YAŞAR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR EEG BASED CONTROLLING OF A ROBOTIC ARM

GÜLŞEN AYLUÇTARHAN

THESIS ADVISOR: ASST. PROF. DR. İBRAHİM ZİNCİR

MASTER OF SCIENCE IN COMPUTER ENGINEERING

PRESENTATION DATE: 13.12.2019

BORNOVA / İZMİR
DECEMBER 2019

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

Jury Members:	**Signature:**
Asst. Prof. Dr. İbrahim ZİNCİR Yasar University	....
Asst. Prof. Dr. Nalan ÖZKURT Yasar University	....
Asst. Prof. Dr. Samsun M. BAŞARICI Adnan Menderes University	....

Prof. Dr. Cüneyt GÜZELİŞ

Director of the Graduate School

ABSTRACT

IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR EEG BASED CONTROLLING OF A ROBOTIC ARM

AYLUCTARHAN, GULSEN
MSc, Computer Engineering
Advisor: Asst. Prof. Dr. Ibrahim ZINCIR
December 2019

Electroencephalography (EEG) analysis has been an important subject of several studies like neuroscience, medical diagnosis and rehabilitation engineering. EEG is widely used with brain-computer interface (BCI) systems because of its ability to use brain signals not muscles to control an external BCI prosthetic device. With the development of technology, it became possible to use large EEG datasets and BCI method to extract an understandable information. In this present work, EEG-based BCI system is used by making participants perform a series of grasping and lifting hand movements. Dataset which consists of EEG and EMG information has been implemented via 15 machine learning algorithms as multiclass classification. The best results came from IB1 algorithm. But, random forest, bagging and classification via regression algorithms also have promising outcomes. Hence, this study successfully proved that it is possible to help patients with no hand function to gain control.

Key Words: electroencephalography (EEG), brain-computer interface (BCI), machine learning, neuroscience, technology

ÖZ

EEG TABANLI ROBOTİK KOL KONTROLÜ İÇİN MAKİNE ÖĞRENME ALGORİTMALARININ UYGULANMASI

AYLUÇTARHAN, GÜLŞEN
Yüksek Lisans, Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Danışman: Yrd. Doç. Dr. İbrahim ZİNCİR
Aralık 2019

Elektroensefalografi (EEG) analizi; sinir bilimi, tanı ve rehabilitasyon mühendisliği gibi birçok çalışmanın önemli bir konusu olmuştur. EEG'nin büyük ölçüde beyin-bilgisayar ara yüzü (BCI) ile kullanılmasının nedeni bu sistemin dış protez cihazlarını kaslar yerine beyin dalgaları ile kullanabilme yeteneğidir. Teknolojinin gelişmesi ile, anlaşılabilir bir bilgi çıkarmak için daha büyük EEG veri tabanları ve BCI metodu kullanmak mümkün hale gelmiştir. Bu çalışmada, katılımcılara bir seri kavrama ve kaldırma el hareketleri yaptırılarak EEG tabanlı BCI metodu kullanıldı. EEG ve EMG bilgisinden oluşan veri 15 makine öğrenimi algoritmasıyla çok sınıflı sınıflandırma kullanılarak uygulandı. En iyi sonuçlar IB1 algoritmasından geldi. Ancak, random forest, bagging ve classification via regression algoritmaları da umut verici sonuçlar gösterdi. Böylece bu çalışma başarılı bir şekilde el fonksiyonu çalışmayan hastaların kontrol kazanmasına yardımcı olmanın mümkün olduğunu kanıtladı.

Anahtar Kelimeler: Elektroansefalografi (EEG), beyin-bilgisayar ara yüzü (BCI), makine öğrenimi, sinir bilimi, teknoloji

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor Ibrahim ZINCIR for his guidance and patience during this study.

I would like to express my enduring love to my dear friend Dilda ERBAKAN, who is always supportive, loving and caring to me in every possible way in my life.

And my biggest gratitude is to my mother. She always gives me strength, love and happiness. I could not finish this project without her.

Gülşen Ayluçtarhan

İzmir, 2019

TEXT OF OATH

I declare and honestly confirm that my study, titled “IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR EEG BASED CONTROLLING OF A ROBOTIC ARM” and presented as a master’s Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Gülşen Ayluçtarhan

Signature

…………………………………..

December 13, 2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS:

ABBREVIATIONS:

EEG Electroencephalogram

BCI Brain-Computer Interface

EMG Electromyography

ERD Event-Related Desynchronization

SVM Support Vector Machine

WAY Wearable Interfaces for Hand Function Recovery

GAL Grasp And Lift

J48 Decision Tree

kNN k-Nearest Neighbor

SMO Sequential Minimal Optimization

OP Quadratic Programming

WEKA Waikato Environment for Knowledge Analysis

ARFF Attribute Relation File Format

TP True Positive

ROC Receiver Operating Characteristic

CHAPTER 1
INTRODUCTION

People use their hands for nearly everything during daily life. For example, once you get up in the morning you turn off the alarm then brush your teeth then have breakfast and then get dressed before you leave from your home for your work. Imagine doing all those things without using your hands. Many people who have lost their hand/hands either as a result of medical problems, congenital complications or accidents have to deal with this kind of situation in their daily life. For them, depending on the complexity of the condition, these routine activities become impossible to do. It is believed that regaining the ability to perform their daily activities via using a brain-computer interface (BCI) prosthetic device would significantly increase their quality of life.

An electroencephalogram (EEG) may be an accustomed check to realize problems regarding the electrical activity of the brain. It is one of the main diagnostic tests for brain disorders. An EEG graph simply tracks and records the brain wave patterns. In order to achieve this goal tiny metal discs placed on the scalp connected with electrodes, send signals to a computer. The results, which are the extracted signals, are then analyzed via implementing different signal processing techniques so that the system can differentiate signals related to eye blinking, noise, temperature and etc. Once filtered the signal could then be analyzed for the desired signal features. Multiple machine learning techniques are preferred for this classification process (Ramzan & Dawn, 2019).

BCI is a favorable technique for generating a real time link between human brain and an external device. Although, there are many different brain wave recording techniques to measure the electrical activity of brain to use in BCI implementations, scalp graph is accepted as the most successful one amongst the others as a result of its non-invasive nature and straightforward usage (Islam & Rastegarnia, 2019).

EEG based BCI mechanisms guarantee an improved life standard for disabled patients without the risks of surgical procedures. These patients benefit from the BCI devices for limb or wheelchair control via regulation of EEG signals during movement related tasks.

The working principle of any EEG based BCI systems shown in the Figure 1. First step is data acquisition. Varying with the purpose of the study, data are collected using an EEG cap from the participants in different ways. Like; speech, movement and capturing intention. After signals are recorded, extracting some useful information is in order. Fourier transform is the way to turn discrete EEG signals into applicable continuous signals. At last, classification methods are applied according to the application areas.

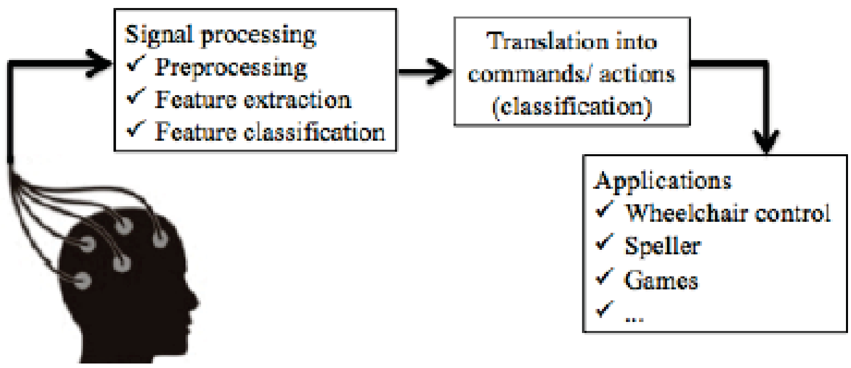


Figure 1 Functional Model of BCI (Chenane & Touati, 2018)

CHAPTER 2
LITERATURE REVIEW

Alomari et al. proposed an automated pc framework for classifying EEG signals related to left hand and right hand actions manipulating a hybrid system that inherits machine learning algorithms and other feature extraction techniques (Alomari, Samaha & AlKamha, 2013). They tried to find out which hand was used during their experiments via implementing artificial neural networks and support vector machines algorithms. Their results presented that a BCI context could be implemented via using brainwaves to manage a machine.

Shimizu et al. aimed to rebuild the control system of the brain and therefore the body for stroke rehabilitation (Shimizu, Dobashi, Ito & Nagai, 2014). In order to achieve this goal, they propose a new BCI method applying EEG signals to find motion characteristics for unfit components of stroke patients. Proposed system reduced the influence of noises, handled the differences of people and successfully proved with experiments.

Robinson et al. studied on a thought-controlled device using a business affordable EEG electronic equipment to implement and to control a BCI system. Their aim was to extract brain waves to understand motor activities and movement directions that may be accustomed to establish the motor tasks performed by the user. It was stated that this EEG BCI system could be used to add motor control to a robotic device (Robinson & Vinod, 2015).

Norman et al. investigated how event-related desynchronization (ERD) changed as a part of audio visual stimuli, specific movement from the user, and robotic implementations. The research inherits a robotically aided medical care for the people with disabilities. In their experiment, twelve subjects played a video game specifically designed for medical rehabilitation purposes via using a finger eccentric skeleton system (Norman, Dennison, Wolbrecht, Cramer, Srinivasan &

Reinkensmeyer, 2016).

Zhang et al. proposed a novel framework of a hybrid EEG based robotics system to unravel grasping hand movements. They asked participants to perform a series of grasp an object tasks and achieved an effective performance via implementing a BCI system that logs brainwaves while doing specific hand movements such as grasping (Zhang, Sun, Liu, Su, Tan & Liu, 2017).

Yavuz and Aydemir studied a new EEG data set recorded from 29 people during imagination of hand opening/closing movement (Yavuz & Aydemir, 2017). The focus of the study to combine EEG and BCI systems to ease the lives of paralyzed patients who do not have any problems with their cognitive functioning. As a machine learning technique, k-nearest neighbor method was preferred. The results showed a high performance in general.

Alazrai et al. presented an EEG based BCI system to classify basic motor representational processes of identical hands. They achieved this via implementing empirical mode decomposition technique. 18 participants were used to make them perform wrist, finger and grasp related tasks. Their results showed the effectiveness of the proposed framework via analyzing EEG signals (Alazrai, Aburub, Fallouh & Daoud, 2017).

Cho et al. analyzed the recording of five different hand movements from EEG signals. There were five subjects who performed hand motions. The proposed framework searched for similar spatial patterns and used regular linear discriminant analysis techniques for to understand singularities. The outcome was encouraging to be used for a BCI-driven automaton hand management (Cho, Jeong, Shim, Kim & Lee, 2018).

Butt et al. proposed a hand rehabilitation robotic device mainly based on an EEG BCI system to study the technical practicability of in hand motor recovery of post-stroke patients. They preferred 2D games to make participants perform hand movements. Support vector machine was the only machine learning algorithm in this study. The outcome showed that such gaming scenarios should be used for stroke patients (Butt, Naghdy, Naghdy, Murray & Du, 2018).

Alazrai et al. presented a novel EEG based BCI system to decipher the movements of each finger via studying the EEG signals. They implemented quadratic time-frequency distributions and achieved successful results while predicting the hand movements. EEG signals analyzed to capture movements. 18 subjects performed 12 finger actions utilizing their correct hands. Multi-class SVM classifier approach provided higher control on robotic devices (Alazrai, Alwanni & Daoud, 2019).

As you can see through the aforementioned references, EEG based BCI systems preferred and advanced over the years. These are just the papers that focused on hand movements and used machine learning algorithms.

CHAPTER 3
DATA PREPARATION

The database used in the project is called WAY-EEG-GAL (WAY: Wearable interfaces for hAnd function recoverY, EEG: Electroencephalography, GAL: Grasp And Lift). The data is collected from twelve participants by making them perform a series of specific hand movements. Twelve participants executed a series movements during which they lifted a group of objects that weighted 165, 330, and 660 grams. In addition, they were asked to release the objects to over different surfaces from time to time. To complete the every trial, the participant had to cue for the item, then grasp it with the thumb and forefinger, and then raise and hold it for at least one or two of seconds, and finally place it back on the support surface and leave it there as explained in Table 3.1. 32 channels of EEG, EMG of five arm and hand muscles, the 3D position of each the hand and object, and force/torque at each contact plates were recorded (Luciw, Jarocka & Edin, 2014).

Table 3.1 Steps of GAL

EVENT	MEANING
Lift	The object lifted from the support
Replace	The object replaced on the support
Release	Both fingers release the object

The GAL have been repeated on and on for hundreds of times and it is called as series and the data are stored in one csv file. Every participant performed eight series resulting eight csv files. In all trials, the specific object to be lifted were changed via an unpredictable manner to the participant with reference to the weight (165, 330, or

660 grams), contact surface (sandpaper, suede, or silk), or both (Luciw, Jarocka & Edin, 2014).

Each subject had two kinds of csv files named like subjNumber_seriesNumber_data and subjNumber_seriesNumber_event. For example; subj1_series1_data.csv and subj1_series1_event.csv. There are eight data and its corresponding eight event files, in all 16 files, for each person. Data files hold the EEG cap information, 32 columns which comes from 32 electrodes, and id number(number of the row).

Table 3.2 Column Information of Data Files

Columns	Minimum Value	Maximum Value
id	subj1_series1_0	subj12_series8_154679
Fp1	-4286	17307
Fp2	-3353	4066
F7	-2834	3503
F3	-1685	1378
Fz	-1315	1044
F4	-1889	1816
F8	-5746	5058
FC5	-1968	1823
FC1	-746	1211
FC2	-1437	1323
FC6	-3287	2637
T7	-3472	5610

C3	-679	1983
Cz	-765	1291
C4	-2133	2554
T8	-5297	9550
TP9	-3840	7323
CP5	-1386	1206
CP1	-1032	1757
CP2	-1177	1552
CP6	-1683	1880
TP10	-3299	3714
P7	-2411	2369
P3	-1315	1323
Pz	-1684	1351
P4	-1289	1515
P8	-1419	2948
PO9	-3678	3806
O1	-1393	1346
Oz	-1627	1334
O2	-1526	1483
PO10	-2436	4952

Event files consist of GAL information in 3 columns and id column.

Table 3.3 Column Information of Event Files

Columns	Values
Id	subj1_series1_0 – subj12_series8_154679
Lift	0 or 1
Replace	0 or 1
Release	0 or 1

Figure 2 and Figure 3 show some part of the chosen files for understanding the concept.

id	Fp1	Fp2	F7	F3	Fz	F4	F8	FC5	FC1	FC2	FC6	T7	C3	Cz	C4	T8	TP9	CP5	CP1	CP2	CP6	TP10	P7	P3	Pz	P4	P8	PO9	O1	Oz	O2	PO10
subj1_series1_0	-31	363	211	121	211	15	717	279	35	158	543	-166	192	230	573	860	128	59	272	473	325	379	536	348	383	105	607	289	459	173	120	704
subj1_series1_1	-29	342	216	123	222	200	595	329	43	166	495	-138	201	233	554	846	185	47	269	455	307	368	529	327	369	78	613	248	409	141	83	737
subj1_series1_2	-172	278	105	93	222	511	471	280	12	177	534	-163	198	207	542	768	145	52	250	452	273	273	511	319	355	66	606	320	440	141	62	677
subj1_series1_3	-272	263	-52	99	208	511	428	261	27	180	525	-310	212	221	542	808	115	41	276	432	258	241	521	336	356	71	568	339	437	139	58	592
subj1_series1_4	-265	213	-67	99	155	380	476	353	32	165	507	-320	242	230	545	865	180	89	288	444	275	275	550	324	346	76	547	343	446	171	67	581
subj1_series1_5	-254	135	45	78	152	246	490	414	30	176	510	-197	209	224	560	834	278	60	283	469	288	347	586	343	363	74	580	389	471	190	100	661
subj1_series1_6	-156	103	295	105	177	330	429	499	48	156	474	-131	206	189	518	558	301	45	265	416	237	351	579	339	341	29	530	422	467	180	98	599
subj1_series1_7	-23	175	381	148	198	412	490	539	44	155	486	-126	215	179	518	378	269	37	258	410	302	370	581	324	331	45	573	402	446	213	122	695
subj1_series1_8	43	354	382	147	217	569	591	529	57	153	545	-177	212	183	587	591	219	37	246	423	399	263	506	279	306	44	554	250	340	69	-4	560
subj1_series1_9	99	518	320	178	233	650	568	482	75	169	543	-113	224	178	565	895	208	53	257	423	284	241	523	307	310	43	511	292	374	79	8	609
subj1_series1_10	89	518	258	98	219	600	529	346	60	139	491	-175	179	172	511	953	228	8	257	411	137	261	553	326	338	13	464	357	439	175	74	657
subj1_series1_11	104	488	274	200	247	501	503	290	86	161	468	-230	184	198	532	872	180	4	278	443	197	167	545	358	363	33	448	346	475	175	66	543
subj1_series1_12	76	417	247	214	221	397	430	301	80	136	452	-223	216	200	529	789	171	22	268	439	228	202	505	345	360	80	512	285	477	215	115	734
subj1_series1_13	94	363	257	257	211	306	420	353	98	121	444	-141	202	192	502	733	110	11	237	408	164	179	472	306	318	-8	378	205	361	68	-58	426
subj1_series1_14	76	401	212	333	253	395	447	374	69	150	510	-267	203	195	504	723	48	-8	239	409	187	183	454	285	317	-20	364	179	331	44	-78	390
subj1_series1_15	27	362	268	309	261	502	406	379	70	153	617	-352	200	189	513	679	60	3	241	407	198	204	478	292	314	-17	386	264	393	104	-23	419
subj1_series1_16	34	336	371	310	269	515	432	421	48	147	588	-197	201	177	500	653	89	41	247	411	170	183	508	319	315	-31	394	301	416	98	-14	455
subj1_series1_17	106	408	424	319	250	395	600	438	46	111	378	-58	211	171	474	732	200	73	240	400	171	273	524	342	323	-25	438	329	447	124	16	580
subj1_series1_18	144	592	415	338	260	467	642	433	53	159	525	-35	225	179	525	813	167	45	253	428	206	301	519	321	321	-9	454	227	410	103	-15	550
subj1_series1_19	249	548	451	237	247	494	597	461	52	160	556	-84	230	199	531	789	223	59	265	428	225	238	571	327	341	4	467	287	426	129	3	564
subj1_series1_20	269	391	377	126	227	402	513	476	60	160	532	-122	223	180	527	771	237	76	276	420	227	247	616	353	344	12	513	337	482	178	65	632

Figure 2 Example Data From subj1_series1_data.csv

id	Lift	Replace	Release
subj1_series1_0	0	0	0
subj1_series1_1	0	0	0
subj1_series1_2	0	0	0
subj1_series1_3	0	0	0
subj1_series1_4	0	0	0
subj1_series1_5	0	0	0
subj1_series1_6	0	0	0
subj1_series1_7	0	0	0
subj1_series1_8	0	0	0
subj1_series1_9	0	0	0
subj1_series1_10	0	0	0
subj1_series1_11	0	0	0
subj1_series1_12	0	0	0
subj1_series1_13	0	0	0
subj1_series1_14	0	0	0
subj1_series1_15	0	0	0
subj1_series1_16	0	0	0
subj1_series1_17	0	0	0
subj1_series1_18	0	0	0
subj1_series1_19	0	0	0
subj1_series1_20	0	0	0

Figure 3 Example Data From subj1_series1_event.csv

Id columns are removed from both data and event files. Next, the data files of eight series have been united into one csv file like subj1_data. Naturally, the same process has been applied on the event files. Then, we have added the three columns of event files to the data files. Eventually, we formed that one file for each subject which possesses all the necessary information.

Event files have lift, replace and release columns consisting of zeros and ones, presenting that whether the corresponding event has occurred or not between ±150ms. Since these events are performed in order, there cannot be two ones in a row. Before the trials begin, there is a preparation process. As a result of this, first rows have zero values but those values cannot be used for analysis hence they are excluded. Searching through the rows, it is observed that there are three specific rows. And it is expected because there are three columns and they all must have one value for at least one time but two of them cannot have one value at the same time. And that situation creates only three patterns. After that, it is decided that these rows are going to be used to create a nominal value. The nominal column is called LRR (lift, replace and release) and it has one, two and three values. It was easy to convert this three-column-information into one nominal value as can be seen in Table 3.2.

Table 3.4 LRR Pattern

Lift	Replace	Release	LRR
0	0	1	1
0	1	0	2
1	0	0	3

After preparing the nominal class value LRR, it is added to the data as the last column and the final version of the file is created like shown in Figure 4.

Fp1	Fp2	F7	F3	Fz	F4	F8	FC5	FC1	FC2	FC6	T7	C3	Cz	C4	T8	TP9	CP5	CP1	CP2	CP6	TP10	P7	P3	Pz	P4	P8	PO9	O1	Oz	O2	PO10	LRR
1153	-66	1210	358	469	832	358	807	421	365	166	346	529	615	-72	91	393	-194	420	200	181	559	-131	143	367	197	-434	1469	110	473	-94	-26	3
1178	-56	1207	356	464	827	527	781	416	385	158	252	542	634	-10	125	462	-204	443	219	220	605	-26	166	407	238	-386	1562	170	533	-30	89	3
1123	-59	1179	364	459	821	362	786	429	408	216	103	547	654	35	197	352	-228	458	238	236	583	-114	151	399	253	-309	1538	164	542	-52	24	3
1070	-112	1175	334	450	817	333	731	408	369	224	192	535	645	-102	184	244	-225	449	201	179	546	-259	129	409	197	-289	1498	126	484	-108	-168	3
1066	-101	1152	326	453	817	344	615	397	383	271	304	450	667	-118	4	256	-294	402	214	115	522	-206	109	418	154	-411	1434	101	485	-104	-117	3
1056	-84	1089	347	448	796	309	659	431	392	178	289	451	654	-69	76	273	-275	426	210	147	471	-172	120	413	174	-448	1363	99	482	-85	18	3
1047	-118	1147	368	429	792	324	694	436	379	185	263	487	650	-66	135	256	-288	440	201	178	463	-171	134	414	192	-359	1419	119	465	-93	-56	3
1083	-96	1167	353	432	806	341	689	425	383	175	255	491	659	-70	99	289	-298	413	205	187	509	-208	103	420	181	-395	1477	155	474	-128	-105	3
1092	-62	1148	311	432	799	345	697	413	369	159	233	490	643	-72	160	244	-280	416	225	185	524	-224	124	420	176	-417	1469	164	509	-104	-51	3
1100	-63	1159	340	439	802	335	708	411	362	198	235	494	647	-53	234	216	-212	439	239	176	528	-141	182	425	211	-358	1514	207	560	-48	-71	3
1124	-38	1123	357	464	836	384	708	433	397	255	324	519	649	-46	241	239	-198	448	231	202	507	-100	184	436	217	-312	1431	191	562	-54	-61	3
1100	-45	1117	330	484	878	395	694	419	411	243	243	472	609	-27	151	309	-326	402	209	229	480	-273	130	416	171	-321	1544	195	557	-121	-102	3
1081	-30	1149	340	454	889	385	696	403	393	282	210	465	633	-39	70	314	-311	395	174	168	497	-166	142	370	122	-452	1568	225	576	-119	-57	3
1093	-33	1131	347	449	887	383	675	414	428	308	182	471	658	-4	-66	246	-308	411	198	206	540	-207	139	380	132	-399	1589	268	609	-83	-42	3
1102	-82	1134	328	435	845	361	686	398	372	205	206	473	636	-92	-192	219	-296	408	185	137	492	-169	129	376	116	-474	1530	251	571	-131	-70	3
1121	-102	1173	331	460	806	328	697	406	379	155	168	465	623	-124	4	265	-342	372	166	98	450	-228	72	341	91	-553	1416	109	468	-226	-134	3
1121	-97	1193	341	451	779	328	697	410	381	104	155	452	624	-128	-52	252	-370	379	159	144	418	-295	61	344	108	-482	1416	97	465	-219	-119	3
1142	-99	1217	363	435	783	326	729	405	352	106	251	455	634	-136	-23	255	-325	400	169	135	438	-252	100	382	105	-481	1433	116	452	-195	-78	3
1195	-27	1242	383	462	790	364	747	415	349	102	275	484	637	-128	205	270	-312	390	148	-55	493	-223	124	384	30	-555	1474	127	467	-171	-86	3
1193	-9	1232	389	452	825	380	743	405	353	121	237	474	591	-59	143	236	-316	355	157	25	472	-284	87	347	68	-539	1470	122	453	-186	-129	3
1171	-14	1218	393	475	830	377	754	402	372	145	224	471	599	-7	116	265	-312	357	180	108	454	-217	83	359	105	-462	1418	111	451	-149	-112	3

Figure 4 Example Data From subj3.csv

CHAPTER 4
MACHINE LEARNING

Machine learning techniques became very popular over the last twenty years and they have been implemented over various research areas for widespread industrial use. The application of machine learning algorithms over applied science and across a variety of industries involved with user data problems; such as e-commerce services, faulty production systems, supplied chain managements and optimization systems. In addition to this, these approaches applied by many different scientist to a broad effect as machine learning techniques presented a novel way to predict the final results for complex problems (Jordan & Mitchell, 2015).

Machine learning techniques are classified as supervised, unsupervised, semi-supervised and reinforcement learning. Supervised means that the labels of the training data are familiar. Classification means that the output of the training system is separate, for instance "hit" or "stand". For unsupervised learning, machine learning is given with information samples with unknown label. A part of the coaching task is to cluster that training information belongs along, for instance that photos depict a similar person. Semi-supervised learning uses data both with labels and unknown labels to obtain a template. And, reinforcement learning makes its best to maximize the result of a complex goal that it is assigned without the training data (Heyden, 2016).

4.1 MACHINE LEARNING ALGORITHMS

As explained above, over the years machine learning algorithms are implemented for a variety of purposes and many different approaches are applied with different and successful results. For instance, these algorithms are used for; image and speech recognition, diagnosis, prediction, classification, learning associations, extraction and regression. Since machine learning uses statistical models and relies on patterns and inference, there will always a mathematical model to meet any need. They extend from the simple to the profoundly complex. Here are the algorithms that we used in this project.

4.1.1 MULTILAYER PERCEPTRON

The multilayer perceptron is made up of multiple layers of plain, two-state, sigmoid system parts or neurons, via using weighted connections. Inside multilayer perceptron there is an input layer, then there is a number of intermediate layers followed by an output layer. There is no interconnection within a layer but all neurons in an exceeding layer are totally connected to neurons in the adjacent layers. Weights present the degree of the correlations between the activity levels of neurons that they connect. An external input vector is inherited within the input layer imposed at the nodes. During training stage, the output node is imposed to 1 and while the others are imposed to 0 (Pal & Mitra, 1992).

4.1.2 DECISION TREE (J48)

A decision tree algorithm appoints a weight to every selection of a supported decision. In order to achieve this, the algorithm inherits a simple principle: P(f|h), where P is probability, f is the set of choices and h is the context of decision. The probability P(f|h) is decided by calculating an n number are questions i.e. q1, q2,. . . , qn , where n is decided by the answers to the n-1 past questions. Each question asked by the decision tree is presented by a tree node and the predicted answers to this question are presented with branches coming out from this specific node. Each node here shows a series of possible decisions with their respective probabilities. When the decision tree stops asking questions at a node then this means that this is a leaf node which represent a solution to the decision making (Magerman, 1995).

4.1.3 RANDOM FOREST

Random forest is an algorithm developed for classification problems by Breiman that applies an ensemble of classification trees. Each one of the classification trees applies a sample of information and at every split employs a random set of variables which in return enables a bootstrap aggregation approach. As a result random forest presents a novel method combining random variable choices with unstable learners.

In addition, every tree left unpruned to achieve low bias and low correlation which in return results to low variance (Díaz-Uriarte & De Andres, 2006).

4.1.4 IBK KNN (IB1)

k-nearest neighbor (kNN) algorithm basically randomly selects k number of instances within the training set that our closest to the test instance and make the decision according to the class of this neighborhood. Owing to its simplicity, kNN is straightforward to change for a lot difficult classification issues like multimodal categories. One of the foremost pronto offered kNN applications can be found in WEKA as IB1. The main approach of IB1 is to permit one or two of distances' weights specifically which in return enables the algorithm to utilize cross validation. In simple the algorithm selects the closest neighbor to make a decision (Steinbach & Tan, 2009).

4.1.5 CLASSIFICATION VIA REGRESSION

The classification via regression is a supervised learning method that remodels issues into regression functions. This method implements the basic principles a winner regression algorithm and a decision tree algorithm via using two steps:

1. First implement a decision tree over the set of instances by increasing the partition of attributes and their variations regarding the output while applying the deviation reduction.

2. Pruning the decision tree via applying regression algorithm over subtrees and subsequently their leaves (Yulita, Fanany & Arymurthy, 2019).

4.1.6 RANDOM TREE

Random Tree algorithm is an ensemble learning classification algorithm that inherits several learning algorithm. Random tree supplies a random set of information to construct a decision tree via bagging. In random forest every node is split via employing the best among the set of predicators arbitrarily at that specific node. This in return affects the regression hence a collection of tree predictors are implemented. The algorithm calculates the input feature vector then classifies it with each tree and finally makes a decision regarding to the class (Kalmegh, 2015).

4.1.7 NAIVE BAYES

Naive Bayes is the foremost economical, efficient and aggressive classification algorithm. What makes Naive Bayes spectacular is that its working principle consists of a conditional independence assumption. Naive Bayes is the plainest style of Bayesian structure, during that every attribute is freelance assigned the worth of the category parameter also known as conditional independence. One simple method to overcome the limits of the algorithm is to increase the format to act expressly dependencies between variables (Zhang, 2004).

4.1.8 BAYES NET

Bayesian Network is a combination of graphs, nodes, links and conditional probability tables. Nodes mean family, link merges nodes, tables ,which every node has one and generates probability distribution, shows the power of the links. Probability distribution changes depending on the condition that whether a node has parent/parents or not. When there is no parent it is unconditional, but in case of one or more parents it is a conditional distribution (Williams, Zander & Armitage, 2006).

4.1.9 WEKA DEEP LEARNING 4J

Deep learning has increased progression in numerous machine learning tasks via forming multi layered information. WekaDeeplearning4j, deep learning package of WEKA, is created by a java library to include renovating features of the algorithm to WEKA (Lang, Bravo-Marquez, Beckham, Hall & Frank, 2019).

4.1.10 LOGISTIC

Logistic regression is a straightforward extended part of binary logistic regression which allows more than two result value. The algorithm utilizes probability estimation for considering categorical membership resemblance. Logistic is employed for guessing possibility of a class involvement for a variable supported in many freelance attributes. The freelance variables are either divided or continuous (Starkweather & Moske, 2011).

4.1.11 SMO

Sequential Minimal Optimization (SMO) is an algorithm to solve quadratic programming (QP) issue that is caused by training of Support Vector Machines (SVM). It is a straightforward solution to rapidly resolve the SVM QP downside with none further matrix storage and while not employing numerical QP improvement steps by no means. As oppose to other solutions, SMO prefers unraveling the general problem into tiniest problems utilizing Osuna's theorem for each stage. Each tiniest problem includes two Lagrange multipliers, as a result of the Lagrange multipliers should adopt an equality limitation. For every stage, algorithm selects two Lagrange multipliers for collectively adjust, gets an optimum value, and renews the SVM for replicating the latest optimum rates (Platt, 1998).

4.1.12 BAGGING

Bagging, referred as bootstrap aggregating, is a meta-algorithm employed in machine-learning and information discovery. Bagging uses resampling to create a group of additional datasets based on an initial dataset. The additional datasets are referred to as bootstrap samples. The bootstrap samples may be used in parallel for data analytic tasks which increases speed of the analysis. The bootstrap samples may be used to train a group of classification or regression models. The results obtained by combining the results of individual models ran in parallel is often less time and resource consuming and often more accurate to the results of one model trained on the whole initial dataset (Dygas, Iwanowski, Plonski & Rokicki, 2015).

4.1.13 LOGIT BOOST

Boosting shows the foremost improvements in classification algorithms and performs via consecutively using a classifying rule for reweighted forms of the training data, so ends up getting almost every vote of the series of classifiers. And, Logit Boost represents a category to execute classification for employing additive logistic regression because of the multi class issues. Logit Boost and AdaBoost are close to one another within the sense that each perform an additive logistic regression. The distinction is that AdaBoost reduces the exponential damage, where Logit Boost reduces the logistic loss (Friedman, Hastie & Tibshirani, 2000).

4.1.14 DECISION TABLE

Decision Table is a correct technique that uses decision trees to get a numeric prediction and consists of IF-THEN logic which is a lot of efficient and comprehendible than the decision trees. Choice to explore decision tables as a result of its simplicity, less figure intensive algorithm than the decision-tree-based approach. It calculates attribute subclasses utilizing best-first search and may utilize cross-validation to analyze. A choice uses the nearest-neighbor technique to work out the category for every instance that is not coated by a decision table entry, rather than the table's global majority, supported a similar set of options (Kohavi, 1995).

4.1.15 HOEFFDING TREE

A HOEFFDING tree, kind of decision tree induction rule, can learn with large information if distribution samples do not amendment by the time of progress. HOEFFDING trees take advantage of the fact that a tiny section will typically be sufficient for settling on an optimum dividing element. The main asset is that the rule is attainable for ensuring underneath reasonable expectations which the trees it generates are asymptotically at random getting ready to those made by batch learner (Domingos & Hulten, 2000)

CHAPTER 5
IMPLEMENTATION AND EVALUATION

5.1 IMPLEMENTATION

After data preparation, there is now twelve csv files for twelve participants. Since, WEKA (Waikato Environment for Knowledge Analysis) is preferred as a studying environment, it is obligatory to create the corresponding ARFF (Attribute-Relation File Format) file for each csv content as the WEKA only accepts ARFF extension.

WEKA GUI Chooser has explorer, experimenter, knowledge flow, workbench and simple cli applications options. In this study, explorer application which has data preprocessing and classifying sections is used.

An ARFF file is a document which contains information about the data. It consists of two parts. First part, header, includes the name of the relation, attribute names and attribute types. Second part only has the data. Figure 5 and 6 shows an example header and data respectively.

```
@relation subj1
@attribute Fp1 numeric
@attribute Fp2 numeric
@attribute F7 numeric,
@attribute F3 numeric,
@attribute Fz numeric,
@attribute F4 numeric,
@attribute F8 numeric,
@attribute FC5 numeric,
@attribute FC1 numeric,
@attribute FC2 numeric,
@attribute FC6 numeric,
@attribute T7 numeric,
@attribute C3 numeric,
@attribute Cz numeric,
@attribute C4 numeric,
@attribute T8 numeric,
@attribute TP9 numeric,
@attribute CP5 numeric,
@attribute CP1 numeric,
@attribute CP2 numeric,
@attribute CP6 numeric,
@attribute TP10 numeric,
@attribute P7 numeric,
@attribute P3 numeric,
@attribute Pz numeric,
@attribute P4 numeric,
@attribute P8 numeric,
@attribute PO9 numeric,
@attribute O1 numeric,
@attribute Oz numeric,
@attribute O2 numeric,
@attribute PO10 numeric,
@attribute LRP {1,2,3},
```

Figure 5 An Example Header From subj1.arff File

```
@data
290,285,460,486,186,547,726,720,170,165,676,64,363,78,553,922,638,211,296,425,484,839,665,371,392,388,703,214,341,203,189,632,3
376,519,550,538,201,416,747,709,179,148,501,127,407,103,548,860,617,267,326,450,558,985,691,404,436,423,796,334,420,280,297,712,3
469,620,608,437,224,493,809,653,165,134,607,185,384,90,563,1076,690,230,309,449,459,945,746,411,431,412,758,383,479,310,310,736,3
466,693,602,407,197,496,890,592,149,147,636,161,335,80,572,1160,681,190,304,454,451,941,754,417,465,434,775,444,517,364,377,832,3
425,690,727,416,184,504,837,633,168,156,655,202,344,110,583,1047,701,226,323,471,500,907,750,457,478,477,795,396,487,361,385,824,3
392,639,676,406,189,421,808,673,174,141,648,283,377,113,572,1027,735,279,340,493,503,943,788,467,510,534,854,396,530,433,453,930,3
369,639,668,414,196,474,831,668,143,157,688,337,395,122,595,1015,683,291,351,506,541,1019,788,477,529,533,906,380,530,439,479,950,3
315,624,720,439,187,543,825,652,126,168,709,388,361,104,620,866,672,257,345,498,642,1060,768,461,515,519,928,361,509,380,438,931,3
339,566,729,440,201,560,788,750,139,162,670,354,383,101,614,1006,698,249,341,491,663,1029,741,449,481,508,905,303,468,341,376,882,3
403,573,703,468,227,624,894,754,151,178,710,270,395,111,581,1254,736,294,335,470,446,1034,787,463,485,474,847,374,523,392,401,879,3
404,607,733,486,216,616,872,784,142,176,738,261,374,100,585,1154,716,290,343,476,490,923,816,469,481,487,853,441,575,413,416,836,3
391,640,715,489,221,551,847,770,140,186,716,324,356,99,600,1105,741,296,342,492,543,901,848,476,492,499,874,474,594,414,402,824,3
```

Figure 6 An Example Data From subj1.arff File

In the header section, attributes are listed in a specific way. After stating it is an attribute by declaring @attribute, then comes the attribute type and only after that the value can be written. And the most important point is that the listing order of the attributes must be the same with the column order of the data. Means that if an attribute is the third one declared then its corresponding data must be in the third one from the left order.

The data type can be any of the four varieties approved by WEKA: numeric (integer and real types are also numeric), nominal, string and date. In this project, only numeric and nominal attributes are utilized. Numeric attributes contain integer numbers. And, nominal has labeled values. First 32 attributes that holds the EEG data are numeric and the last LRR value is nominal.

5.2 EVALUATION

Fifteen chosen algorithms were implemented on the WEKA with 10-fold cross validation. Cross validation is a commonplace analysis technique accustomed for checking the effectiveness of machine learning models, additionally a re-sampling procedure utilized to measure a model if we have a restricted information. Divide a dataset into ten folds, then hold out each bit successively for test and train on the remaining nine along. This provides ten analysis results, that are averaged. In "stratified" cross-validation, once doing the initial division, we have a tendency to make sure that every fold contains or so the right proportion of the category values. Having done 10-fold cross-validation and computed the analysis results, WEKA invokes the training formula a final (11th) time on the whole dataset to get the model that it prints out.

After completing the running-the-algorithms process, all the results are compared to decide whether this subject shows any promise or not. To settle on the best result, there are certain to check.

- True Positive (TP): the proportion of actual positives that are properly known.

		Predicted	
		Positive	**Negative**
Observed	**Positive**	TP (# of TPs)	FN (# of FNs)
	Negative	FP (# of FPs)	TN (# of TNs)

Figure 7 A Confusion Matrix of Binary Classification (Saito & Rehmsmeier, 2016)

- Precision: refers to the closeness of two or more measurements to each other.

$$\mathrm{PREC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Figure 8 Precision Formula (Saito & Rehmsmeier, 2016)

- Receiver Operating Characteristic (ROC): performance analysis of classification process at various threshold settings. Analysis of ROC curve is made by TPs and FPs concluding with a scale to decide the accuracy of them. The scale is like (Thomas & Tape, 2009):
  - .90-1 = excellent (A)
  - .80-.90 = good (B)
  - .70-.80 = fair (C)
  - .60-.70 = poor (D)
  - .50-.60 = fail (F)

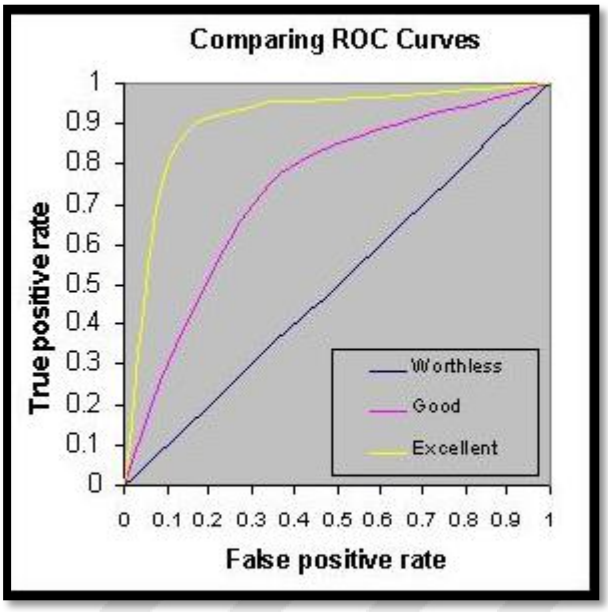


Figure 9 ROC Curve Graph (Thomas & Tape, 2009)

Considering TP, Precision and ROC results, it has seen that IB1 algorithm has the best result with over 96% TP and ROC rate. Random forest is the second best algorithm with the range of 93% to 99% TP and 99% to 1 ROC rate. There are also other very successful results as you can see in the figures above. Bagging and Classification via Regression algorithms both have TP rate around 80% but their ROC results are over 90%. Means that these algorithms are also achieved the purpose of this study. The results below are colored according to the success rates.

- IB1 and Random Forest -> Yellow
- Bagging and Classification Via Regression -> Green
- If there is any other algorithm with over 80% TP and Precision->Light Pink

For subject one, IB1 gives the best result with 98% TP, Precision and ROC. Then comes Random Forest with 93% TP and Precision and 99% ROC. Bagging has 84% TP, 83% Precision and 95% ROC. Classification via Regression has 82% TP and Precision and 93% ROC. Decision Tree, Multilayer Perceptron and Random Tree stays under 80%.

	TP	Precision	ROC
Bagging	0,841	0,839	0,956
Bayes Net	0,522	0,509	0,701
Classification via Regression	0,826	0,824	0,939
Decision Table	0,594	0,577	0,770
Decision Tree	0,778	0,778	0,844
Hoeffding Tree	0,608	0,588	0,747
IB1(IBK-KNN 1)	0,982	0,982	0,987
Logistic	0,644	0,632	0,813
Logitboost	0,598	0,575	0,780
Multilayer Perceptron	0,756	0,753	0,900
Naive Bayes	0,485	0,480	0,663
Random Forest	0,932	0,932	0,990
Random Tree	0,762	0,762	0,822
SMO	0,655	0,642	0,766
Weka Deep Learning 4J	0,636	0,623	0,806

Figure 10 Results of Subj1

For subject two, IB1 gives the best result with 96% TP and Precision and 97% ROC. Then comes Random Forest with 95% TP and Precision and 99% ROC. Bagging shows 90% TP and Precision and 98% ROC. Classification via Regression is resulted with 88% TP and Precision but 96% ROC. Also, Decision Tree has an outcome of 85% TP and Precision and 90% ROC. Multilayer Perceptron provides 83% TP, 82% Precision and 93% ROC. Finally, Random Tree has 82% TP and Precision and 87% ROC.

	TP	Precision	ROC
Bagging	0,902	0,901	0,981
Bayes Net	0,522	0,556	0,732
Classification via Regression	0,886	0,885	0,969
Decision Table	0,653	0,648	0,839
Decision Tree	0,853	0,853	0,903
Hoeffding Tree	0,638	0,618	0,785
IB1(IBK-KNN 1)	0,962	0,962	0,974
Logistic	0,669	0,657	0,835
Logitboost	0,636	0,622	0,819
Multilayer Perceptron	0,830	0,828	0,935
Naive Bayes	0,530	0,528	0,711
Random Forest	0,958	0,958	0,995
Random Tree	0,822	0,822	0,870
SMO	0,679	0,668	0,782
Weka Deep Learning 4J	0,662	0,650	0,831

Figure 11 Results of Subj2

For subject three, IB1 gives the best result with 99% TP, Precision and ROC. Then comes Random Forest with 98% TP and Precision and 1 ROC. Bagging has 94% TP and Precision and 99% ROC. Classification via Regression shows 92% TP, 91% Precision and 98% ROC. Decision Tree provides 89% TP and Precision and 92% ROC. Random Tree has an outcome of 88% TP and Precision and 91% ROC. Finally, Multilayer Perceptron has 80% TP and Precision and 93% ROC.

	TP	Precision	ROC
Bagging	0,949	0,949	0,994
Bayes Net	0,524	0,529	0,689
Classification via Regression	0,920	0,919	0,981
Decision Table	0,682	0,694	0,861
Decision Tree	0,891	0,891	0,928
Hoeffding Tree	0,564	0,538	0,682
IB1(IBK-KNN 1)	0,996	0,996	0,997
Logistic	0,562	0,538	0,732
Logitboost	0,560	0,544	0,712
Multilayer Perceptron	0,802	0,800	0,931
Naive Bayes	0,454	0,452	0,601
Random Forest	0,988	0,988	1,000
Random Tree	0,880	0,880	0,911
SMO	0,573	0,555	0,655
Weka Deep Learning 4J	0,560	0,536	0,725

Figure 12 Results of Subj3

For subject four, IB1 gives the best result with 99% TP, Precision and ROC. Random Forest follows with 97% TP and Precision and 99% ROC. Also, Bagging has 91% TP and Precision and 98% ROC. Classification via Regression shows 89% TP and Precision and 96% ROC. Decision Tree has an outcome of 86% TP and Precision and 90% ROC. Random Tree is resulted with 82% TP and Precision and 87% ROC. Multilayer Perceptron stays under 80%.

	TP	Precision	ROC
Bagging	0,917	0,917	0,986
Bayes Net	0,566	0,564	0,752
Classification via Regression	0,891	0,890	0,969
Decision Table	0,663	0,662	0,843
Decision Tree	0,860	0,860	0,907
Hoeffding Tree	0,595	0,585	0,746
IB1(IBK-KNN 1)	0,995	0,995	0,997
Logistic	0,596	0,586	0,774
Logitboost	0,570	0,562	0,742
Multilayer Perceptron	0,797	0,794	0,924
Naive Bayes	0,501	0,491	0,664
Random Forest	0,976	0,976	0,998
Random Tree	0,829	0,829	0,872
SMO	0,601	0,592	0,715
Weka Deep Learning 4J	0,591	0,582	0,770

Figure 13 Results of Subj4

For subject five, IB1 gives the best result with 99% TP, Precision and ROC. Random Forest follows with 99% TP and Precision and 1 ROC. Bagging has 96% TP and Precision and 99% ROC. Classification via Regression shows 92% TP and Precision and 98% ROC. Decision Tree also has an outcome of 90% TP and Precision and 93% ROC. And, Random Tree provides 88% TP and Precision and 91% ROC. Multilayer Perceptron stays under 80%.

	TP	Precision	ROC
Bagging	0,961	0,961	0,996
Bayes Net	0,516	0,513	0,691
Classification via Regression	0,925	0,925	0,980
Decision Table	0,736	0,743	0,905
Decision Tree	0,902	0,902	0,937
Hoeffding Tree	0,535	0,520	0,688
IB1(IBK-KNN 1)	0,997	0,997	0,998
Logistic	0,539	0,520	0,717
Logitboost	0,521	0,503	0,692
Multilayer Perceptron	0,743	0,739	0,900
Naive Bayes	0,464	0,459	0,623
Random Forest	0,994	0,994	1,000
Random Tree	0,885	0,885	0,914
SMO	0,540	0,519	0,656
Weka Deep Learning 4J	0,534	0,514	0,712

Figure 14 Results of Subj5

For subject six, IB1 gives the best result with 99% TP, Precision and ROC. Random Forest follows with 97% TP and Precision and 99%, almost 1, ROC. Bagging has 93% TP and Precision and 99% ROC. Classification via Regression shows 90% TP and Precision and 97% ROC. Decision Tree provides 87% TP and Precision and 92% ROC. At last, Random Tree has an outcome of 86% TP and Precision and 90% ROC. Multilayer Perceptron stays under 80%.

	TP	Precision	ROC
Bagging	0,931	0,930	0,990
Bayes Net	0,566	0,556	0,752
Classification via Regression	0,906	0,906	0,977
Decision Table	0,678	0,688	0,864
Decision Tree	0,879	0,879	0,920
Hoeffding Tree	0,619	0,599	0,759
IB1(IBK-KNN 1)	0,994	0,994	0,995
Logistic	0,626	0,606	0,794
Logitboost	0,617	0,595	0,803
Multilayer Perceptron	0,785	0,780	0,896
Naive Bayes	0,524	0,512	0,701
Random Forest	0,979	0,979	0,999
Random Tree	0,864	0,864	0,901
SMO	0,654	0,633	0,761
Weka Deep Learning 4J	0,617	0,598	0,789

Figure 15 Results of Subj6

For subject seven, IB1 gives the best result with over 99%, almost 1, TP, Precision and ROC. Random Forest follows with 99% TP and Precision and 1 ROC. Bagging has 94% TP and Precision and 99% ROC. Classification via Regression shows 93% TP and Precision and 98% ROC. Decision Tree has an outcome of 90% TP and Precision and 93% ROC. And, Random Tree is resulted with 89% TP and Precision and 92% ROC. Finally, Multilayer Perceptron provides 84% TP and Precision and 94% ROC.

	TP	Precision	ROC
Bagging	0,949	0,949	0,994
Bayes Net	0,622	0,605	0,797
Classification via Regression	0,932	0,931	0,984
Decision Table	0,710	0,711	0,880
Decision Tree	0,908	0,907	0,939
Hoeffding Tree	0,665	0,651	0,798
IB1(IBK-KNN 1)	0,999	0,999	0,999
Logistic	0,611	0,602	0,790
Logitboost	0,641	0,627	0,812
Multilayer Perceptron	0,845	0,844	0,942
Naive Bayes	0,544	0,529	0,716
Random Forest	0,990	0,990	1,000
Random Tree	0,894	0,894	0,921
SMO	0,651	0,643	0,762
Weka Deep Learning 4J	0,609	0,600	0,787

Figure 16 Results of Subj7

For subject eight, IB1 gives the best result with over 99%, almost 1, TP, Precision and ROC. Random Forest follows with 99% TP and Precision and 1 ROC. Bagging has 95% TP and Precision and 99% ROC. Classification via Regression shows 93% TP and Precision and 98% ROC. Decision Tree has an outcome of 90% TP and Precision and 94% ROC. And, Random Tree provides 89% TP and Precision and 92% ROC. Finally, Multilayer Perceptron is resulted with 83% TP and Precision and 94% ROC.

	TP	Precision	ROC
Bagging	0,955	0,955	0,995
Bayes Net	0,554	0,560	0,748
Classification via Regression	0,937	0,936	0,986
Decision Table	0,723	0,728	0,894
Decision Tree	0,907	0,907	0,940
Hoeffding Tree	0,636	0,623	0,782
IB1(IBK-KNN 1)	0,998	0,998	0,999
Logistic	0,643	0,630	0,834
Logitboost	0,621	0,605	0,811
Multilayer Perceptron	0,838	0,841	0,943
Naive Bayes	0,531	0,540	0,711
Random Forest	0,993	0,993	1,000
Random Tree	0,894	0,894	0,922
SMO	0,661	0,653	0,775
Weka Deep Learning 4J	0,639	0,626	0,830

Figure 17 Results of Subj8

For subject nine, IB1 gives the best result with over 99%, almost 1, TP, Precision and ROC. Random Forest follows with 99% TP and Precision and 1 ROC. Bagging has 96% TP and Precision and 99% ROC. Classification via Regression shows 93% TP and Precision and 98% ROC. Decision Tree has an outcome of 92% TP and Precision and 94% ROC. And, Random Tree provides 90% TP and Precision and 92% ROC. Multilayer Perceptron stays under 80%.

	TP	Precision	ROC
Bagging	0,965	0,965	0,996
Bayes Net	0,538	0,549	0,731
Classification via Regression	0,934	0,934	0,984
Decision Table	0,688	0,690	0,867
Decision Tree	0,922	0,922	0,949
Hoeffding Tree	0,620	0,611	0,780
IB1(IBK-KNN 1)	0,997	0,997	0,998
Logistic	0,599	0,591	0,780
Logitboost	0,585	0,579	0,770
Multilayer Perceptron	0,797	0,794	0,921
Naive Bayes	0,502	0,497	0,676
Random Forest	0,992	0,992	1,000
Random Tree	0,903	0,903	0,928
SMO	0,606	0,599	0,730
Weka Deep Learning 4J	0,589	0,581	0,774

Figure 18 Results of Subj9

For subject 10, IB1 gives the best result with over 99%, almost 1, TP, Precision and ROC. Random Forest follows with 99% TP and Precision and 1 ROC. Bagging has 94% TP and Precision and 99% ROC. Classification via Regression shows 90% TP and Precision and 97% ROC. Decision Tree has an outcome of 86% TP and Precision and 91% ROC. And, Random Tree is resulted with 85% TP and Precision and 89% ROC. Multilayer Perceptron stays under 80%.

	TP	Precision	ROC
Bagging	0,940	0,940	0,991
Bayes Net	0,524	0,542	0,711
Classification via Regression	0,909	0,909	0,975
Decision Table	0,624	0,633	0,818
Decision Tree	0,869	0,869	0,912
Hoeffding Tree	0,566	0,556	0,731
IB1(IBK-KNN 1)	0,999	0,999	0,999
Logistic	0,575	0,562	0,769
Logitboost	0,539	0,526	0,726
Multilayer Perceptron	0,765	0,763	0,910
Naive Bayes	0,496	0,505	0,673
Random Forest	0,991	0,991	1,000
Random Tree	0,857	0,857	0,894
SMO	0,579	0,565	0,709
Weka Deep Learning 4J	0,568	0,557	0,762

Figure 19 Results of Subj10

For subject 11, IB1 gives the best result with 99% TP, Precision and ROC. Random Forest follows with 98% TP and Precision and 99% ROC. Bagging has 94% TP and Precision and 99% ROC. Classification via Regression shows 92% TP and Precision and 98% ROC. Decision Tree has an outcome of 90% TP and Precision and 93% ROC. And, Random Tree provides 88% TP and Precision and 91% ROC. Multilayer Perceptron stays under 80%.

	TP	Precision	ROC
Bagging	0,949	0,949	0,993
Bayes Net	0,624	0,622	0,813
Classification via Regression	0,927	0,927	0,982
Decision Table	0,648	0,648	0,838
Decision Tree	0,903	0,903	0,935
Hoeffding Tree	0,655	0,649	0,830
IB1(IBK-KNN 1)	0,990	0,990	0,993
Logistic	0,600	0,590	0,792
Logitboost	0,601	0,592	0,797
Multilayer Perceptron	0,768	0,770	0,913
Naive Bayes	0,570	0,564	0,759
Random Forest	0,985	0,985	0,999
Random Tree	0,882	0,882	0,913
SMO	0,612	0,601	0,741
Weka Deep Learning 4J	0,591	0,581	0,786

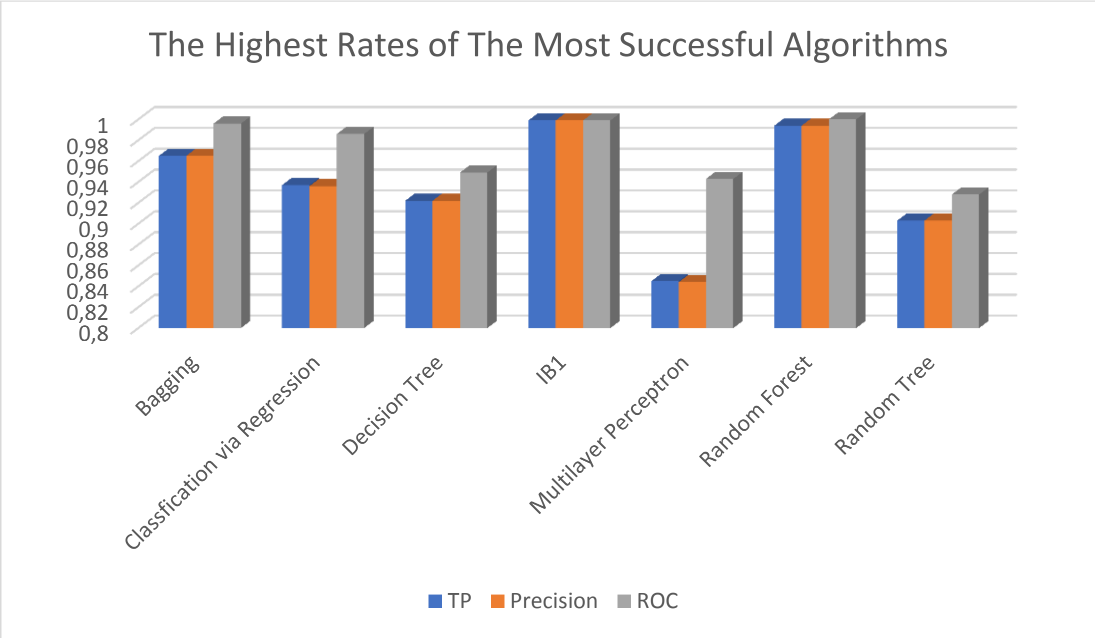
Figure 20 Results of Subj11

For subject 12, IB1 gives the best result with 96% TP and Precision and 97% ROC. Random Forest follows it with 93% TP and Precision and 99% ROC. Bagging has an outcome of 85% TP and Precision and 96% ROC. Classification via Regression shows 82% TP and Precision and 93% ROC. Decision Tree is resulted with 80% TP and Precision and 86% ROC. Multilayer Perceptron could not pass 80% TP and Precision but its ROC value is almost 90%. Random Tree also stays under 80% for TP and Precision it has 81% ROC.

	TP	Precision	ROC
Bagging	0,858	0,856	0,964
Bayes Net	0,565	0,559	0,746
Classification via Regression	0,824	0,821	0,937
Decision Table	0,593	0,594	0,778
Decision Tree	0,804	0,804	0,865
Hoeffding Tree	0,568	0,543	0,715
IB1(IBK-KNN 1)	0,969	0,969	0,979
Logistic	0,588	0,567	0,767
Logitboost	0,561	0,536	0,749
Multilayer Perceptron	0,731	0,728	0,896
Naive Bayes	0,520	0,508	0,702
Random Forest	0,935	0,935	0,991
Random Tree	0,750	0,751	0,813
SMO	0,592	0,570	0,697
Weka Deep Learning 4J	0,580	0,559	0,761

Figure 21 Results of Subj12

To sum up the above results in the figures, it can be said that there are seven successful algorithms which are Bagging, Classification via Regression, Decision Tree, IB1, Multilayer Perceptron, Random Forest and Random Tree. IB1 is the most successful algorithm in case of TP, Precision and ROC values. Second successful algorithm is Random Forest and the algorithms follow. All of their TP, Precision and ROC values range from 0.845 to 1. Although the results are between zero and one, in the graph the lowest value is described as 0.80 to help the readers see the results more clearly.



The Highest Rates of The Most Successful Algorithms
1
0,98
0,96
0,94
0,92
0,9
0,88
0,86
0,84
0,82
0,8
Bagging
Classfication via Regression
Decision Tree
IB1
Multilayer Perceptron
Random Forest
Random Tree
TP
Precision
ROC

CHAPTER 6
CONCLUSION AND FUTURE WORK

The study successfully proved that a robotic gadget controlled by brain signals is suitable to be used by people. Having a large and diverse dataset answers some questions such as when to aim or release an object, position of the hand and changes of the features of the item. With asking these right questions and selecting the right machine learning algorithms, obtaining hand movement signals form EEG graph looks affordable. Also, it is learned what proportion of the data will truly be extracted from signals. Especially, the limitation of finding the right signals to watch or manage tasks like controlling upper limb for a grasp includes thumb and forefinger.

IB1 is proved to be the most successful algorithm in fifteen algorithms. It has 99% TP, Precision and ROC value. Actually, the value is 0,999 almost 1.The study shows that IB1 is not the only accomplished algorithm. Random Forest has great results which compete with IB1.It has an outcome of 94% TP and Precision value. And, its highest ROC value is 1.Even though there are two successful algorithms with astonishing results, it has seen that there are other promising results. Bagging and Classification via Regression algorithms have more than 90% TP, Precision and ROC values. Such that the highest result of Bagging algorithm is 96% TP and Precision and 99% ROC. In the same way, Classification via Regression algorithm has a result of 93% TP and Precision and 98% ROC value as the highest value. Then comes Decision Tree algorithm with again over 90% outcome. It gives 92% TP and Precision and almost 95% ROC. Yet another algorithm is Random Tree. This algorithm holds the value of 90% TP and Precision and 92% ROC. These numbers can be categorized as promising. And the last one is Multilayer Perceptron. Since it is decided to draw the lowest line at 80%, this algorithm is included among the successful ones. Multilayer Perceptron provides 84% TP and Precision and 94% ROC. With some arrangements these results may be increased so it can be said that the algorithm shows promise.

And these outcomes demonstrate that EEG is dependable when it comes to managing robotic tools. Trying EEG with grasp and lift concept reveals deep information about EEG and shows promise on the area of robotic device management. Higher knowledge about EEG signals and hand movements is crucial for developing a BCI gadget which might offer people the power to maneuver in life.

In later times, a couple of changes could be done in this study. For starters, number of the participants can be increased. While rising the number, they should be selected from different countries, weights, ages and professions. In addition, the object would be lifted from other contact surfaces. There are already three different surfaces, but it may not be enough. Speaking of the object, it is a good target when aiming for extending. Very different objects must be provided in terms of all kinds of specifics. Like; shape, weight, height, size, usage, etc. All these supplementations will affect the validity of the database.

The development of sturdy analytical strategies advantages BCI users providing quicker, correct classification and long clinical use (Podmore, Breckon, Aznan & Connolly, 2019). BCI oriented EEG studies became extraordinarily widespread lately since its main benefit is to help people in need controlling a robotic device via their brain signals (Paranjape, Dhabu, Deshpande & Kekre, 2019). This idea of robotic device really aids people with hand disabilities to get back to their daily life.

REFERENCES

Ramzan, M., & Dawn, S. (2019, January). A Survey of Brainwaves using Electroencephalography (EEG) to develop Robust Brain-Computer Interfaces (BCIs): Processing Techniques and Algorithms. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 642-647). IEEE.

Islam, M. K., & Rastegarnia, A. (2019, February). Probability Mapping Based Artifact Detection and Wavelet Denoising based Artifact Removal from Scalp EEG for BCI Applications. In 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS) (pp. 243-247). IEEE.

Chenane, K., & Touati, Y. (2018, December). EEG Signal Classification for BCI based on Neural Network. In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 2573-2576). IEEE.

Alomari, M. H., Samaha, A., & AlKamha, K. (2013). Automated classification of L/R hand movement EEG signals using advanced feature extraction and machine learning. arXiv preprint arXiv:1312.2877.

Shimizu, T., Dobashi, H., Ito, K., & Nagai, K. (2014, December). Consideration on a new BCI method using EEG to detect motion intention. In 2014 IEEE/SICE International Symposium on System Integration (pp. 785-789). IEEE.

Robinson, N., & Vinod, A. P. (2015, October). Bi-directional imagined hand movement classification using low cost EEG-based BCI. In 2015 IEEE International Conference on Systems, Man, and Cybernetics (pp. 3134-3139). IEEE.

Norman, S. L., Dennison, M., Wolbrecht, E., Cramer, S. C., Srinivasan, R., & Reinkensmeyer, D. J. (2016). Movement anticipation and EEG: implications for BCI-contingent robot therapy. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 24(8), 911-919.

Zhang, W., Sun, F., Liu, C., Su, W., Tan, C., & Liu, S. (2017, October). A hybrid EEG-based BCI for robot grasp controlling. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3278-3283). IEEE.

Yavuz, E., & Aydemir, Ö. (2017, July). Classification of EEG based BCI signals imagined hand closing and opening. In 2017 40th International Conference on Telecommunications and Signal Processing (TSP) (pp. 425-428). IEEE.

Alazrai, R., Aburub, S., Fallouh, F., & Daoud, M. I. (2017, November). EEG-based BCI system for classifying motor imagery tasks of the same hand using empirical mode decomposition. In 2017 10th International Conference on Electrical and Electronics Engineering (ELECO) (pp. 615-619). IEEE.

Cho, J. H., Jeong, J. H., Shim, K. H., Kim, D. J., & Lee, S. W. (2018, October). Classification of Hand Motions within EEG Signals for Non-Invasive BCI-Based Robot Hand Control. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 515-518). IEEE.

Butt, M., Naghdy, G., Naghdy, F., Murray, G., & Du, H. (2018, October). Investigating Electrode Sites for Intention Detection During Robot Based Hand Movement Using EEG-BCI System. In 2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE) (pp. 177-180). IEEE.

Alazrai, R., Alwanni, H., & Daoud, M. I. (2019). EEG-based BCI system for decoding finger movements within the same hand. Neuroscience letters, 698, 113-120.

Luciw, M. D., Jarocka, E., & Edin, B. B. (2014). Multi-channel EEG recordings during 3,936 grasp and lift trials with varying weight and friction. Scientific data, 1, 140047.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

Heyden, M. (2016). Classification of EEG data using machine learning techniques.

Pal, S. K., & Mitra, S. (1992). Multilayer perceptron, fuzzy sets, and classification. IEEE Transactions on neural networks, 3(5), 683-697.

Magerman, D. M. (1995, June). Statistical decision-tree models for parsing. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics (pp. 276-283). Association for Computational Linguistics.

Díaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. BMC bioinformatics, 7(1), 3.

Steinbach, M., & Tan, P. N. (2009). kNN: k-nearest neighbors. The top ten algorithms in data mining, 151-162.

Yulita, I. N., Fanany, M. I., & Arymurthy, A. M. (2019, July). Comparing Classification via Regression and Random Committee for Automatic Sleep Stage Classification in Autism Patients. In Journal of Physics: Conference Series (Vol. 1230, No. 1, p. 012010). IOP Publishing.

Kalmegh, S. (2015). Analysis of weka data mining algorithm reptree, simple cart and randomtree for classification of indian news. International Journal of Innovative Science, Engineering & Technology, 2(2), 438-446.

Zhang, H. (2004). The optimality of naive Bayes. AA, 1(2), 3.

Williams, N., Zander, S., & Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Computer Communication Review, 36(5), 5-16.

Lang, S., Bravo-Marquez, F., Beckham, C., Hall, M., & Frank, E. (2019). Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. Knowledge-Based Systems, 178, 48-50.

Starkweather, J., & Moske, A. K. (2011). Multinomial logistic regression. Consulted page at September 10th: http://www. unt. edu/rss/class/Jon/Benchmarks/MLR_JDS_Aug2011. pdf, 29, 2825-2830.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

Dygas, S. A., Iwanowski, M. T., Plonski, P., & Rokicki, M. (2015). U.S. Patent Application No. 14/524,531.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics, 28(2), 337-407

Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).

Domingos, P., & Hulten, G. (2000, August). Mining high-speed data streams. In Kdd (Vol. 2, p. 4).

Saito, T., & Rehmsmeier, M. (2016). Basic evaluation measures from the confusion matrix. Retrieved from https://classeval.wordpress.com/introduction/basic-evaluation-measures/

Thomas, G. T., & Tape, M. D. (2009). Interpreting diagnostic tests. Retrieved from http://gim.unmc.edu/dxtests/ROC3.htm

Podmore, J. J., Breckon, T. P., Aznan, N. K., & Connolly, J. D. (2019). On the Relative Contribution of Deep Convolutional Neural Networks for SSVEP-Based Bio-Signal Decoding in BCI Speller Applications. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(4), 611-618.

Paranjape, P. N., Dhabu, M. M., Deshpande, P. S., & Kekre, A. M. (2019). Cross-Correlation Aided Ensemble of Classifiers for BCI Oriented EEG Study. IEEE Access, 7, 11985-1199