In *Signal Processing and Communication Application Conference (SIU), 2016 24th* (pp. 113-116). IEEE.

[62] **Phinyomark, A., Phukpattaranont, P. and Limsakul, C**. (2012). Feature reduction and selection for EMG signal classification. *Expert Systems with Applications*, *39*(8), 7420-7431.

[63] **Boostani, R. and Moradi, M. H.** (2003). Evaluation of the forearm EMG signal features for the control of a prosthetic hand. *Physiological measurement*, *24*(2), 309.

[64] **Veer, K. and Sharma, T.** (2016). A novel feature extraction for robust EMG pattern recognition. *Journal of medical engineering & technology*, *40*(4), 149-154.

[65] **Holmgaard, S. and Nielsen, J. L.** Classification of non- stationary surface EMG signals for controlling myoelectric prostheses.

[66] **Keogh, E., Chu, S., Hart, D., & Pazzani, M.** (2004). Segmenting time series: A survey and novel approach. In *Data mining in time series databases* (pp. 1-21).

[67] **Ghodsi, A.** (2006). Dimensionality reduction a short tutorial. *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada*, *37*, 38.

[68] **Shlens, J.** (2014). A tutorial on principal component analysis.

[69] **Domingos, P.** (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78-87.
[70] **Cover, T. and Hart, P.** (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.
[71] **Friedman, J., Hastie, T. and Tibshirani, R.** (2001). *The elements of statistical learning* (Vol. 1, pp. 337-387). New York: Springer series in statistics.
[72] **Cha, S. H.** (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City*, *1*(2), 1.
[73] **C.L. Blake and C.J. Merz** (1998). UCI repository of machine learning databases. University of California. [http://www.ics.uci.edu/~mlearn/MLRepository.html]
[74] **Quinlan, J. R.** (1986). Induction of decision trees. *Machine learning*, *1*(1), 81-106.
[75] **Quinlan, J. R.** (2014). *C4. 5: programs for machine learning*. Elsevier.
[76] **Breiman, L.** (2017). *Classification and regression trees*. Routledge.
[77] **Mitchell, T. M.** (1997). Machine learning, ser. *Computer Science Series. Singapure: McGraw-Hill Companies, Inc*.
[78] **Berzal, F., Cubero, J. C., Cuenca, F. and Martín-Bautista, M. J.** (2003). On the quest for easy-to-understand splitting rules. *Data & Knowledge Engineering*, *44*(1), 31-48.
[79] **Url-12** <https://www.mathworks.com/help/stats/fitctree.html> date retrieved 20.03.2018
[80] **Ng, A.** (2000). CS229 Lecture notes. *CS229 Lecture notes*, *1*(1), 1-3.
[81] **Fletcher, T.** (2009). Support vector machines explained. *Tutorial paper*.
[82] **Narsky, I. and Porter, F. C.** (2013). Multiclass Extensions of Support Vector Machines.
[83] **McCulloch, W. S. and Pitts, W.** (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115-133.
[84] **Krenker, A., Bester, J. and Kos, A.** (2011). Introduction to the artificial neural networks. In *Artificial neural networks-methodological advances and biomedical applications*. InTech.
[85] **LeCun, Y., Bottou, L., Orr, G. B. and Müller, K. R.** (1998). Efficient backprop. In *Neural networks: Tricks of the trade* (pp. 9-50). Springer, Berlin, Heidelberg.
[86] **Sadowski, P.** (2016). Notes on backpropagation. *homepage: https://www. ics. uci. edu/~ pjsadows/notes. pdf (online)*.
[87] **Popescu, M. C., Balas, V. E., Perescu-Popescu, L. and Mastorakis, N.** (2009). Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, *8*(7), 579-588.
[88] **Cömert, Z. and Kocamaz, A. F.** (2017). A study of artificial neural network training algorithms for classification of cardiotocography signals. *Bitlis Eren University Journal of Science and Technology*, *7*(2), 93 -103.
[89] **Golik, P., Doetsch, P. and Ney, H.** (2013, August). Cross-entropy vs. squared error training: a theoretical and experimental comparison. In *Interspeech* (Vol. 13, pp. 1756-1760).
[90] **Ruder, S.** (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[91] **Shaharin, R., Prodhan, U. K. and Rahman, M.** (2014). Performance study of TDNN training algorithm for speech recognition. *Int. J. Adv. Res. Comput. Sci. Technol*, *2*, 90-95.

[92] **Cilimkovic, M.** (2015). Neural networks and back propagation algorithm. *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, *15*.

[93] **Gershenson, C.** (2003). Artificial neural networks for beginners. *arXiv preprint cs/0308031*.

[94] **Anastasiadis, A. D., Magoulas, G. D. and Vrahatis, M. N.** (2005). New globally convergent training scheme based on the resilient propagation algorithm. *Neurocomputing*, *64*, 253-270.

[95] **Riedmiller, M. and Rprop, I.** (1994). Rprop-description and implementation details.

[96] **Shaharin, R., Prodhan, U. K. and Rahman, M.** (2014). Performance study of TDNN training algorithm for speech recognition. *Int. J. Adv. Res. Comput. Sci. Technol*, *2*, 90-95.

[97] **Kişi, Ö. and Uncuoğlu, E.** (2005). Comparison of three back-propagation training algorithms for two case studies.

[98] **Baghirli, O.** (2015). Comparison of Lavenberg-Marquardt, Scaled Conjugate Gradient And Bayesian Regularization Backpropagation Algorithms for Multistep Ahead Wind Speed Forecasting Using Multilayer Perceptron Feedforward Neural Network.

[99] **Sharma, B. and Venugopalan, K.** (2014). Comparison of neural network training functions for hematoma classification in brain CT images. *IOSR-JCE*, *16*(1), 31-35.

[100] **Møller, M. F.** (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, *6*(4), 525-533.

[101] **Sokolova, M. and Lapalme, G.** (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427-437.

[102] **Kelleher, J. D., Mac Namee, B. and D'Arcy, A.** (2015). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT Press.

# CURRICULUM VITAE

**Name Surname** **: Engin KAYA**

**Place and Date of Birth** **: Konak, 05.03.1992**

**E-Mail** **: kayaengn@gmail.com**

**EDUCATION** **:**

- **B.Sc.** **:** 2015, Istanbul Technical University, Control and Automation Engineering

**PROFESSIONAL EXPERIENCE AND REWARDS:**

- 04.2017 – Present The Scientific and Technological Research Council of Turkey (TÜBİTAK)
- 05.2016 – 07.2016 Visiting Researcher, University of Coimbra, Portugal
- 10.2015 – 05.2016 Graduate Research Assistant, Robotics Lab, Institute of Biomedical Engineering, Bogazici University

**OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:**

- Atasoy A., **Kaya E.**, Toptas E., Kuchimov, S., Kaplanoglu E., and Ozkan M., (2016), 24 DOF EMG controlled hybrid actuated prosthetic hand. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the* (pp. 5059-5062). IEEE.