G. TONBUL

IMPLEMENTATION OF MACHINE LEARNING METHODS TO UNDERSTAND SURGICAL RESIDENTS' SKILL LEVELS THROUGH THEIR HAND MOVEMENTS GENERATED BY COMPUTER-BASED SIMULATION TRAINING ENVIRONMENTS

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

GÖKÇEN TONBUL

DOCTOR OF PHILOSOPHY THESIS
IN
THE DEPARTMENT OF SOFTWARE ENGINEERING

JUNE 2023

ATILIM UNIVERSITY 2023

IMPLEMENTATION OF MACHINE LEARNING METHODS TO UNDERSTAND SURGICAL RESIDENTS' SKILL LEVELS THROUGH THEIR HAND MOVEMENTS GENERATED BY COMPUTER-BASED SIMULATION TRAINING ENVIRONMENTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

GÖKÇEN TONBUL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF SOFTWARE ENGINEERING

JUNE 2023

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

Prof. Dr. Ender Keskinkılıç
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Doctor of Philosophy in Software Engineering, Atılım University**.

Prof. Dr. Ali Yazıcı
Head of Department

This is to certify that we have read the thesis IMPLEMENTATION OF MACHINE LEARNING METHODS TO UNDERSTAND SURGICAL RESIDENTS' SKILL LEVELS THROUGH THEIR HAND MOVEMENTS GENERATED BY COMPUTER-BASED SIMULATION TRAINING ENVIRONMENTS submitted by GÖKÇEN TONBUL and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Asst. Prof Dr. Damla Topallı
Co-Supervisor

Prof Dr. Nergiz Ercil Çağıltay
Supervisor

Examining Committee Members:

Prof. Dr. Yunus Gökmen
Public Rel. and Publ. Dep., Başkent University

Prof. Dr. Nergiz Ercil Çağıltay
Software Eng. Department, Atılım University

Asst. Prof. Dr. Erol Özçelik
Psychology Department, Çankaya University

Asst. Prof. Dr. Beytullah Yıldız
Software Eng. Department, Atılım University

Asst. Prof. Dr. Arda Sezen
Computer Eng. Department, Atılım University

Date: *June 23, 2023*

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Gökçen Tonbul

Signature :

ABSTRACT

IMPLEMENTATION OF MACHINE LEARNING METHODS TO UNDERSTAND SURGICAL RESIDENTS' SKILL LEVELS THROUGH THEIR HAND MOVEMENTS GENERATED BY COMPUTER-BASED SIMULATION TRAINING ENVIRONMENTS

Tonbul, Gökçen

Ph.D., Software Engineering Department

Supervisor: Prof. Dr. Nergiz Ercil Çağıltay

Co-Supervisor: Asst. Prof. Dr. Damla Topallı

June 2023, 129 pages

Medical disciplines have been experiencing big challenges in its existing complex nature, parallel with the development of the new technologies. Classical approaches evolve into modern solutions in the adaptation process even some are becoming completely obsolete. The natural complications of an ordinary open surgery directed this evolution towards the term minimally invasive operations. Minimally invasive surgery (MIS), as a general term, uses or creates cavity in the body to reach the desired body part by using necessary tools. The aim is to give less pain to the patient by keeping less incision and tissue damage. However, there are still several problems for the education programs of related surgical procedures. For instance, defining and objectively measuring the surgical skill levels is a challenging process. In this regard, first a systematic review study is conducted to better understand the surgical skill level classification approaches. Afterwards, it is aimed to classify intermediate and novice surgical skills with higher accuracy compared to the previous classification efforts using any possible hand movement-oriented data gathered through virtual reality environments in an experimental study. The results show that it is possible to improve the classification more using different data engineering techniques based on a

reproducible adapted framework. It is believed that, in the future, it is possible to adapt this research study effort to any virtual environment with a proper set of tools, the applicable software engineering efforts on top of data science discernment, as well as possible innovative machine learning approximations.

Keywords: Virtual Reality, Surgical Education, Hand Movement, Feature Engineering, Machine Learning

ÖZ

CERRAHİ ASİSTANLARIN BECERİ DÜZEYLERİNİN ANLAŞILMASI AMACIYLA BİLGİSAYAR TABANLI SİMÜLASYON EĞİTİM ORTAMLARININ OLUŞTURDUĞU EL HAREKETLERİ VERİSİNE MAKİNE ÖĞRENME YÖNTEMLERİNİN UYGULANMASI

Tonbul, Gökçen

Doktora, Yazılım Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Nergiz Ercil Çağıltay

Tez Danışmanı: Dr. Damla Topallı

Haziran 2023, 129 sayfa

Tıp disiplinleri, yeni teknolojilerin gelişimine paralel olarak kendi mevcut karmaşık yapısı içinde büyük zorluklar yaşamaktadır. Klasik yaklaşımlar, adaptasyon sürecine girerek modern çözümlere evrilmekte hatta bazıları tamamen geçerliliğini yitirmektedir. Sıradan bir açık ameliyatın doğal olarak ortaya çıkan komplikasyonları, minimal invaziv ameliyatların gelişmesine yol açmıştır. Minimal invaziv cerrahi ile istenen vücut bölgesine ulaşmak için gerekli aletler yardımıyla, küçük kesiler açılarak vücuttaki boşluklar kullanılır veya yenisi oluşturulur. Böylece daha az kesi ve doku hasarı sayesinde hastanın da daha hızlı ve rahat bir iyileşme süreci geçirmesi amaçlanır. Bununla birlikte, ilgili cerrahi işlemlerin eğitim programlarında hala çeşitli sorunlar bulunmaktadır. Örneğin, cerrahi beceri düzeylerinin tanımlanması ve objektif olarak ölçülmesi zorlu bir süreçtir. Bu bağlamda, öncelikle cerrahi beceri düzeyi sınıflandırma yaklaşımlarını daha iyi anlamak için sistematik bir derleme çalışması yapılmıştır. Daha sonra deneysel bir çalışmada sanal gerçeklik ortamları aracılığıyla elde edilen el hareket verileri kullanılarak orta ve acemi cerrahi becerilerin önceki sınıflandırma çabalarına göre daha yüksek doğrulukla sınıflandırılması amaçlanmaktadır. Sonuçlar, yeniden üretilebilir uyarlanmış bir çerçeveye dayalı

olarak farklı veri mühendisliği teknikleri kullanılarak sınıflandırmanın daha iyi hale getirilmesinin mümkün olduğunu göstermektedir. Gelecekte bu araştırma çalışmasını, uygun bir araç seti, veri bilimi muhakemesinin üzerine inşa edilen yazılım mühendisliği çabaları ve muhtemel yenilikçi makine öğrenimi yaklaşımları ile herhangi bir sanal ortama uyarlamanın mümkün olduğuna inanılmaktadır.

Anahtar Kelimeler— Sanal Gerçeklik, Cerrahi Eğitim, El Hareketi, Öznitelik Mühendisliği, Makine Öğrenimi

To My Mom

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my supervisor; Prof Dr. Nergiz Ercil Çağıltay for her continuous support and strong motivation throughout my thesis study. She inspired me with her unwavering academic and ethical understanding. Her guidance to explore the unknown and enthusiasm to enrich scientific value are precious lifelong experiences. I offer sincere thanks for her extensive knowledge and endless patience during this period.

Additionally, I would like to express my sincere appreciation to my thesis co-supervisor; Asst. Prof. Dr. Damla Topallı. Her unwavering work discipline has been a constant source of motivation, while her extensive knowledge has brought scientific depth to this study. Moreover, she has consistently offered me genuine assistance whenever required. It is through her invaluable support that this work has evolved into a profound scientific endeavor, and for that, I am genuinely thankful.

Furthermore, I express my gratitude to all those who have contributed to the successful completion of Educational Computer-based-simulation Environment project. Their diligent efforts have provided us with the opportunity to conduct scientific exercises on a unique environment. I am truly grateful for their invaluable contribution in facilitating access to these precious resources.

I would like to extend my sincere gratitude to my defense committee, whose feedback with abundant knowledge and expertise have been invaluable throughout this journey. Additionally, I would like to express my heartfelt appreciation to head of Software Engineering Department, Prof. Dr. Ali Yazıcı, whose initial encouragement paved the way for this incredible experience for me.

Finally, I would like to express my love and appreciation to my wife, Özge, whose unwavering support and understanding allowed me to fully dedicate myself to this study. Furthermore, I am grateful to my sisters and father for their love and continuous

support throughout this research endeavor. Lastly, I express heartfelt thanks to my mother, whose nurturing guidance and unconditional love bestowed upon me the foundation of my early learning abilities and my scientific curiosity.

.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANNs	Artificial Neural Networks
AR	Augmented Reality
DL	Deep Learning
ECE	Educational Computer-based-simulation Environment
EL	Ensemble Learning
ICA	Independent Component Analysis
ILP	Inductive Logic Programming
LCS	Learning Classifier Systems
LDA	Linear Discriminant Analysis
LLE	Locally Linear Embedding
LSA	Latent Semantic Analysis
MDR	Multifactor Dimensionality Reduction
MIS	Minimally Invasive Surgery
ML	Machine Learning
MSL	Multi-linear Sub-space Learning
NN	Neural Network
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
RFC	Random Forest Classifier
SDE	Semi-definite Embedding
t-SNE	t-distributed Stochastic Neighbor Embedding
VR	Virtual Reality

CHAPTER 1

INTRODUCTION

Nowadays, an efficient application may hold a meaning of a system that uses insights from the data in a better way. Any scientific discipline, or business organization is aware of the importance of such insights and tries to handle data in a proper way. This is a clear revolution that we all have gone through which objectively relies on the tools and methodologies recently developed. It is possible to say that this situation has many similarities with the long and painful historical journey of the humanity. Where this cognitive journey takes us all, is the understanding of scientific heritage, that will somehow reduce the effects of this painful journey, overcome the obstacles that seems like an impossible puzzle to solve, and sometimes turns into a big question to get lost in it. Data has always been the most important asset of this continues progress. We were able to produce an alternate path with the tools necessary to adapt whenever the limits of our cognitive capacity were challenged. This perpetual revolution, guided by our imagination, did not changed when we encounter machines. “How we can build machines perform tasks like we do?” may had been one of the earliest questions that led us create todays’ subfields of computer science like Machine Learning (ML), Artificial Intelligence (AI) or Deep Learning (DL). Big data concepts were begun to appear with the power of same roots when human start to produce larger and larger data that is even hard to store properly. The cognitive capacity has been challenged and the response was to create many disciplines, tools and methodologies like todays’ data analytic tools, scientific computation approximations and visualization techniques.

This instinctive nature may direct us to handle a task that pushes our cognitive limits by new methodologies or to automate it if it is an exhaustive repetitive task, through machines and algorithms, and effectively solve this problem with latest approximations. Systematic computational approximations stand somewhere between this cognitive limitation and automation. For example, a software development process

may allow a developer to write a software about image recognition. However, it is not possible to reach a level of reliability to recognize objects between millions of images with different backgrounds, light conditions, or colors with traditional software development habits. To handle such pattern changes or complex decision-making processes, the effort needed is extreme and not applicable without proper disciplines. ML may be defined as a discipline of automation in that manner to develop a systematic pattern recognition with labeled or unlabeled datasets using computers. The machine uses many images to train itself and a predictive automated decision-making process discovers the patterns without explicit intervention.

Computer simulation as a scientific literature is not that different. It is also turning into a data-oriented approximation to build better tools, robots, games, or educational environments. In another words, computer-based simulation is turning into a core body of a specific scientific corpus. The ability of interactive learning through simulation games can take a place in any curriculum that is designed to find an applicable alternate pathway for effective learning. Technology impact learning process and improve proficiency when used properly [1]. It has a power to shape the way people learn. Any simulation environments can take advantage of objective evaluation approximations. In this respect, any objective approximation which creates a measurable skill evaluation framework have a potential to turn Virtual Reality (VR) simulators into a trusted source to quantify the surgical skill that gives insights about the surgical capabilities.

Accordingly, this study attempts to implement the ML techniques in a VR-based simulation environment developed for surgical training programs. The main aim is to analyze the surgical residents' hand movements while they are working in the VR-based simulation environment. It is aimed at helping them better understand their skill levels through these analyses and providing appropriate feedback for their skill improvements during their training period. The study uses hand movement data collected from training sessions of surgical residents. As the participants' skill levels were already identified as novice or intermediate, and in the literature, there are very limited studies classifying these very close skill levels, the results of this study are

expected to provide important insights for the assessment process of these surgical skill levels.

This thesis study is organized as follows: Chapter 2 covers the background efforts. Chapter 3 describes the general information and direction of the methodology of this thesis study. Chapter 4 introduces the strategy and setting-up process of the framework. Chapter 5 contains information about the details of data analysis and goal settings. The feature engineering process and ML application are discussed in Chapter 6. Results and Analysis is the main subject of Chapter 7. Chapter 8 provides the discussion and conclusion of the study, and finally, limitations and future work are handled in Chapter 9.

CHAPTER 2

BACKGROUND OF THE STUDY

The background of this study is organized under two main headings. In order to better understand the framework for this study, firstly a systematic review study is conducted. This systematic review is conducted on the studies exploring experimental approaches, focusing on the roots of different simulation studies to observe surgical skill assessment, as well as the importance of hand movements in ML studies. Hence, in the first section, the results of the systematic review study is given where it is also published by International Journal of Medical Informatics [2]. In the second section, a background is provided to better understand how ML and feature extraction techniques can be used for skill assessment or hand-movement analysis.

2.1 A Systematic Review on Classification and Assessment of Surgical Skill Levels for Simulation-Based Training Programs.

The most vital result for a successful surgical operation is to reach an acceptable level of overall surgical process from diagnosis to postoperative patient care. Nowadays, surgical education tries to successfully bring as many surgeons as possible to this satisfactory level of performance. However, medical disciplines have been experiencing big challenges in their complex nature, parallel with the development of medical technologies. The natural complications of an ordinary open surgery directed this evolution towards the term minimally invasive surgery (MIS). MIS results in a lower number of complications, which means a shorter hospital stay [3]. Competency is a basic quality factor for a MIS procedure. Moreover, in any profession, a measurable skill is an important performance indicator. The transparency and measurability of a skill provides an alternate way to assess the surgical skill objectively which is an important challenge [4]. For example, Fundamentals of Laparoscopic Surgery [5] is currently considered as a standard in evidence based laparoscopic surgical skill training.

Systematic approaches provide an opportunity to successfully adapt organized, structured surgical educational programs as well as a chance to adapt the simulation environments in a proper way.

The technical maneuver of a surgeon is not the only inclusive property to define a surgical skill; hence, it is an important part of practical training because of its ability to quantify some properties of the surgical proficiency. For this reason, simulation studies play an important part in surgical training and are sometimes a must when there is a need for objective observations. However, the current literature is concentrated on innovative applications while sometimes losing connection with basic principles. In addition, turning virtual training into high clinical performance has been questioned many times, and this study evaluates it as a lack of understanding the evolution of simulation approaches, measurement techniques, and basic principles, therefore an interdisciplinary communication problem. For instance, diagnosis through rich, ready-to-use ML tools and algorithms may not be a true disruptive technology application if there are no basic ethical concerns. Traditional practices or principles, such as Halsted's surgical principles [6] regarding tissue handling, may provide insights for proper validation and verification of innovative solutions as well as current efforts to improve the effectiveness of surgical skill acquisition [7].

In the literature, arguing the proper way to improve the efficiency of the training methodologies of a surgeon candidate is a hot topic [8]. The traditional surgical skill investigation studies focus on the skill as a hands-on experience. However, the simulation-based training approximations provide an opportunity for surgical residents to gain insights before becoming professional surgeon. In addition, technology challenges the understanding of entrenched surgical proficiency as the most recent dynamic asset in the operating room. Such situations may force the training techniques to evolve in the direction of simulation-based training across surgical education.

2.1.1 Materials and Methods

In this study, the main aim was to review the surgical skill level comparison studies, keeping the connection between goal and result. Observing how the literature defines

surgical proficiency, it was important not to deep dive into the innovative applications that may take this research study away from its core focus. It would be more important to observe how a surgeon acquires a basic, measurable skill set before gaining the ability to achieve consistent good clinical results. Additionally, it is aimed to show how studies change over time to fit the terms, definitions, or other attributes against the evolving technology. The following research questions were aimed to be addressed by this study:

RQ. 1.1 What are the highest-cited article and total number of published articles, based on the web of science research area definitions, considering a specific time-period?

RQ. 1.2 How many articles were published per publisher classified by Web of Science research area definitions?

RQ. 2.1 What are the titles of the participants who take roles in the skill level investigation experiments?

RQ. 2.2 How are the attributes like gender, handedness or age investigated?

RQ. 3.1 How were surgical skill levels categorized in the literature, and which is the most used one?

RQ. 3.2 How are the surgical skill levels compared in literature?

RQ. 4 Which tools and training environments are used in these studies?

RQ. 5 What type of measurement techniques and metrics are used in these studies?

Study Design

The process started with article selection and ended with a systematic mapping result. It contains three main processes, as can be seen in Figure 2.1 [9]. “Search and Selection” is a filtering phase and the aim was to reach a final pool set of journals. A

preliminary set of attributes was used to reach a final map which contains a finalized set of attributes as in "Systematic Map Building" phase. Finally, "Systematic Mapping and Review" conducted to discuss and review the literature with the help of findings from previous activities.

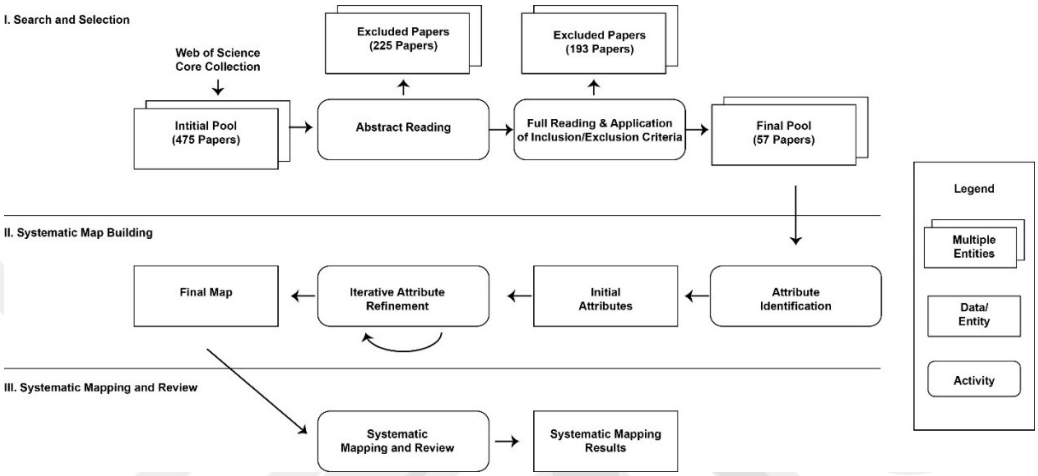


Figure 2.1 Systematic Mapping and Review Process

Search String

An automatic search strategy was followed within the Web of Science application. The search string was built around predetermined keywords with wildcard symbols.

Search string: TS= (surg* AND skill AND train* AND hand AND (endos* OR lapa*))

Timespan: All years.

Indexes: SCI-EXPANDED, SSCI, A&HCI, CPCI-S, CPCI-SSH, BKCI-S, BKCI-SSH, ESCI

Applying this search string brought 475 articles. In this phase, a direct abstract reading is applied for the creation of initial pool studies for filtering.

Abstract Reading Activity

The exclusion of unrelated articles is conducted according to the following questions:

- Is there any kind of surgical skill or training area to measure or review?
- Is there any kind of measurement techniques or tools used to measure or review?

The article was kept if there is a clear answer for any of these two questions, and the following question was implemented otherwise:

- Is there any skill assessment activity?

As a final check the article was excluded if there is no clear answer for this question. This phase is rather a subjective quality assessment phase depending on the reviewers' input. The output of the abstract reading activity was the exclusion of 225 articles.

Full Reading and Filtering

The selected articles are read in full paper form. At this stage the following inclusion and exclusion criteria were implemented:

Inclusion criteria:

1. The main objective must be related with the evaluation of surgical skills.
2. The full text should be reachable.
3. The article should be in English.

Exclusion criteria:

1. Robotic surgery or robotic assisted surgery articles will be excluded if they are not related to the ability to carry out a direct surgical maneuver.
2. Vague surgical skill categorization.
3. Lack of distinct surgical experience levels (studies containing only medical students as participants will be excluded).
4. Review articles will be excluded.

Finally, 193 articles were excluded and 57 studies were selected as the corpus of this study.

Systematic Mapping

Attribute identification was the first process after reaching a final pool with approved research materials. "Attribute identification 1" process contains a set of attributes (46

attributes) provided automatically by the Web of Science application (see Table 2.1). "Attribute identification 2" contains any possible manual attribute created by one author. A partial view of these preliminary attributes is provided in Table 2.1. These initial 98 attributes were identified and used as input for an iterative attribute refinement process in which other authors helped refine this crowded set of attributes set to reach the most relevant set of data.

Table 2.1 A Partial View of Attribute Identification Process

Attribute identification 1	**Attribute Identification 2**
Web of Science Attributes	**First Generated attributes**
Year Published	Paper ID
U1(Usage Count (Last 180 Days))	Attendee Info.
NR (Cited Reference Count)	Beginner
Web of Science Categories	Novice
AU (Authors)	Intermediate
Author Full Name	Sub-expert
Document Title	Expert
Publication Name	Advanced
Document Type	Proficient
Conference Title	# Of subjects

Iterative Attribute Refinement

In this phase, the relevancy of each attribute was analyzed by reviewers. Finally, 37 attributes were identified and organized into six groups (see Table 2.2). The overall applied systematic process was a combination of some other systematic mapping and review procedures [9]–[12]. Table 2.2 shows a preview of the final classification, which is also a picture of the final map. The goal of classification is to get a quick idea about attributes based on their groups. The final map contains completely numerical data except for three "bibliographic" attributes: "publication sources" "Web of Science categories" and "research areas."

A systematic map is built by this attribute scheme, and data is extracted accordingly based on research questions. Each pass is evaluated and reviewed against the erroneous entries by one reviewer and two evaluators. The results are handled in the following section.

Table 2.2 Final Attribute Organization

Classification by	List of Attributes
Surgical Skill level	Beginner, Novice, Intermediate, Sub-expert, Expert, Advanced, Proficient, Skill Comparison
Demographic Information	# of subjects, Gender Female, Gender Male, Median age
Handedness	Right, Left, Ambidextrous
Bibliographic Information	Publication Source, Year Published, Usage Count (Last 180 Days), NR (Cited Reference Count), Web of Science Categories, Research Areas, Total Times Cited Count
Surgical Roles	Surgeon (Expert), Surgeon (PGY<=3), Surgeon (PGY>=4), Attending surgeons, Junior Fellows, Attending Physicians, Expert Veterinarian, Veterinarian, Students, Medical Students, Cardiology or Vascular Residents, Cardiovascular Fellows, Lab technician - Operator, Doctor, Interns, Residents
Secondary Attributes	Simulation Environments (VR, Physical), Surgical Method, Training Area, Objective Measurement Tools (such as haptic, EMR and feedback), Surgical Tasks, Training Tools, Metrics

2.1.2 Results

The data collected through the systematical methodology described above was analyzed to find some answers for each research question of this study. Accordingly, the results are provided for each research question as below.

The highly cited articles and total number of published articles, based on the research areas in a specific time-periods? (RQ. 1.1)

Table 2.3 is organized through a time period that is specified in the upper row with related years. The N stands for the total number of published articles through the related time period. The C stands for a study with the highest citation number published during the specified time period.

For instance, the most cited study with 97 citations has detected between 2000 and 2004 and added to C cell, between related time intervals. The most cited and active research area is “surgery” surgical training, and how to assess and improve it. The most-cited study is from 2001. It is an “engineering” journal with 154 citations. A study from “surgery” with 97 citations, is the second most cited study. It may be possible to think that time is one good reason for an academic study to be a guide for other researchers. In both studies from different disciplines, the aim is to measure the

forces and torques at the interface between the surgeon's hand and the endoscopic grasper handle [13], [14].

Table 2.3 Citation analysis based on Research Area

Research Areas / Time period	2000 2004		2005 2009		2010 2014		2015 2020		2000 2020	
	N	**C**	**N**	**C**	**N**	**C**	**N**	**C**	**N**	**C**
Surgery	2	97	7	58	10	90	17	15	36	97
Engineering; Radiology, Nuclear Medicine & Medical Imaging; Surgery			1	0	2	13	2	9	5	13
Education & Educational Research; Surgery					2	18	1	12	3	18
Computer Science					1	21	1	0	2	21
Engineering	1	154			1	21			2	154
Otorhinolaryngology			1	21			1	1	2	21
Computer Science; Engineering							1	6	1	6
Computer Science; Engineering; Mathematical & Computational Biology; Medical Informatics					1	7			1	7
Mathematical & Computational Biology							1	4	1	4
Neurosciences & Neurology; Surgery					1	11			1	11
Obstetrics & Gynecology							1	6	1	6
Ophthalmology							1	0	1	0
Veterinary Sciences							1	3	1	3
Total, N	**3**		**9**		**18**		**27**		**57**	
Highest, C		**154**		**58**		**90**		**15**		**154**

N: Number of published articles through the related time period
C: The highest number of citations for a study published in the same time period

During the early 2000s, there were few publications. However, as shown by the N column in Table 2.3, a greater number of research papers have been published in the last decade.

Number of articles per publisher sources classified by web of science research area definitions (RQ. 1.2)

Table 2.4 shows each publication source grouped by research area. The count column keeps track of the number of related sources, while the % column shows the percentage of the related count among all selected primary studies. Surgery is a dominant category in research areas as well as among the "web of science" categorization (see Table 2.4). It may be inferred that the diversity of research areas indicates the growing need for interdisciplinary surgical skill investigation studies.

"Engineering, Biomedical; Radiology, Nuclear Medicine & Medical Imaging; Surgery" is an interesting categorization as it is a combination of multiple categories,

including “Surgery” with 5 published studies. It is also the second-most active research area after surgery. Computational background, engineering, and educational interest hold an important part of Table 2.4, which represents 26.29% of the total 57 research studies.

Table 2.4 Publishing sources based on Research Area

Research Areas	Source Abbreviation	Count	%
Surgery	SURG ENDOSC	21	36.84
Surgery	SURG INNOV	5	8.77
Eng.; Radiology, Nuclear Med. & Med. Imaging; Surgery	INT J COMPUT ASS RAD	5	8.77
Education & Educational Research; Surgery	J SURG EDUC	3	5.26
Engineering	IEEE T BIO-MED ENG	2	3.51
Surgery	AM J SURG	2	3.51
Surgery	AM SURGEON	1	1.75
Surgery	ANZ J SURG	1	1.75
Surgery	INT J MED ROBOT COMP	1	1.75
Surgery	J SURG RES	1	1.75
Surgery	JSLS-J SOC LAPAROEND	1	1.75
Surgery	LANGENBECK ARCH SURG	1	1.75
Surgery	MINIM INVASIV THER	1	1.75
Surgery	SURG ENDOSC-ULTRAS	1	1.75
Computer Science	IEEE T HAPTICS	1	1.75
Computer Science	IEEE T HUM-MACH SYST	1	1.75
Computer Science; Engineering	BEHAV INFORM TECHNOL	1	1.75
Comp. Sc.; Eng; Math & Comp. Bio.; Med. Inf.	MED BIOL ENG COMPUT	1	1.75
Mathematical & Computational Biology	COMPUT MATH METHOD M	1	1.75
Neurosciences & Neurology; Surgery	J NEUROSURG	1	1.75
Obstetrics & Gynecology	J MINIM INVAS GYN	1	1.75
Ophthalmology	J EYE MOVEMENT RES	1	1.75
Otorhinolaryngology	AM J RHINOL	1	1.75
Otorhinolaryngology	INT FORUM ALLERGY RH	1	1.75
Veterinary Sciences	VET SURG	1	1.75

Titles of the participants who take roles in the skill level investigation experiments (RQ. 2.1)

Experimental designs that contain only medical students are excluded to capture all the comparative skill-level studies. The medical students used as a sub-group in the experiments are included if there are other groups that can clearly provide expertise.

Table 2.5 Medical Titles

#	Titles	n	%
1	Resident	134	19.23
2	Surgeon (PGY<=3)	131	18.79
3	Surgeon (Expert)	128	18.36
4	Medical Student	108	15.49
5	Surgeon (PGY>=4)	83	11.91
6	Attending Surgeon	21	3.01
7	Doctor	21	3.01
8	Interns	21	3.01
9	Veterinarian Student	14	2.01
10	Veterinarian (Expert)	11	1.58
11	Cardiology or Vascular Resident	7	1.00
12	Attending Physicians	6	0.86
13	Cardiovascular Fellow	6	0.86
14	Junior Fellow	4	0.57
15	Lab technician - Operator	2	0.29
	Total	697	100

By means of the "skill evaluation" term, titles are important, because they are shaped through excessive education and training. Accordingly, for the corpus of this study, many medical titles were detected during the data extraction process (see Table 2.5).

Investigation of the attributes like gender, handedness, or age (RQ. 2.2)

40.35% of all participants are reported as female, while 43.5% are reported as male. Some task-oriented skills, such as hand-eye coordination or depth perception, are heavily investigated rather than subjected to gender-specific analysis [15]–[18]. The overall median reported age from 44 studies is 32.95. The average age of a surgeon in a definite position in a hospital, is around 36.8 (age range 30–45) [19]. The 91.58% of the overall reported 689 cases of handedness include right-handed participants, while others are left-handed. No ambidextrous case is reported from the final pool studies.

Categorization of surgical skill levels in the literature (RQ. 3.1)

The skill level labels are subjective parameters. One experimental design's "expert" surgeons may be evaluated as “intermediate” or even "novice" in different studies. It is important to notice that the skill level names in this section are dependent on a specific experimental setup. In other words, ‘expert’ does not always mean that the related attendee is an expert surgeon. This issue will be thoroughly discussed in the results section.

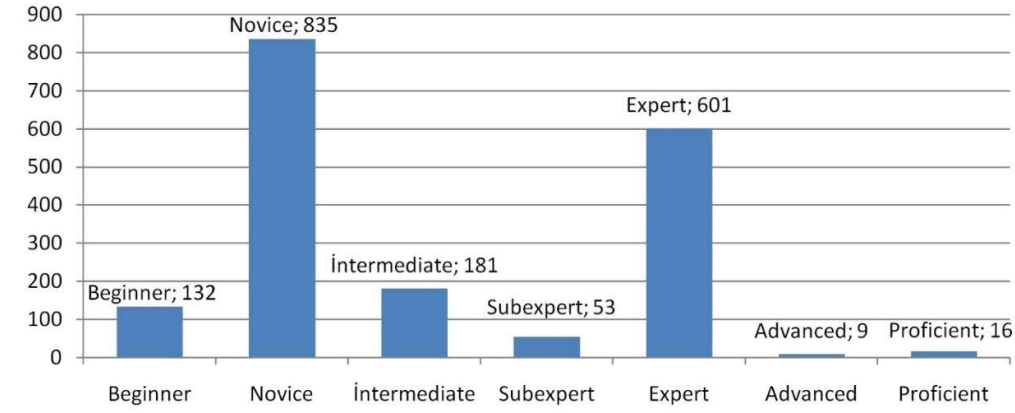


Figure 2.2 Skill Level Information

The most common categorization for skill level specifications is the "Novice" category, as can be seen in Figure 2.2 with 835 participants. Many studies are related to the evaluation of MIS-oriented tasks. The popularity of the "Novice" categorization is due to the availability of participants who are early in their professional journeys or are close to academic environments. Second, the "Expert" category is reported by 47 studies, and there are 601 participants in total. The popularity of surgical expertise is expected because the probability of the existence of many comparative studies looking for a standard value to measure and compare surgical talent is high. The findings support this argument with the extracted data.

The advanced skill level is used twice [20], [21], while the proficient level category naming is used only in one experimental evaluation study [22]. The surgical skill category "intermediate" which has 181 participants, is detected in 17 articles. Only 7 studies reported beginner category with 132 participants among 57 articles.

How are the surgical skill levels compared in literature? (RQ. 3.2)

Table 2.6 is organized with the help of the detected skill levels. Comparison has much potential in different applications such as validating developed measurement frameworks by discriminating a skill level group from others [23] or measuring the training capability of an environment [24]. The most common comparison that is observed in this study is "Novice vs. Expert" in 34 studies (see Table 2.6).

Table 2.6 Skill Comparison Table

	Beginner	Novice	Intermediate	Sub-expert	Expert	Advanced	Proficient	n
Beginner		*	*		*			1
Beginner					*	*		1
Beginner		*	*	*				1
Beginner			*					2
Beginner					*			1
Beginner			*			*		1
Novice					*			34
Novice			*					4
Novice			*		*			7
Novice							*	1
Novice				*	*			2
Novice				*				1

For instance, the authors correctly classify 84 percent of all participants as experts or novices using the force parameters [25]. In skill assessment studies, there may be some environmental effects on the evaluation phase, such as the task, experimental design, or tools used. Such shortcomings are pointed out in another study [26], where the authors discuss that residents of different specialties may not have equal laparoscopic experience. Furthermore, it is stated that the performance of basic psychomotor skills gained in a VR simulator training environment transferred to another VR simulator's task performance, and differences between medical students and experts were reduced [27]. Transferring basic psychomotor skills is a good example of a VR simulator's effectiveness.

Which tools and training environments are used in these studies? (RQ. 4)

The robotic assisted surgery techniques have been excluded to focus on the foundations around the surgical training topic. This also helps to match up with the research goal, which is to investigate the fundamental aspects of surgical skill evaluation and training.

The surgical skill evaluation studies, based on specific evaluation criteria, aim to successfully simulate surgical procedures for a research-oriented goal. Some concepts are visible frequently and contain heavily used terms in the surgical simulation studies, such as Augmented Reality (AR) or VR. At some level, it creates confusion around the definitions of AR and VR [28]. Some surgical procedures themselves may be defined as AR processes, in which an operator conducts a surgical procedure while getting information through a computer screen. This broad definition may be seen as

a more appropriate definition for the surgical simulation research area where the term AR is frequently [29].

For this reason, many studies provide mixed simulations, where participants see a VR model while working on a physical model. Any existence of a physical interaction detected during the analysis was added to the physical model section (see Table 2.7, Physical Model) [13]–[15], [20]–[22], [24], [30]–[61]. The VR concept is rather related to changing the reality to a computer-based simulated one. For example, creating a simulation that the user can manipulate or see using a 3D anatomic model, picture, video, or any virtual task is a common application (see Table 2.7, VR) [34]. Animal organ [44], or any physical simulation are counted as a physical simulation setup. There were 33 AR models detected during the full reviews [16], [17], [24], [30]–[33], [35], [37], [41], [42], [45], [46], [51]–[53], [58], [61]–[76]. They contain both VR and physical setups, and it is hard to put their inclusive definition aside while classifying the simulations. The surgical skill assessment domain struggles with definitions and technology to find a clear path.

Table 2.7 Tools and Environments

Environmental Setups	Total Number of Articles	%
Physical	39	68.42
VR	16	28.07
AR	33	57.89

What type of measurement techniques and metrics are used in these studies? (RQ. 5)

The detected measurement techniques and metrics are grouped into five different categories. One experimental design can exist in different groups. The “Performance” category is for any detected study with a realized serious focus on measuring or analyzing performance related activities [16], [30], [32], [34], [37], [39], [47], [51], [53], [54], [56], [57], [59]–[62], [66]–[69], [75]. In this category, “performance data”, “task performance”, “error data”, “task metrics”, “VR task metrics”, “task specific data” are major detected keywords. Table 2.5 is inferred with the help of detected applied metrics or analysis; to understand what the focus of experimental application is. Next, the “Hand Movement” row contains information about any hand movement

or related data measurement activities that are specified with keywords like economy of movements, motion analysis metrics, ergonomics, and the wrist movements [16], [24], [40], [43], [44], [47], [48], [50], [52], [55], [59], [70], [71], [77]. Another category is "eye movement" which includes the smallest number of studies focusing on analysis of the eyes and their movements [72], [74]. The "Skill" category contains specific objectives around skill evaluation such as laparoscopic skill assessment and FLS task-related metrics [20], [34], [41], [49], [56], [58], [76]. Finally, the "Force" category studies cover objectives around force parameters such as force feedback, haptic data, or related metrics [13], [14], [36], [37], [54], [73]. All related articles are stated in the column for each category as a reference. The column "Total Number of Articles" shows the total number, while "%" holds the percentage indicator showing the weight for the related category among all reviewed articles.

Table 2.8 Metrics and Measurements

Metrics and Measured Data	Total Number of Articles	%
Performance	21	36.84
Hand Movement	14	24.56
Eye Movement	2	3.50
Skill	7	12.28
Force	6	10.52

The internal structure of MIS procedures, which are directly measured or simulated, such as fundamental laparoscopic tasks [24], [40], [52], [73] or VR based simulation tasks, contain some AR approximations naturally combining real and virtual worlds [70], [73], [75], [76]. In computer-based VR simulations, the absence of haptic stimuli is also problematic [28]. In addition to introducing different haptic devices in the surgical assessment domain [70], [75], there are many attempts that may be thought of as the roots of haptic devices, heavily based on sensory force inputs [73].

2.1.3 Discussion and Conclusion

The skill evaluation studies aim to investigate how abilities are developed over time by applying some activities that are either part of special training or not. A basic question may be, "Is it possible to improve skills for real-life objectives using artificial environments?" For instance, the success of a candidate during a training session on

an animal model somehow may have a relation with the candidate's video gaming history [78]. It is possible to observe that some external activities can help improve skills. Skill transfer studies try to analyze such issues. However, it is hard to define improved hand eye coordination as a profound skill. It looks like the transfer of skills is a hot topic for this reason, as well as the increased number of VR applications for training purposes. More objective measurement techniques try to overcome this barrier by using more sensitive hardware or software and data analysis around the skill investigation issue. The results indicate the influence of significant technological innovations and the uninterrupted progress of computational and information technology on the surgical domain (see Table 2.1). The results also indicate that the technological advances are increasing the chances of new approaches in surgical education that entirely affect surgical skill evaluation. Moreover, the problem of surgical skill assessment may be at the center of such changes (see Table 2.2).

Learning itself is a complex process, and time is an inevitable need. For this reason, some external activities may be designed with repetitive tasks between long time intervals, which may give learners a chance to build up an observable difference in their cognitive capability. Transferring MIS skills to robotic surgery is another issue that has also been addressed in some studies with the advances in robotics [79]. Training is an important part of the skill development process, and surgical simulators can provide instant hands-on training [80]. Such approximations of the skill development processes can lead to better educational perspectives. For instance, examining brain activity to observe the effect of surgical [81] or gathering the hand gesture data of a surgeon is also gaining momentum for this reason.

The skill level categorizations need structured definitions with distinct study groups. Skill-level estimation studies may not show good congruence with each other about the naming of the experience levels. In a recent study, there is highly detailed demographic data about participants, and the participants were equally divided into two groups, namely, surgical "residents" and expert [73]. Surgical resident is a position rather than a label for an experience level category. The complexity of medical education is making similar approximations problematic to some degree because a

surgical residency role may contain hands-on experience as well as a candidate without any experience. Moreover, it is possible to come across a person, who may already have enough experience to be considered at an expert skill level and still have a residency status in another surgical domain.

Accordingly, the medical titles are hard to categorize into groups. First, the results indicate that participant identification is not clear in many studies. The 64.55% of the total experimental participants have no clear title information, despite their importance around in skill assessment problems. In addition, 6.51% of all participants are defined as expert surgeons in the experimental studies. This result shows that it is hard to reach expert skills. Notice that the expert surgeon's total number is low even if the reported "expert" participant's total number is high. This also gives an idea of how to name relative skill levels across different study designs.

On the other hand, skill levels can be categorized depending on the aim of the study, the target group, the experimental design, or historical data. It is possible to express even such small differences with systematic skill level categorization (see Figure 2.2). In an earlier study, the authors proposed a classification and definition for expertise and skill in [82]. They have defined a beginner as a person with "merely non-specialist knowledge of a domain", a novice as a person who started to develop the elementary knowledge in the domain [82]. Later, several research studies were conducted by adapting this definition to their research environment. Among them, some studies classified the surgical residents as novices if they had no experience conducting a surgical operation on their own and intermediates if they conducted some operations on their own [69], [70], [72], [74], [75]. Accordingly, earlier studies may guide the researchers toward a more standardized classification to increase consistency. Such an optimal framework may increase the effectiveness of surgical skill evaluation studies as well as make small skill differences measurable. The complexity of medical education and training can be a big obstacle to such an attempt.

As a conclusion, it can be reported that, majority (63%) of the studies conducted in this scope are classified under the "surgery" domain. The ones considered under

education (5%) and computer science and engineering-related fields in total (10.5%) are low (see Table 2.1). A higher level of collaborative studies with fields like engineering and education will provide several benefits for the endoscopic surgery education programs. Additionally, there is no standardized classification of the surgical skill definition studied in these research articles, considering the participant classifications and their skill level classifications as given in Figure 2.2. This limits the comparability of the results. Additionally, in the literature, there are limited studies comparing the skill levels of all groups of surgical residents. In the future, by developing some measures and threshold values specific to each skill level and each skill that needs to be developed by the surgical residents, more standardized and efficient educational programs can be developed.

2.2 How are Machine Learning and Feature Extraction Techniques Used to Understand Skill-levels or Hand Movements?

Nowadays, large datasets with many features have become a common problem with emerging digital innovation in the last few decades. Rapid growth of the data volumes naturally raises many questions. The high number of datasets flowing through many devices in complex internet infrastructures brings to the information technology discipline's desk new problems everyday with an expectation to reach solutions dynamically with diverse approximations. In a white report from Cisco [83], it is stated that there will be 13.1 billion mobile connected devices by 2023, up from 8.8 billion in 2018. Businesses are expected to adapt by focusing on integration through their infrastructure. The future of the internet is becoming more and more diverse, with all new requirements coming from different kinds of domain structures, software, and hardware. The change is inevitable, and the classical approaches may not be ideal for such big changes. The concept of "big data" emerged from such requirements when traditional data analysis methods were not enough to analyze or extract information from very large datasets. Traditional computer algorithms are also shaped by the need for such dense and complex streaming datasets. Businesses are reorganizing their processes to add AI to their information flow because of the ability to turn their large datasets into a business advantage. Human intervention has shortcomings for handling dynamic high-volume data flows through advancing communication technologies. ML

as a form of AI is a very popular tool for the academic community to improve themselves through learning and with the help of existing datasets without human intervention. The ML term has been a visible concept since Arthur Samuel's study. He provided an early application of ML concepts through his checkers game study [84] in 1959.

The learning process itself is a complex phenomenon that includes the gathering of new knowledge, the development of motor and cognitive skills, the organization of new knowledge into effective representations, and the discovery of new facts and theories through observation and experiments [85]. Many diverse approximations exist when the learning phenomenon is handled in the context of computational intelligence studies. What do we want a system to learn? There are many answers to this basic question based on the goal. Moreover, "How do we want a system to learn?" is a critical question asked many times that creates foundations for different research areas that have been closely related to ML for years. For instance, ML is different from AI in terms of how it gathers information to learn and take actions. Agents are used to interact with an environment in AI learning processes, so there is an active learning process. The intelligent agent learns from experience and makes appropriate choices given its perceptual limitations and finite computation [86]. ML processes, as a subset of AI, depend on mainly passive observations from data. Data mining is also another closely related research area with ML. However, data mining is different because knowledge discovery is the main goal. For this specific purpose, data mining approximations may use ML as a tool. "Unsupervised learning" is a critical term for this method. In a study, a simple example, which is sorting out apples from oranges by their appearances, has been provided for this term, and the authors use the definition "learning without a teacher", also known as self-organization [87]. On the other side, ML uses data mining methods as a preparation phase to improve accuracy if necessary. In addition, such foundations bring mathematical optimization problems to the table, and ML has strong ties to optimization problems. ML algorithms create a model with sample data. This model is used to make decisions later by the software. This sample data is called training data. A subset of ML theories is based on purely statistical learning methods. In any learning problem or data science approximation, the selection

of the best inputs is a preliminary event. These theories are part of the learning process for a computing "machine "to accomplish a task without programming for that specific purpose. The optimization algorithms and ML algorithms differ in the goal of generalization. This difference may help us to define a learning process for an ML algorithm. The training data that provides the experience for the machine and later the accuracy of the algorithm to perform on new sample data sets that it encounters in each new task define the basic learning processes. The selection of the training data with minimal error is closely related to the optimization algorithms; however, ML rather focuses on the accuracy for the new sample tasks based on the experience when we think about the generalization context. The learner machine must build a strong model with the provided training data to improve the accuracy on new tasks. ML approximations have been evaluated performance-wise in computational learning theory studies to understand the statistical bounds for their performances. Performance issues are investigated systematically because the nature of ML applications depends on a finite set of training data and unknown future samples. The performance of ML methods is also heavily dependent on the choice of data representation (or features) on which they are applied [88].

2.2.1 The Core Terms and Definitions

ML algorithms try to gain experience with the help of known data. A learner process is employed to gain experience for this purpose. This experience and generalization ability, gained from training data, were later used to make new generalizations on new datasets. For example, a supervised machine learning algorithm can simply be defined as a target function that maps input variables to an output variable. This definition itself may reveal problematic limits because it gives bounds for the prediction or classification. Therefore, there is a range for the success of the model. The performance of ML models is related to the success of the target function and how effective it is at generalizing unseen data.

Features and the Learning Problem

In a statistical manner, density, mass, or time can be examples of independent variables that also have no dependency on other variables in a related experimental environment.

Independent variables are called predictor variables, explanatory variables, or regressors, depending on the domain in which they are handled. Similarly, the independent variables are defined in the ML domain as features. They are independent variables with no dependency on other variables, and they process like discrete inputs for a ML system. Then the predictions or classifications are conducted using these features. In different ML models, features are used to detect other features. Statistical classification perspectives are the subject of investigation in the ML domain, and features can be defined as feature vectors in such works. For example, there are two categories predefined for a supervised ML algorithm to predict future observations. This is a problem called binary classification. A feature vector is used as an input in this situation. Binary classification uses another ML term, the linear predictor function, which combines a set of weights and feature vectors to be used to predict the outcome calculated with other dependent variables. In a typical application of ML approximation, the number of features depends on the problem specification as well as the data characteristics. Feature space contains a number of feature vectors.

The ML literature tries to overcome the complexity of the learning process, which is a goal-oriented problem with conditions that have been analyzed in the literature for years. Training datasets sit at the center of such approximations, in addition to the features. Improper training datasets are a critical issue because they result in problems that are difficult to solve later as well as feature selection that is irrelevant. Feature selection or feature construction is also an important foundation, and existing improper features may create a domino effect that makes the learning process fail. Feature construction is the process of creating new features from existing ones. It helps to filter a feature vector by counting its elements under some conditions to build up a new one or using generalizing approaches inferred from a feature vector. Such applications offer new ways to produce rich feature pools, but they also lead to some other issues around how to organize them to fit the learning problem. For instance, the dimensionality reduction methods are used to reduce the dimension of feature spaces to make them a good fit for the sake of accuracy gain through overall ML application. Moreover, the large number of features with too much detail may cause problems. It may lead to other problems, like inefficient learning or overfitting. Feature abstraction

or feature extraction aims to keep the feature pool in the sweet spot to reach a highly accurate output. It helps to reach a feature space at an acceptable level. In a typical ML application, the first sets of features are not used as input directly because they are not suitable for reaching accurate patterns, reducing learning problems, or reaching high generalization capabilities. Therefore, the selection, construction, or extraction of features is a fundamental step in ML techniques, and it is sometimes called "feature engineering," which focuses on the discipline of how to present data to a learner.

Generalization in Machine Learning

In the inductive reasoning method, antecedent opinion is viewed as a source of information, but it is not guaranteed to lead to a conclusion. Statistical generalization is an inductive reasoning argument in which a conclusion is inferred using a statistical sample. Such arguments provide a basic characterization of ML approximations. The ML wants a machine to learn from specific input to realize general concepts, exactly like the induction term, which is to reach general concepts from specific examples. To make more accurate predictions, generalization context is the key for ML because the training data provides only bases for later inputs. There are two basic facts that may be deduced from basic ML methods: there are limited training datasets and the future is not known. The foundations of computational learning theory are related to these facts in order to analyze the success of ML algorithms and to discover the limits of learning ability by induction. The generalization problem is mission-critical for ML processes. It may be inferred that the algorithmic approximations try to develop hypotheses for each unique case they encounter, which means that success is not an exact result but rather a probabilistic value from the algorithmic perspective.

Goodness of fit for ML Methodologies

The goodness of fit is a statistical model and is not directly part of the ML terminology. It refers to measuring how well a statistical model fits a set of data. Goodness of fit focuses on analyzing the model by means of inconsistencies between observed values and the model's expected results. The definition itself looks like an exact fit for the ML domain, but the statistical goodness of fit approximates the known target function. In the case of ML, the aim is to learn the target function with the help of training datasets. Therefore, the goodness of fit for the ML methodologies can be defined as

how well they learn an approximation from training data. The performance of ML approaches is mainly investigated around two terms that are based on statistical modeling: overfitting or underfitting data.

In the statistical context, overfitting is the process of reaching so many parameters that a set of data cannot fit. A ML model tries to make predictions or classifications by learning approximations based on training data, and that sample data may provide so much detail that it affects the performance negatively, which creates an overfitting problem. In this case, the ability to generalize is not applicable for unseen samples. It is because not only is the training data detailed but also it can be noisy or the probabilistic distribution is unknown. Overfitting is more visible if the model's learner has more flexibility. Some techniques are well known to limit how much detail is learned. For instance, pruning a tree is a data compression technique that is employed to avoid overfitting problems in nonparametric decision trees. Decision trees in noisy domains need "pruning" to reach better generalization performance [89].

Underfitting, in the statistical context, happens when the statistical model has no ability to adapt to the data structure. Similarly, a ML model is underfit when it has no ability to learn from the training data or generalize to new data. There is no special technique to avoid the underfitting problem because it is easy to detect with performance analysis and gives an obvious poor performance. The general tendency is to apply and measure the performance of the model before trying another ML method. In addition to the underfitting and overfitting problems, time complexity analysis and feasibility studies are conducted to measure the performance of ML approximations.

2.2.2 ML Approaches

There are many ML approaches depending on the ML system design. Traditional approximations are investigated in three basic paradigms. They are supervised learning, unsupervised learning, and reinforcement learning.

Supervised Learning

Supervised learning starts with a chosen mathematical model based on existing training samples that contains a set of inputs and an output. Each sample is represented by an array or a vector, and they combine in a matrix called training data. The provided data supervises the algorithm to build a strong model for future samples. A successful ML system finds the desired output for new samples with the help of the experience gained from the provided training data. The learning process is an iterative approach, and it is expected to optimize the function for the best output for each new case. An improvement in the accuracy of the output over time is an accomplishment for the algorithm.

Active learning is a kind of supervised learning algorithm. The key idea behind active learning is defined in [90]. The authors say that a ML algorithm can achieve greater accuracy with fewer labeled training instances if it is allowed to choose the data from which it learns. In some cases, labeling is inexpensive or free. For example, labeling spam mail is provided to the learner as a flag, so it has no cost. However, supervised learning systems may have a huge number of labeled instances in some complex systems, such as speech recognition systems, which are extremely time-consuming and require trained linguists [90]. Active learning systems try to overcome this labeling problem by providing some groups of unlabeled instances in an interactive query form, which will be labeled by an information source. The main goal of the active learning system is to minimize the labeled training data, which would otherwise produce a huge cost.

Classification is another problem that a supervised learning algorithm also deals with. The classifier is an algorithm or a mathematical function that turns the input data into a corresponding category. In the previous spam email filtering example, the output was restricted to a limited data set, and it is a popular example for the classification algorithms. Hence the classification problem is about which data belongs to which known category. A classifier often uses observations to distinguish the quantifiable properties of the data. In ML literature, those observations are known as instances. Feature vectors are also an ML term, which is a structure that holds the data properties,

and the goal of the prediction is the classes, which we mentioned as categories. In other words, the ML classification system recognizes patterns in the set of input variables, which is provided in the form of a feature vector, and maps them into the right classes.

Regression algorithms are another example of supervised learning approximations. It is a well-known statistical model to estimate the relationship between a dependent and one or more independent variables. The regression algorithms are suitable when the ML system gives a numerical output within a range.

Unsupervised Learning

The unsupervised learning process is an exploration of the data without even knowing its structure, as we mentioned in the relation between ML methods and data mining studies. The learning algorithm starts from scratch and tries to explore the data without any labels. Unsupervised learning itself has been turned into the goal of some ML techniques to reveal hidden features from data, and an example is called feature learning or representation learning [88]. Unsupervised feature learning tries to detect features or classify the raw data automatically. Those attempts enable the machine to explore the data on its own, without any classification to perform a task. In unsupervised learning methodologies, each data piece is an information source at the same time for the identification of its own characteristics. The comparison of each piece of information source results in a pattern, and the algorithm reacts based on the knowledge discovery.

Another common example of knowledge discovery is cluster analysis, which is an area of research in different disciplines. The aim of the cluster analysis is to organize similar objects into groups. The same cluster means that the objects have more common properties than other groups. Cluster analysis is called "clustering as a general term, and it is not one algorithm or model. It is more like a task for which many research fields build up models or define algorithms to reach an exact goal for a specific objective. In the ML perspective, clustering is grouping the set of observations depending on a predetermined criteria pool.

Reinforcement Learning

Reinforcement learning in ML provides a framework that makes software agents evaluate the environment to be able to act against it and get rewarded for it. In each step, the process state aims to maximize the reward. It is an interdisciplinary general framework. In machine learning, the form of environment is a Markov decision process (MDP). In MDP, the decision-making process is partially controlled by the decision-maker. In an MDP state, an actor decides on an action. Then the state changes randomly, and a reward is provided for motivation to proceed to the next state. Reinforcement learning is used in autonomous vehicles and manufacturing because of its automation applicability.

Other ML Paradigms

There are ML approximations for the situations that cover a problem but are not a fit for traditional learning categories. For example, "meta learning" allows ML experts to apply the multiple learning algorithms to the data that comes out of ML exercises. It opens a way to explore new perspectives for improving existing learning algorithm performance as well as investigating the learning process itself in depth as a subfield of ML. Hence it is defined as "learning to learn" in different studies [91]. Semi-supervised learning is another example of different ML applications. It is used when supervised and unsupervised algorithms are both lacking to cover the related problem. In other words, some of the training data is missing labeling, so it is used with the labeled training data. It is also shown that learning accuracy can be significantly improved with proper semi-supervised learning applications [92].

Another early ML methodology introduced an architectural perspective called Crossbar Adaptive Array (CAA) based on artificial neural networks, which contains a "self-learning" system in addition to aiming to solve a problem called delayed reinforcement learning for neural networks [93]. The reinforcement learning paradigm is employed in a secondary role to provide emotion towards the output instead of active communication with the environment itself. One "situation" is provided as an input, and an action or behavior is reached as an output without environmental impact. Then the emotion is a consequence, which is attributed as an evaluation context. It looks like

the "self-learning" paradigm has inspired some other ML perspectives, which result in diverse approximations directed to explore different angles. For example, "Robot Learning Driven by Emotions" is a study that introduced an emotion model integrated into reinforcement learning architectures to build an emotion-dependent architecture [94]. The authors define the three roles of emotion as an attribute of reinforcement: influencing perception, providing reinforcement value, and determining when to reevaluate decisions. It may be argued that the more the cognitive depth of human beings is explored, the greater the scientific curiosity about behavior-emotion interaction as a function of artificial proxy.

Robot Learning

The robot learning algorithms are also getting richer with the developments in robotics. Such algorithms generally aim to acquire necessary skills through a kind of self-learning architecture or with guidance. The learning process happens through direct interactions or the self-learning routine, which results in a series of gained experiences. Advances in the automated learning capabilities of ML methodologies, for instance, a robot discovering its environment, look like an answer to an old question: "How can a machine learn on its own and cheaper?"

Representation (Feature) Learning

The representation learning paradigm, also called feature learning, is another ML approximation that gathers many different algorithms and mathematical models together in principle. They generally add an automated knowledge discovery layer as a preprocessing phase of ML applications, which makes manual feature engineering less important and allows the learner to learn and transform input data automatically. The representation of the data is critical for ML paradigms because accuracy generally depends on properly defined data features. The aim of reaching a better data feature discovery process for the training datasets is a quality development process for ML approximation itself. Learning the data features increases the chances of the success of a machine using these features more effectively to perform a task or classification by providing a convenient foundation.

The importance of data representation for ML approximations comes from the relationship between performance and the data preparation process. Hence, ML experts spend more time to improve the design of the data representations, transformations, and adaptations. However, the lack of capabilities of manual feature analysis results in a natural tendency for automated feature learning methods. Such efforts open a larger area for ML research by simplifying the extraction of useful information from raw data and, in addition, increasing the performance of ML algorithms to accomplish a specific objective. For instance, autoencoders are an unsupervised feature learning method, which is a type of neural network. They learn to copy and compress its input into a code, then try to decode it to recreate the original input. There are different approximations for autoencoders, which may take noisy original data as a training sample and try to create a less noisy output. Autoencoders are also a feature extraction method because they are capable of reducing high-dimensional projected data into low-dimensional data with their non-linear transformation methods.

Sparse dictionary learning (sparse coding) is another example of feature learning methods. A sparse matrix is a numerical analysis structure that contains many zeros as elements. The sparsity is a measure for the data structure to be defined as a sparse matrix, which is calculated by dividing the number of elements containing zeros by the total number of elements. The matrix is called dense if most elements have a non-zero value. The sparse coding in ML tries to realize this sparse distribution among the combinations of elements, starting with the basic training dataset. These elements are stored in dictionaries. The advantage of sparse coding is its ability to learn representation, which can cover more inputs. In other words, it can represent more input regions using fewer parameters [88]. Sparse applications find their place in signal processing, machine learning, and image processing. The ability to detect the sparse representation with a smaller number of inputs results in efficient algorithms. Sparse coding algorithms use a small number of features among a large number of detected ones. Most of the features are equal to zero, and only suitable ones are selected to reveal sparse representation. For instance, the application of the same algorithm can learn to detect different edges of an image as well as learn different frequency patterns

for a given speech audio record [95]. To realize the importance of sparse coding schemes, a car owner who thinks her or his car alarm system is activated because of an auto burglary event may be an example. However, the alarm system is activated because of a storm, which is another rare event. These two distinct events may have different probabilities, but they have the same consequence. Application of sparse coding, whether probabilistic or not, provides a directed graphical model that can reveal many patterns to reach the same goal.

Artificial Neural Networks

The inspiration for artificial neural networks (ANNs) is biological neural networks. A standard neural network (NN) consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations [96]. ANNs combine with a series of connected artificial neurons. Transmission between neurons is carried out using signals. Each neuron is connected by edges up to some other neurons. The output signal is calculated with a nonlinear function based on the sum of the inputs. Each node and their connections have a weight, which is adjusted through a learning process based on strength. The signal transmission may be triggered based on a condition. The structure of neurons in an ANN may be designed with layers depending on their function. The examples are used to train ANNs, and each of them contains inputs for the neuron's function, which produces an output depending on the neuron's properties like weight. The ANNs, as a supervised learning approximation, have been provided a target output, and the learning process tries to adjust the overall network output according to the goal based on weighted associations. The weighted associations are calculated using the difference between network output and target output in addition to the properties of the learning process. Each adjustment is expected to increase the similarity between the network output and the target output until a predefined similarity degree is reached, which terminates the overall process. Examples provided to the ANNs take a critical place in the learning phase. The ANNs are not generally programmed explicitly to learn from data; instead, they are dependent on the examples provided.

Deep Learning

DL methods are based on ANNs using feature learning approximations. Different kinds of deep learning approximations exist and are applied to a variety of fields, from automation to board games. The AlphaGo program of Google's Deepmind Company is a popular example of a superior learning machine that surpasses human expertise. Go is a complex game with a high number of possibilities. A combination of an advanced search tree and deep neural networks is used to reach the results. The learner picks up a representation of the board as an input and processes it through many connections in different network layers [97]. The "deep" terms came out of this layered structure of NNs, and the higher levels create more general features than lower layers using raw data. Therefore, the different layers take different groups of independent variables, and these rich knowledge perspectives increase the capability of the learning machine's accuracy to higher levels. Similarly, AlphaGo's "policy network" gives an output as a decision maker for the next move, while its "value network" predicts the winner of the game.

Association rules

In association rule-based learning, a method called "interestingness" is an early example of knowledge discovery that discovers the dependencies of variables in large databases. Support, confidence, lift, correlation, and collective strength are a set of sample metrics used in this associative pattern discovery process [98]. In ML perspectives, the machine learns the discovered rule-based dependencies and improves them gradually for a specific objective. Learning classifier systems (LCS) provide a set of rule-based ML algorithms that take advantage of both discovery and learning processes together. A genetic algorithm is generally employed as a discovery method based on the rules in collaboration with the learner to reach a more flexible learner mechanism [99]. Inductive logic programming (ILP) is another approximation that puts the rule-based learning mechanism together with logic programming as a common feature framework. The ILP system tries to develop hypotheses based on compressed knowledge and examples. It is rather a philosophical perspective than a mathematical approximation, which needs strong rules and a properly defined set of data. The

advantage of ILP is that it provides a higher level of representational view for the problem domain, depending on the strong dependencies between data variables.

Decision Tree Learning

In decision tree learning, the tree's branches are observations, and the target value is leaves. Branches and leaves are input features and class labels, respectively, in classification decision trees. The specification of the target value, aka leaves, depends on the decision tree's learning model. In classification trees, the leaves take a set of class labels, while in regression trees, they take continuous values.

Support-Vector Machines

Support Vector Machines are a kind of binary classification technique using a combination of supervised learning methods. A model was developed based on training data to predict the new sample's class.

Bayesian networks

Bayesian networks are a kind of directed acyclic graphical model. A network is used to calculate the probability of relationships between input and output by using independent variables based on conditions. Inferences are introduced to the learning process with these relationships. The Bayesian networks are capable of handling the decision-making process without certainty or dealing with sequential datasets with some improvements in the modeling of the ML approximation.

Genetic algorithms

Genetic algorithms provide an evolutionary perspective to optimization and search problems in the ML domain. They try to mimic natural selection with heuristic approximations. Like the mutations that generate new genotypes in natural selection, the mutation operation in genetic algorithms tries to generate new genotypes.

2.2.3 Feature Extraction

The natural tendency towards data analysis is dealing with large datasets. Those large datasets require huge computing resources to process. Features are a way to present these datasets at a higher level. In machine learning, the dimension of a feature is

defined as the total number of variables. Moreover, feature extraction is a core process to combine or select this huge number of variables into higher-level representations, aka features. In other words, it is a dimensionality reduction technique. It is also a common solution for the overfitting problem, which is mentioned in the core terms and definitions section. An accurate feature extraction process turns relevant raw data into a higher level of presentation completely by reducing the number of features and creating easy-to-manage new features without losing any necessary data. The need for huge computing power reduces with the reduction of data volume, which is represented at a higher level. In addition, this dimensionality reduction technique can also result in reducing the amount of irrelevant data, which speeds up ML processes like generalization. The learner faces a summarized version of the feature set. For instance, the words to conduct natural language processing and the different shapes in image processing are features extracted and classified by related approximations.

Principal component analysis

The Principal Component Analysis (PCA) are a vector sequence, which enables the application of linear algebra approximations to define a different projection of the data with reduced or refined resources. PCA is a frequently used feature extraction method because of its dimensionality reduction capabilities. The projection of principle components gives a new feature set without losing information. To make this happen, some matrix operations are used as tools, such as covariance matrix. Covariance is a statistical term that measures the variability of two random variables. Similarly, PCA is an effective way to analyze the variances in a data set depending on the covariance between random vectors. Calculated pairwise distances minimize error with the help of the best-fitting line, while data is projected into orthogonal components to maximize variance. The orthogonal coordinate system describes the variances with a rank based on importance. PCA is a popular linear feature extraction method that takes original data and tries to find a combination of features efficiently. Some other methods are developed based on PCA such as the Kernel PCA or principal component regression. The Kernel PCA is an extended PCA method using Kernel methods. Kernel PCA enable to use a feature space to compute images of data pairs. Another extension deals with a multilinear extension of PCA (Multilinear principal component analysis). The

principal component regression (PCR) is a statistical perspective which is also influenced from PCA however it uses principal components as regressors to calculate the unknown coefficients in a linear regression model. Therefore, Partial least squares (PLS) have similarities with PCR however it focuses on finding a linear regression model using a projection of observations and prediction into a new space instead of the data variables. PCA provides foundations for different models in addition to the wide range applicability for diverse studies.

Independent component analysis

Independent Component Analysis (ICA) is another feature extraction method that searches the linear transformation similar to PCA, but it also minimizes the statistical dependence between its components [100]. In statistics, "multivariate" means including observations and analysis at the same time for multiple outcomes. ICA is useful to analyze such multivariate data. Classical feature extraction methods may fail to cover mixed data with and without dependent variables. In such cases, ICA is applied to find mixed sources and other data components. It increases the ability to discover knowledge. For example, a speech recognition system with conversation data between two different persons that aims to identify each person fits in the ICA approximation. ICA's is a data-driven method that does not need labeled organized data to differentiate the mixed target data source in contrast to the classical approximations such as classification, thus it may be defined as an unsupervised ML method. Signal or frequency analysis are sample application areas for ICA methods.

Isomap

Isomap is a nonlinear feature extraction methodology. An isomap algorithm determines the neighbor of each point to create a weighted graph under some conditions to compute the shortest path between two nodes and to map a representation based on the similarity of individual cases. The realization of manifold structure in data distribution based on this algorithmic perspective creates an applicable method for a wide range of cases. Even a single short-circuit error can lead to an incorrect low-dimensional embedding [101]. Therefore, the preprocessing of the dataset should be handled carefully.

Latent semantic analysis

Latent semantic analysis (LSA) aims to analyze documents and their content to create concepts related to them based on the assumption that close-meaning words exist in expected places in text. Hence the application area of LSA is natural language processing.

Multifactor dimensionality reduction (MDR)

MDR is basically a statistical model that focuses on the combination of attributes or variables that have a relation with a dependent variable. The MDR approach has no hypothesized statistical value because it is parametric; there is no assumption for inheritance, and it is directly applicable to case-control and discordant sib-pair study designs [102]. MDR algorithm uses exhaustive search for knowledge discovery, and it has a classifier to identify the best combination of specific prediction processes [103].

Multi-linear subspace learning (MSL)

Multi-linear algebra is a mathematical term that extends the theory of vector spaces into an element of algebraic systems, which produces an algebraic construction used in a higher-dimensional projection of geometric studies. MSL provides higher-order dimensional generalization capability for linear feature extraction methodologies like PCA or ICA. The tensor is another important algebraic object that describes the other objects in a vector space. In MSL, data tensors are used to organize data observations to conduct the feature extraction process.

Semi-definite embedding (SDE)

In geometry, a subfield of Euclidean space is called convex if any two points can join a line segment. Convex optimization focuses on minimizing the convex function over a convex set. Semidefinite programming is also a part of the study area of the convex optimization problem. SDE uses this programming technique to conduct a non-linear dimensionality reduction. The aim is to map this high-dimensional vector space into low-dimensional space.

Linear discriminant analysis (LDA)

In statistics, Fisher's linear discriminant is a method to find linear sets of features that produce different combinations depending on more than one object class or event. LDA uses these combinations either for ML classification or feature extraction techniques. LDA and PCA are two close approximations for feature extraction based on their goals. However, LDA finds combinations of variables with different class specifications to reach a representation of the dataset, whereas PCA tries to find different feature combinations without class specifications to look for only differences. The goal of LDA is to maximize the between-class measure while minimizing the within-class measure [104]. Maximizing the between-class measure increases the classification performance, as does following a normal distribution.

Locally Linear Embedding (LLE)

A manifold is a topological space with n dimensions that can be embedded into a higher-dimensional space. The manifold learning focuses on the idea that the datasets are generally unnecessarily large. LLE is a non-linear feature extraction technique based on manifold learning approximations. The aim is the assumption that high dimensional space can be represented in lower-dimensional space, aka manifold.

t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is basically a statistical visualization technique so that a high-dimensional space can be visualized using located data points in two-dimensional or three-dimensional space. In feature extraction, the visualization of high-dimensional data is embedded into lower-dimensional space.

CHAPTER 3

METHODOLOGY

The main motivation of this study was the expectation not to lose the interdisciplinary understanding between basic surgical skill definitions, surgical education, and simulation-based solutions. Through an established surgical training curriculum, better, more consistent clinical outcomes could be achieved. However, it is important to build a methodologic structure to evaluate the surgical skill to make it better. Accordingly, this thesis study was planned in a way to build a framework to be able to distinguish the intermediate and novice surgeons purely depending on the data-oriented perspectives and ML models based on VR simulator application. The collected data is based on the experiments conducted with 28 participants. First, a research framework was introduced to investigate the substructure of this research. A reproducible and repeatable framework which is an important asset. Then the data preprocessing is conducted to reach an easy-to-use proper data architecture and a sample run through the data processing pipeline is investigated. Next, to increase the applicability and quality of the results, a feature engineering process is designed. These features and all datasets are then used for predicting the intermediate and novice surgical residents using a ML model and the related routine. Finally, the comparison of all feature sets and data processing approximations, and the integrity of the proposed framework is used to reach a high accuracy. The research procedure of this thesis is given in Figure 3.1.

3.1 Experimental Application

The Educational Computer-based-simulation Environment (ECE) has been designed and developed as a capable and flexible framework to simulate any targeted educational procedure. The project contains different simulation exercises with the help of necessary laboratory equipment from haptic devices to VR sets. The ECE team

changes according to the project and the experimental design. Sometimes an interdisciplinary project contains many different stakeholders in a wide range of academic background. This study focuses on the experimental designs containing four different scenarios developed using the Unity3D game development environment and C# and a computer setup with two haptic devices. The goal of these scenarios was to measure the participant's hand motion as well as their performance.

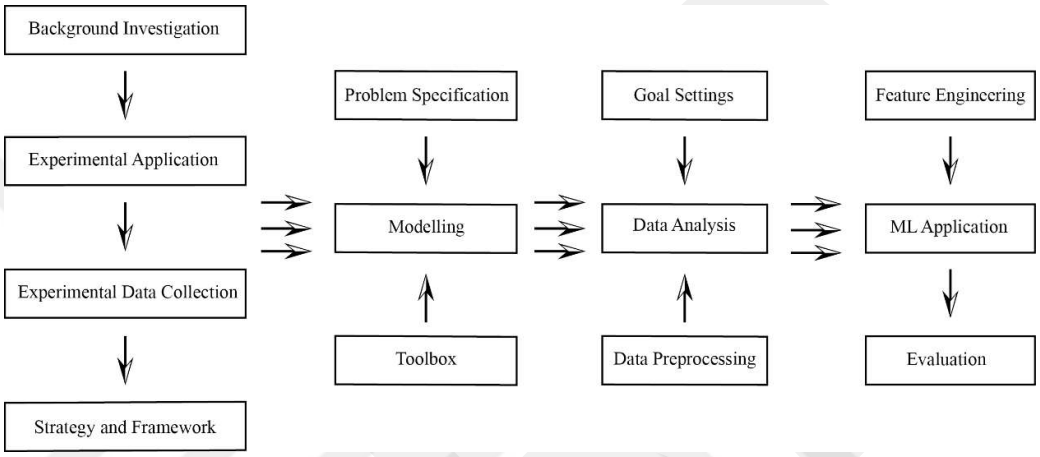


Figure 3.1 Research Procedure

3.1.1 Experimental Setup and Procedure

The experimental setup contains a computer screen, two haptic devices and an eye tracker. In this research study, the eye tracker data is ignored because the aim is to reach an acceptable level of accuracy depending on hand movement-oriented datasets. The brand of the haptic devices is 'Geomagic Touch'. The motorized structure gives force feedback ability while manipulating objects in a 3D design which increasing the realistic sense in the process. A software development kit called "Open Haptics" makes the design and development of haptic-oriented applications easy. It is compatible with Unity3D.

Demographic information like dominant hand, experience level had been collected before the experiment. All participants were informed in detail about procedure. They were provided an instructional video as well as oral explanations for the procedure.

Moreover, they all had a chance to practice on a sample scenario called “Using a Haptic Device” to increase familiarity about haptic devices.

3.1.2 Participants

21 doctors and 7 interns had been participated in this study. Two skill level categorizations were determined as novice and intermediate based on the operation room experience. Novice participants had no previous endoscopic surgery experience and they only assisted or observed endoscopic operations. Novice group contains 16 participants with an average age 26.94. Each of them was a research assistant in the neurosurgery or otolaryngology departments. Their average observation value was 12.69 and assistance value was 5.75 in operating room. The intermediate skill level definition contains participants who had operated a surgery at least one operation [82]. The intermediate group had 12 participants with an average age 30.15. They performed operations with an average value of 16, as surgeons.

3.1.3 Scenarios

There are four basic scenarios had been designed to measure the surgical capability as mentioned. First two experiments concern adaptation which makes participants to gain a depth-perception and general control ability. The others contain a simulated anatomical model which is more realistic experimental approximation.

Scenario-1: Moving the Ball into the Box

One handed scenario 1 (see Figure 3.2, A) is built in a cube shaped room and with an endoscopic tool. The camera and lights are in a fixed position for this experimental setup. The start of the scenario brings a randomly positioned red ball and a green cube as a first task. The participant should hold the ball with the tool by touching it and move it into the center of the cube. Once the ball touches to the exactly center coordinate of the cube in ten seconds, the task is completed successfully else it is failed. Successful or failed task, immediately brings a new cube and a red ball at random locations in the cube shaped room. Then the process continues for ten tasks and the data saved as a text file after the completion of all tasks. This experiment will be repeated for dominant and non-dominant hands (see Table 3.1, ECES0101).

Same scenario is applied again with two haptic devices for both-hand experimental setup (see Figure 3.2, B). This time the participant controls the tool with dominant hand while controlling the camera and light source with non-dominant hand. Note that this experimental setup (see Table 3.1, ECES0102) is a little tricky because participant should focus the camera and light source into the target with non-dominant hand while trying to catch the ball with dominant hand.

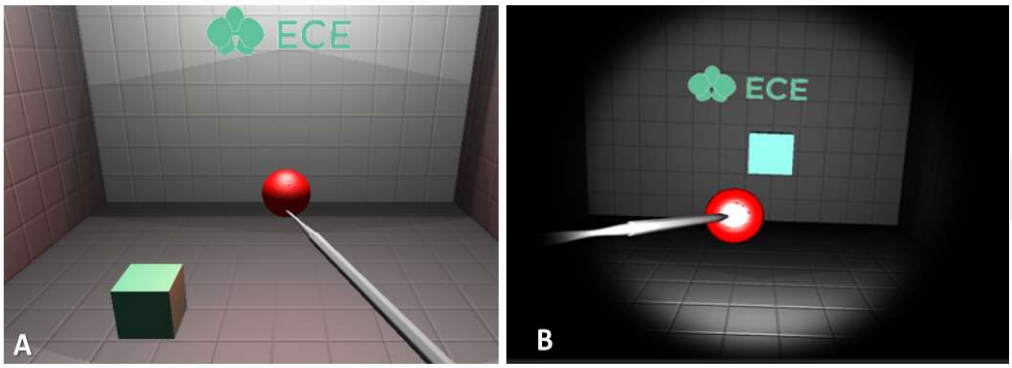


Figure 3.2 Moving the Ball into the Box (Scenario-1)

Scenario 2: Catching the Balls in boxes with an Endoscope

One handed scenario 2 (see Figure 3.3, A) is built in a cube shaped room again but the camera is used as a tool this time. The lights are in a fixed position for this experimental setup. The start of the scenario brings a red ball inside a random blue cube as a first task. The participant should move the camera into the blue cube and touch the red ball to complete the task. However, the camera should reach the red ball with a right angle from the open side of the blue cube else the task will not be completed. Once participant manages to touch the red ball properly with the camera in ten seconds, the task is completed successfully else it is failed. Successful or failed task, immediately brings a new red ball in a random blue cube. Then the process continues for ten tasks and the data saved as a text file after the completion of all tasks. This experiment will be repeated for dominant and non-dominant hands (see Table 3.1, ECES0301).

Same scenario is applied again with two haptic devices for both-hand experimental setup (see Figure 3.3, B). In this setup (see Table 3.1, ECES0302), the participant

controls the camera with dominant hand while controlling the light source with non-dominant hand.

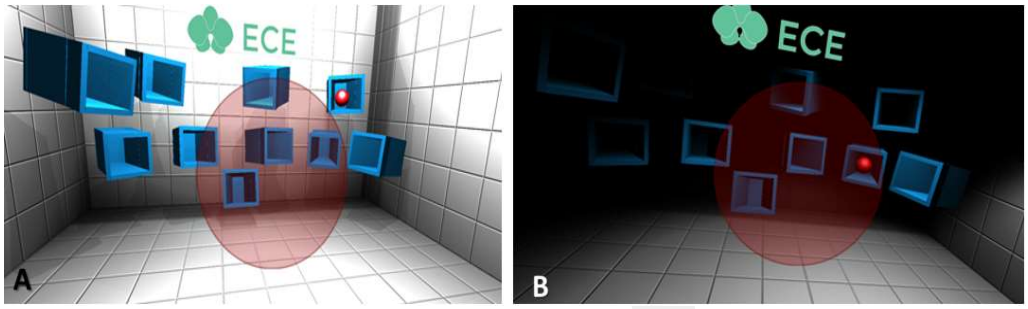


Figure 3.3 Catching the Objects in boxes with Endoscope (Scenario-2)

Scenario-3: Clearing the Nose

The layout of scenario 3 can be seen in Figure 3.4. This scenario is built with a simulated anatomical nose model. The participant advances in a nose model to remove harmful substances in green colors. The camera is used like a tool at the same time as a fixed light that provides a clear view during operation inside the 3D nose model in the one-handed experiment (see Figure 3.4, A). The participant collects random green dots to clean the nose using a haptic device with one hand. The same simulation is applied with two haptic devices in the double-handed experiment (see Figure 3.4, B).

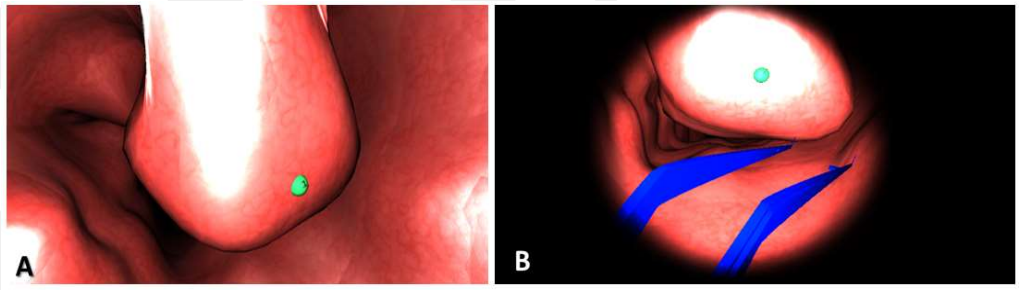


Figure 3.4 Clearing the Nose Scenario (Scenario-3)

The participant uses a camera as a light source while trying to collect harmful substances with a tool that is designed as a cautery model. There are ten tasks. The time, distance, catch time, and success attributes have been recorded for each experiment, as well as hand movement data.

Scenario-4: Following the Ball with an Endoscope

The layout of the Scenario 4 can be seen in Figure 3.5. The layout of scenario 4 can be seen in Figure 3.5. This scenario is built with a simulated anatomical nose model. The participant advances in the same nose model again to move a sphere object from a yellow node to a green node with a visible path line. In the single-handed version, the camera is used like a tool with a fixed light. In the double-handed version of the simulation, one hand is used to control the sphere object while the camera with a light source is controlled with another haptic device. The difficulty of this experimental setup is increased with sharp angles. This scenario contains fifteen tasks. The time, error time, total distance, error distance, deviation count and success attributes are recorded as well as hand movement data that contains geometric and time information per each frame.

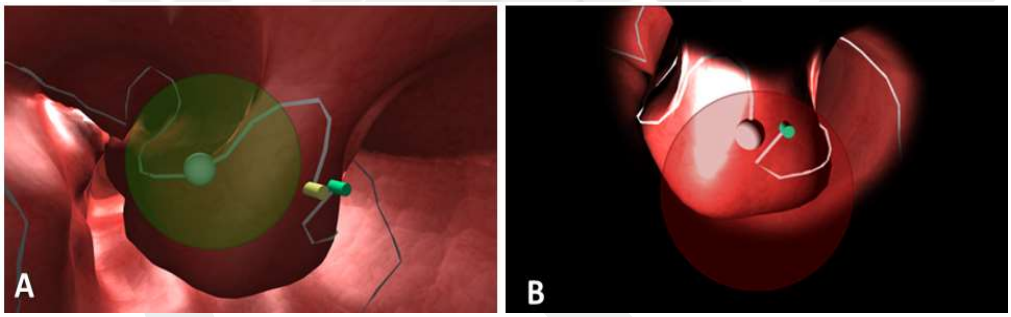


Figure 3.5 Following the Ball with an Endoscope (Scenario-4)

3.2 Experimental Data Collection

The data has been collected frame by frame during each experiment. Two different text files in JSON format were automatically saved in a specified folder. The name of the text files is combined together using different labels. Participant identification labels are stated as N01 that is the first participant, up to the final participant N28 (see Table 3.1, Participant). Second item in a name of the text file is order of scenario application (see Table 3.1, Scenario). Each experiment name is unique (see Table 3.1, Experiment). The name starts with “ECES” string that means ECE scenario. Then a code is added that references the four basic scenarios that are explained in the previous section (01, 02, 03, and 04). Finally last two number decodes the handedness condition

that is either 1 as single-handed scenario or 2 as double-handed. For instance, "ECES0102" means that the related scenario is 1 and "02" shows us that it is applied as a double-handed experiment (see Figure 3.2, B). Moreover, the file name contains "Date" information that holds the month and day information for the experimental application. In addition, "Handedness" column in Table 3.1 shows that which hand controls the virtual endoscopic tool in a related scenario (Dominant Hand or Non-dominant Hand). Finally, text file name contains the "Hand" word if it is a hand movement dataset. Therefore, all experiments contain two separate dataset files as Hand and Performance datasets (see Table 3.1, Hand and Performance columns). As a result, a sample experimental design output is saved in a text file named as "n01__07_ECES0302_14-9_D_HandData.txt" that has all necessary information to distinguish a specific experimental design. A performance data set is also existed in the same folder with the name "n01__07_ECES0302_14-9_D.txt".

Table 3.1 Sample Experimental Data Structure for Participant N01

Participant	Scenario	Experiment	Handedness	Hand	Performance
N01	02	ECES0101	D	+	+
N01	02	ECES0101	N	+	+
N01	03	ECES0301	D	+	+
N01	03	ECES0301	N	+	+
N01	04	ECES0201	D	+	+
N01	04	ECES0201	N	+	+
N01	05	ECES0401	D	+	+
N01	05	ECES0401	N	+	+
N01	06	ECES0102	D	+	+
N01	07	ECES0202	D	+	+
N01	08	ECES0302	D	+	+
N01	09	ECES0402	D	+	+

Each scenario prepared as a single-handed and double-handed versions as it is stated. In double handed version, participant plays the same scenarios with two haptic devices. In these experiments dominant hand controls the 3D endoscopic tool, while non-dominant hand uses camera and light source while it depends on the Handedness condition in single-handed versions. The data recording had been started as soon as scenario starts and finishes when the scenario completed in a predefined time period.

The 8 different one-handed experimental setups (see Table 3.1, first eight columns) have performance and hand data set files in a total 16 text files, while four double-handed experiments have 8 text file data sets (see Table 3.1, last four rows). The file and folder structure are organized and illustrated in Figure 3.6. Each participant experimental procedure has a folder and contains 24 different text files that provides experimental output which will be explained in the following chapters in detail.

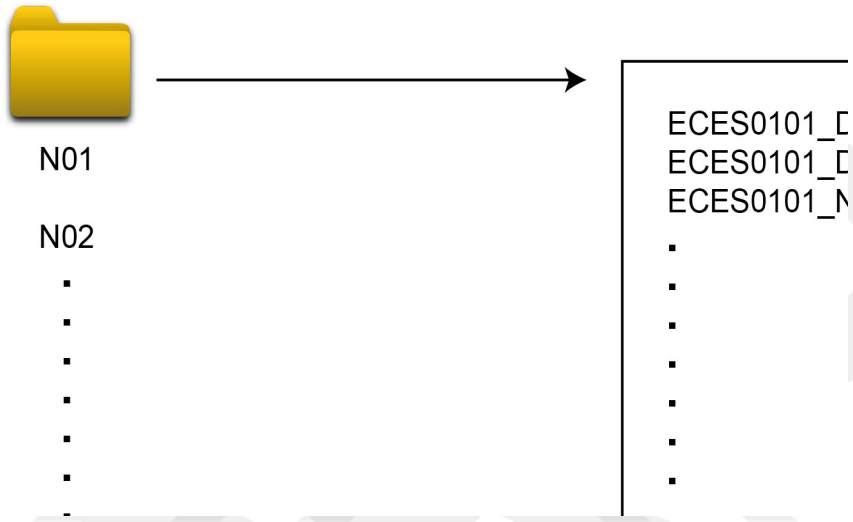


Figure 3.6 File and Folder Structure

To sum up, ECE has 12 different experiments designed based on 4 basic scenarios using haptic devices, a computer, and one computer screen. The output data is organized in JSON format and kept in text files. File names are unique and can be seen as categorical data. Finally, 24 dataset files per 28 participants provides 672 files in total. Each participant completed all experiments except “n08__02_ECES0101_16-9_N” and “N15__08_ECES0202 _18-9_D”. These missing datasets will be handled in the related section (see Section 5.1 Data Exploration).

3.3 Strategy and Framework

The first step towards to reach a strategic goal was to specifying a clear methodology in Figure 3.1. To create this methodological framework, background investigation, experimental application and experimental data collection steps are designed. Following items of the research procedure will be covered in the next sections step by

step. The aim of the framework is basically to take a set of dataset files, produce a meaningful insight and create a reproducible ML product to the top of it. Overall success of this research procedure can be measure by the accuracy and the applicability of the applied model.

The second step is to define the problem. Specifying the problem clearly will help us to explore the data and to select the right ML and data science tools. Selecting and deploying right toolbox in line with a clear boundary that is specified in the problem definition will be useful in the data modelling phase. This is the data exploration phase which will output a basic data model gathered from experimental studies. Further it will be used to integrate and match the right tools to reach the final goal.

The third phase is the goal settings. In this phase, the goals will be defined to reach a clear answer to define the success. For this reason, the data analysis process is the core part of this phase. The data set gathered and investigated in the previous section will be analyzed in detail to prepare data accordingly.

In the next phase, the ML model will be defined. This phase will be the learning phase from selection to implementation of a specific ML model. The data will be aligned accordingly to the selected ML model in each step. Therefore, the feature engineering activities will be the main part of this section. Finally, all the evaluation phase will be covered to provide the accuracy and applicability of the ML model as well as the methodological approximation of this research study.

CHAPTER 4

MODELLING AND PROBLEM SPECIFICATION

This Chapter contains three main activities called “Problem Specification”, “Developing Strategy & Setting up ML Model” and “Choosing the Right Toolbox”. The observations though literature is expected to serve as strong bases towards the accurate surgical skill level estimation using haptic generated hand datasets. Gaining a broader view of the current literature and understanding the limitations and methodologies of related approximations is the main expected outcome. The review of ECE and existing datasets was another important issue handled to build up the methodology. The experiments are available in four different scenarios developed in Unity3D game development environment as mentioned. They contain consistent hand data sets.

4.1 Problem Specification

The aim of building data science framework using a ML model is to classify the surgical skill using hand movements. The participant experience is known before the experiment (see Section 3.1.2 Participants). Demographic information like age, dominant hand, education and operating room performance (Monitored, Assisted or Performed) has been gathered prior to the experimental observations. The ultimate goal is to reach an acceptable level of accuracy using the ML model to predict the participant surgical skill level as intermediate or novice.

4.2 Developing Strategy & Setting up ML Model

A machine gain experience if it has learning abilities. A learner process is employed to generalize the data for this reason. This experience and generalization ability, gained from training data, later used to make new generalizations on new datasets. For instance, a supervised ML algorithm can simply be defined as a target function that maps input variables to an output variable. On the other side, an accurate feature

extraction process turns relevant raw data into a higher level of presentation completely by reducing the number of features and creates easy to manage new features without losing any necessary data. To sum up, these two data science approximations are complementary. For this reason, both methodologies are handled through one systematic study to develop a depth understanding as well as developing a modelling process. The existing literature of the ML methods and feature extraction studies is analyzed based on the attributes such as methods, datasets, purpose, and technology. Currently, it is observed that the surgical skill related studies are limited. However, it may be possible to reach new perspectives with the existence of hand related data science studies by using the relation between surgical skills and hand movements. To this end, the progress of current systematic direction includes wide area of interests like signal processing, image processing as well as hand gesture recognition systems, wearable gadgets or smart devices. Developing a ML model is an important process because it will be the background of the overall process through a set of ML or Feature. The problem statement in the previous section may also be revised in the sense of the adaptability of the ML or Feature Extraction (FE) model that will be selected. The data analysis process contains the data preparation, data preprocessing and analysis. The modelling section deals with what ML method will be applied and how it will be conducted. Therefore, the evaluation criteria are living part of the overall progress. Data analysis contains like reviewing, adapting, processing the data and applying it in a recursive way as long as necessary.

Modelling process contains selecting and applying proper ML algorithm according to the existing data structure. Therefore, it may also contain the adaptation, reprocessing of the data according to the selected approximation as well as feature selection, and evaluation process. Feature extraction is a side tool of the model to improve and tweak the setup to reach better results. Finally, the main goal is to reach optimal results and work on that results to improve the output continuously using the set of tools and technologies that will be mentioned in the following section.

4.3 Choosing the Right Toolbox

Today, the concepts of deep learning, artificial intelligence and ML are intertwined. However, ML is used extensively in scientific research with a wider scope. In this context, the number of studies involving statistical prediction efforts, pattern recognition capabilities or algorithms to perform a specific task is increasing day by day. In order to design a systematic ML application, we need to specify which tools we will use and why, along with the ML model that is identified as the fundamental building block (see Figure 4.1).

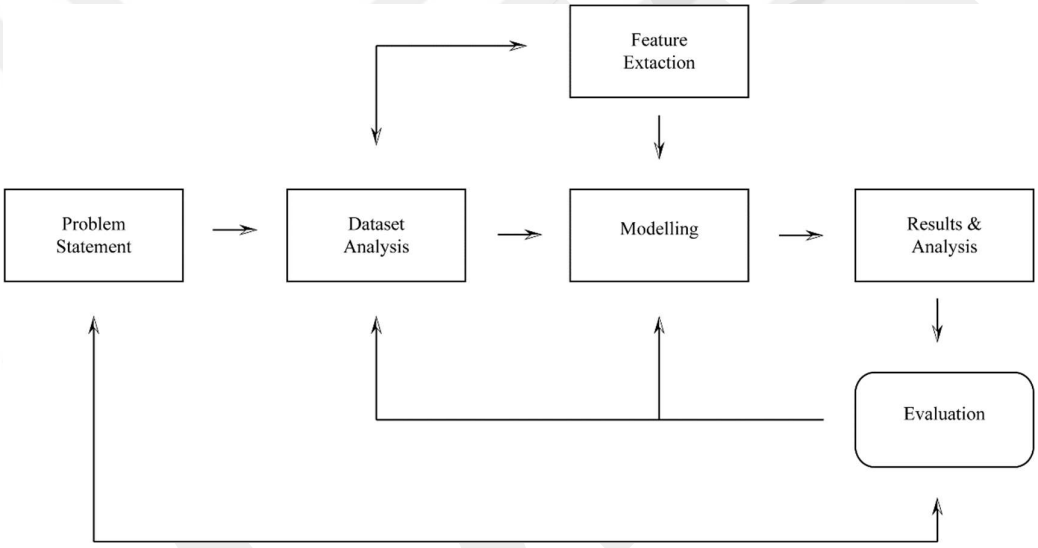


Figure 4.1 ML Model

Creating a framework to investigate data through a set of tools was a complicated process because there are large number of available tools. Python programming language is popular programming language in data science and “Conda” environment provides simple assistive technologies with it. They both has a short learning curve. There are software distributions like “Miniconda” which provides the set of tools and data analysis workbench built on top of Conda environment using python or R programming languages. In the technology exploratory phase, it is possible to come across some common names like pandas, matplotlib or Numpy. To choose the right tools the experience of the author of this thesis study is another important factor that

effects the selection process. Hence this section provides the details of the selected tools.

4.3.1 Python

There have been different programming languages around data science researches to apply ML approaches starting from the middle of twentieth century [84]. In this research study, the first target was to choose a programming language fits in the anticipated ML pipeline. Python programming language is a popular general-purpose high level programming language. It is an open-source programming language that has a large community who contribute to its libraries and tools for data science. The most active data science tools and environments are python based because of this the popularity. The libraries, such as NumPy, Pandas, and Scikit-learn, which provide a solid foundation for data science tasks makes python a perfect choice for an ML project. Moreover, it was anticipated that these powerful libraries alone may be enough to create the necessary foundation for this research. Python's simplicity, flexibility, and rich ecosystem makes it a popular choice for any data-oriented study.

Python as a high-level programming language focus on readability. It means that it is easy to learn. One way to overcome the performance tradeoffs of python is to somehow use faster programming languages. It still can use the power of system-level programming languages, when necessary, with this approach. The libraries like SciPy and NumPy uses such architectural perspectives. For instance, NumPy is a fundamental package when we need to process large volume of data. The magical word here is vectorization. NumPy package operations lives around an n-dimensional array contains same data types. Memory optimizations pushes this rule as the same data type variable holds same size in memory. In another words, there is no need python based built-in programming paradigms like exhaustive explicit loops or indexing methods unlike an NumPy's "ndarray object" oriented operations.

An optimized, pre-compiled C code handles such heavy works. The "ndarray" object has fixed size unlike dynamic python lists. Let x and y be two lists containing million numbers. In python, the multiplication of two lists, element by element is a very

expensive operation. However, “x * y” simply gives the same expected result with an optimized speed of C code with NumPy package behind the scenes. Therefore, developer has this ability without needing to leave the benefits of python language [105]. The broadcasting ability of NumPy package means that x and y can be in different shapes and user can still make arithmetic operations. The compatibility of the result is assured by an optimized code[106].

In a statically-typed language like C, C++, Fortran, variables are known at compile time. Therefore, the variable types should be specified by the developer. In a dynamically typed Python language, it is not a must to specify data types explicitly for variables or functions. This means that in statically typed languages, there is a code compilation where python checks data types during the runtime. Compile time type checking makes statically typed languages faster. Hence, one shortcoming for the Python language is the performance of Python code. However, there is a huge community effort to overcome such obstacles. For instance, Cython compiler which is an extension of Python is designed to allow developers speed up code executions when needed and there is no dependency directly to Cython C-compiler to run a Python software [107]. Another example is the CPython implementation (https://github.com/python/cpython). It is an entire implementation of Python language in C programming language. There is a need for interpreter to convert the CPython code into machine code unlike the C-extension Cython. The compiled Cython language allow developers to produce code that can directly be executed by the CPU so it is much faster. As a result, the rich integration capability of the Python language makes it a perfect asset for the data science community as well as this thesis study.

4.3.2 Technology Stack

The most significant challenge for a data science project is to manage the packages and their dependencies. The Anaconda python environment [108] helps to overcome such issues by providing a python package management system called Conda. In addition, Anaconda Navigator provides graphical user interface to manage and use

thousands of data science and ML packages, ready to use for different operating systems (Figure 4.2).

Anaconda Navigator environment contains Jupyter Notebook with ready to use thousands of libraries which clearly make the process easier across this ML project pipeline building effort. Jupyter Notebook is an open-source project [109] contains a rich set of open-source tools and provides a flexible environment that enable a rapid development infrastructure for a data science project. It is a web application that allows users to create and share document centric mathematical or statistical equations, different programming languages, equations, visualizations, and text. It has become a popular tool to create ML projects as well as many other applications such as data processing, statistical modeling, visualization due to its ease of use.

Figure 4.2 Anaconda Environment

The data handling phase was the first target after determining the environment, programming language and applications. Pandas is another open-source library [110] that provides data analysis and manipulation tools for Python programming language. In the center of Pandas library's approximation, there is a data frame object with indexing capability. It contains a dynamic integrated indexing structure that makes the data manipulation efficient. The Ability to read and write data in many different data formats such as csv, excel, SQL databases, lets users a fast implementation of raw

datasets. Once a dataset transformed into a data frame object, users can easily manipulate data to create a proper production pipeline. High performance and efficiency of pandas' library comes from optimizations written in Cython or pure C ([107]). Easy reding ability and smart data alignment with dynamic indexing enable users to manage and to manipulate huge datasets in a predefined structure, data frame. After creating a data frame, it is also possible to change the shape of data structure. For instance, a column can easily be removed or group engine makes data transformation simple. These and many other flexibilities of Pandas makes it popular across many different scientific domains.

Pandas library is built on top of NumPy [111]. Numpy is a scientific computing library for Python. It provides easy-to-use data structures and data analysis tools for handling different type of raw data such as tables, time-series, and more. In addition to Pandas' data frame structure, Series are another powerful data structure. Series is a one-dimensional labeled array capable of holding any data type (integers, strings, floating-point numbers, Python objects, etc.) where data frame is a two-dimensional data structure with columns. Pandas library is widely used in data science and analysis for tasks such as data cleaning, data exploration, data visualization, and data manipulation [110].

Another important item of the technology stack is Scikit-learn [112]. Scikit-learn contains large collection of ML algorithms. The interface provides a flexibility to conduct task-oriented scientific investigations. Task-oriented structure makes it easy to develop rapid specific applications. By leveraging the scientific Python ecosystem, it can seamlessly integrate into diverse applications beyond traditional statistical data analysis. The algorithms, implemented in a high-level language, can serve as fundamental components for developing customized solutions tailored to specific use cases [112].

CHAPTER 5

DATA ANALYSIS AND GOAL SETTINGS

Once the technology stack had been visible, the first target was clearly building a data science framework. This process starts with importing the whole data into a Pandas' data frame that was anticipated to be a reflection of the folder and file structure. A dictionary data structure was a proper first target for this purpose because it is very easy to build a data frame using a properly prepared dictionary. Therefore, the file and folder structure of the hand and performance datasets has been used as a guide.

```
obj.explorer['N01']
```

```
['n01__02_ECES0101_14-9_D.txt',
 'n01__02_ECES0101_14-9_D_HandData.txt',
 'n01__02_ECES0101_14-9_N.txt',
 'n01__02_ECES0101_14-9_N_HandData.txt',
 'n01__03_ECES0301_14-9_D.txt',
 'n01__03_ECES0301_14-9_D_HandData.txt',
 'n01__03_ECES0301_14-9_N.txt',
 'n01__03_ECES0301_14-9_N_HandData.txt',
 'n01__04_ECES0201_GGM_14-9_D.txt',
 'n01__04_ECES0201_GGM_14-9_D_HandData.txt',
 'n01__04_ECES0201_GGM_14-9_N.txt',
 'n01__04_ECES0201_GGM_14-9_N_HandData.txt',
 'n01__05_ECES0401_14-9_D.txt',
 'n01__05_ECES0401_14-9_D_HandData.txt',
 'n01__05_ECES0401_14-9_N.txt',
 'n01__05_ECES0401_14-9_N_HandData.txt',
 'n01__06_ECES0102_14-9_D.txt',
 'n01__06_ECES0102_14-9_D_HandData.txt',
 'n01__07_ECES0302_14-9_D.txt',
 'n01__07_ECES0302_14-9_D_HandData.txt',
 'n01__08_ECES0202 _GGM_14-9_D.txt',
 'n01__08_ECES0202 _GGM_14-9_D_HandData.txt',
 'n01__09_ECES0402_14-9_D.txt',
 'n01__09_ECES0402_14-9_D_HandData.txt']
```

Figure 5.1 Explorer view for Participant "N01"

A script called Data_Preprocess was used to build a dictionary which contains user IDs as keys and all txt file lists as values. Then calling the data preprocess object's get_explorer () function builds a dictionary which holds the set of experimental text file names per participants (see Figure 5.1). The explorer dictionary has 28 keys which is the total number of participants (N01, ..., N28). The length of the values of each

dictionary key contains 24 string values which is the total number of experimental data files in each participant's folder.

5.1 Data Exploration

The data exploration is a must before the data analysis. All datasets are systematically organized and looks like coherent. However, the performance datasets should be handled carefully because only data features related with hand gestures will be used to align with the goal of this research study. All datasets have some characteristics in common like CamPosition feature. For instance, the experiment starts with a datapoint recorded as 0.014582 second in a hand dataset (see Figure 5.2).

	DataTime	ToolPosition	ToolRotationEuler	ToolRotationQuaternion	CamPosition	CamRotationQuaternion	CamRotationEuler	CurrentDate
0	0.014582	(0.2, 0.2, -33.5)	(341.8, 349.6, 347.0)	(-0.1, -0.1, -0.1, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	9/14/2015 1:29:00 PM
1	0.028444	(0.2, 1.5, -34.4)	(341.4, 349.6, 347.0)	(-0.1, -0.1, -0.1, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	9/14/2015 1:29:00 PM
2	0.031147	(0.5, 2.7, -35.4)	(340.9, 349.5, 347.0)	(-0.2, -0.1, -0.1, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	9/14/2015 1:29:00 PM
3	0.040464	(0.5, 3.4, -35.9)	(340.7, 349.5, 347.0)	(-0.2, -0.1, -0.1, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	9/14/2015 1:29:00 PM
4	0.050014	(0.5, 4.1, -36.6)	(340.5, 349.5, 347.0)	(-0.2, -0.1, -0.1, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	9/14/2015 1:29:00 PM
...	...	...	...	...	...	...	...	...
74306	24.185385	(-15.0, -44.5, -20.5)	(348.8, 346.4, 343.3)	(-0.1, -0.1, -0.2, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	10/7/2015 5:41:29 PM
74307	24.187498	(-14.5, -44.4, -17.8)	(349.7, 346.7, 343.3)	(-0.1, -0.1, -0.2, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	10/7/2015 5:41:29 PM
74308	24.235850	(-14.5, -44.5, -17.7)	(349.7, 346.6, 343.3)	(-0.1, -0.1, -0.2, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	10/7/2015 5:41:29 PM
74309	24.237791	(-14.5, -44.2, -16.0)	(350.2, 346.9, 343.4)	(-0.1, -0.1, -0.2, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	10/7/2015 5:41:29 PM
74310	24.285671	(-14.5, -44.3, -16.0)	(350.2, 346.9, 343.4)	(-0.1, -0.1, -0.2, 1.0)	(0.9, 43.6, -131.0)	(0.2, 0.0, 0.0, 1.0)	(20.0, 0.0, 0.0)	10/7/2015 5:41:29 PM

74311 rows × 8 columns

Figure 5.2 Hand Data Preview of ECES0101

The recorded time value has no fixed time intervals. Instead, it is recorded frame by frame depending on change in the scene during a task in a 3D environment. This recorded value is called DateTime (see Figure 5.2). Two positional and three rotational data records follow it, which are basically presented as coordinate systems in two different formats; Quaternion and Euler Angles. Finally, there is a "Current Date" record that holds the exact time when the experiment conducted up to the seconds.

However, experiments have differences based on their scenario development. Accordingly, their output differs. For instance, first two row of Table 5.1 shows experiments that have only tool position data. It means that the camera and the light in the related scene are fixed.

The next six experiments have no positional data features for tool (see Table 5.1, TP, TRE and TRQ). This situation shows that the camera and the light is integrated to the endoscopic tool in the related scene. Finally, last four rows show double-handed experiments that contain all positional data features meaning that position of tool and camera is saved frame by frame.

Table 5.1 Hand Dataset Features for All Experiments

Experiment	F#	DT	TP	TRE	TRQ	CP	CRQ	CRE	CurD
ECES0101_D	*	*	*	*	*	F	F	F	*
ECES0101_N	*	*	*	*	*	F	F	F	*
ECES0301_D	*	*				*	*	*	*
ECES0301_N	*	*				*	*	*	*
ECES0201_D	*	*				*	*	*	*
ECES0201_N	*	*				*	*	*	*
ECES0401_D	*	*				*	*	*	*
ECES0401_N	*	*				*	*	*	*
ECES0102	*	*	*	*	*	*	*	*	*
ECES0202	*	*	*	*	*	*	*	*	*
ECES0302	*	*	*	*	*	*	*	*	*
ECES0402	*	*	*	*	*	*	*	*	*

F#: Frame number
DT: Data Time
TP: Tool Position
TRE: Tool Rotation Euler
TRQ: Tool Rotation Quaternion
CP: Cam Position
CRQ: Cam Rotation Quaternion
CRE: Cam Rotation Euler
CurD: Current Date

The performance datasets also in the similar situation. This time there are data features that holds similar values. For instance, time and total time measures the same value for two different experiments. The data exploration shows that there is an experimental impact. To match up the datasets with the possible ML algorithms, this issue will be investigated in the following sections.

5.1.1 Missing Files

Two missing file is detected while investigating the datasets. These files are "n08__02_ECES0101_16-9_N" and "N15__08_ECES0202 _18-9_D.txt". Hence, the missing files will cause to remove the experimental data of "ECES0101_N" and "ECES0202" completely in data processing. The experiment count is reduced to 10.

Table 5.2 Performance Dataset Features for All Experiments

Experiment	T	Dis	CamD	CT	S	CrD	ErrT	ErrD	Dct	LD
ECES0101_D	*	*			*	*				
ECES0101_N	*	*			*	*				
ECES0301_D	*	*	Dis		*	*				
ECES0301_N	*	*	Dis		*	*				
ECES0201_D	*	*	*	*	*	*				
ECES0201_N	*	*	*	*	*	*				
ECES0401_D	**	**			*	*	*	*	*	
ECES0401_N	**	**			*	*	*	*	*	
ECES0102	*	*	*		*	*				
ECES0202	*	*	*		*	*				
ECES0302	*	*	Dis		*	*				
ECES0402	*	*	*		*	*	*	*	*	*

T *: Time
T:** Total Time
Dis*: Distance
CT: Catch Time
ErrD: Error Distance
Dct: Deviation Count
Dis:** Total Distance (Equals to **Dis*:** variable)
CamD: Camera Distance
Dis: equal to Dis* data
CrD: Current Time
ErrT: Error Time
LD: Light Distance

5.2 Goal Settings

As the data gets complex with huge volumes, different kind of statistical perspectives are being visible as well as ML models. For instance, a linear regression approximation is said to be more than enough if there is a descriptive purpose. However, some Neural Networks do the similar job with a more complicated model. The quality is measurable depending of the forecasting success through a Neural Network where linear regression says here the equation you can use it to forecast. Forecasting and prediction have very close meaning in dictionary. However, forecasting has different meaning with a series of datasets that provide different observations through time. A prediction is about estimating what will be the value for a given input. Time is not involved in the prediction practice typically. Therefore, the term, "sequence prediction", is used to match up with forecasting term when data with temporal nature is available. Sequence prediction is trying to estimate target values in a specific time intervals if a given set of dependent variables is provided. In another words, it comes into play when time series observations are available. Temporal nature of datasets brings two data classification groups into the table. Firstly, the Cross-Sectional Data is used when the

dataset is not sequential and contains independent data points. Regression models, Random Forest algorithm or NN can be applied for different cross sectional data problems.

Another distinct classification schema is time series data groups. Time series as a mathematical term, is data points indexed in ideally equal time order. This regular time intervals have a collection of observations for a certain variable. These observations are dependent, cannot be mixed and have no missing data to keep order. However, datasets are generally containing both cross sectional and time series components at the same time. For instance, a frame/image can be observed for a given time in a video. For a given time, many other frame/images can be observed without dependency to each other. This view of the data is a sample for cross sectional data groups. In the same video a lot of frame/image follows each other in regular time sequence which puts the problem in the time series data classification group's boundaries for the same video. Dataset structures usually contain suitable characteristics for both of classification approximations and they can be used in combination.

In three dimensional spaces there are three coordinate planes so an ordered triple (x, y, and z) is used to represent a point. A simple internet search brings right hand rule to understand orientation of three-dimension space. It is common to find direction of magnetic fields, rotation, spirals etc. For instance, simply curl your hands around z axis with thumb out stretch. The thumb shows positive z axis if your hand moves in counterclockwise fashion and curls moves direct to the positive x and y axis area. Such conventional approximations make it easy to understand the orientation in 3D space. For instance, drawing a box in 3D space and finding the distance of opposite corners of the box which are specified as ordered triple coordinates. It is possible to find the distance by applying Pythagorean theorem or just adding z plane to a 2D distance formula.

Euler angles are basic geometric representation of rotations. Three angles are used to describe orientation with respect to the fixed coordinate system. Extrinsic rotation means one axes is fixed so rotations are presented as three rotations. Intrinsic rotations

mean that the coordinate system body rotates as well as rotation of three axes. There are twelve possible rotations (z-y-x, y-x-z…z-x-z, x-y-x) even without considering that if the rotation is extrinsic or intrinsic. This moves Euler angles to a point that to use it in 3D computational environment is exhaustive. Moreover, Euler angle has a gimble lock issue which is a specific situation that two axes are locked in parallel that causes loss of freedom of one degree in three dimensional spaces. Locked axes still can rotate together with respect to the coordinate space however it generates a loss of freedom for one axis. The Euler angles had been provided the foundations 3D rotations. However, Hamilton's quaternion is reinvented with its efficiency, intuitive nature. Quaternion was introduced by the mathematician Hamilton [113]. It adds functionality to complex numbers using four-dimensional system. Therefore, many graphical engines use quaternion arithmetic behind the scenes to operate accurately in 3D space including unity3D. For instance, a calculation of rotation can be done using less parameter in quaternion notation. Combination of rotational possibility for each axis adds more complex layer to Euler angle rotation calculations which needs a proper solution to each rotation case. Such issues direct graphical engines to the Hamilton's quaternion notation. Quaternion denoted by below notation:

$$Q = w + xi + yj + zk$$

W is the real part of the equation. The i, j, k and x, y, z are imaginary section which can be expressed as an imaginary vector. It means that the quaternion can be expressed as a q= w + v. The below equations are the basics of quaternion algebraic body:

$$i2 = j2 = k2 = ijk = -1$$
$$ij = k = -ji$$
$$jk = i = -kj$$
$$ki = j = -ik$$

The Unity uses Euler angles in visual representation to show the rotations in degrees as three values X, Y, Z under transform rotation. However, to access these three angles as a vector3 variable, unity documentation says that it returns the rotational vector in

a different order: Z, X, Y. Each represents the degree of rotation about a fixed axis with respect to the parent object. This is an extrinsic rotation situation previously discussed since the rotations use fixed axes. Any try to reach this vector3 variable return three axes which is a description of orientation of an object with respect to the parent coordinate body system.

The environmental impact has been reviewed in this section and should be revisited to reveal the different potential of the hand movement dataset. However, the attempts to measure the 3D movement outside of the graphical engine is hard and the results has not been satisfactory. It is possible to measure many 3D orientation changes in unity3D environment with a short piece of code using the recorded coordinate values. Therefore, it is decided to use methods provided by graphical engine for this purpose when necessary.

To sum up, the datasets have a temporal structure and somehow fits in the definitions of time series in this research study. First investigations show that the short time frames like 0.014 second is not enough to create powerful time series observations. Creating more distinct time frames like time period per second reduces the volume of the data dramatically. Moreover, low volume of the data may limit the accuracy of many deep learning approximations. As a result, the data should be organized as cross-sectional datasets if it is possible. The hand movement datasets need to be analyzed carefully to build a proper understanding of movements in three dimensional spaces and to remove time dependency.

5.2.1 Success Criteria

This research study deals with a classification problem. The aim is specified as to predict a skill level of a participant based on the participants' hand movement related data using some proper ML algorithms. These inferences show that there should be a classifier algorithm. However, how can we define the success clearly? The evaluation of a ML approximation's success is an endless process. Therefore, a definitive goal has to be specified prior to model the processing of ML model. Let remember what is known up to this point. The problem specification phase (see Section 3.3 Problem

Specification) introduced the core definition of the problem. The data exploration phase (see Section 5.2 Data Exploration) provided us insights about the data. The previous section (see Section 5.3 Goal Settings) added more information to that observation to further look at the data in statistical perspective.

Now, it is possible to say that what data is available is known and how these data sets can be aligned with the goal of this thesis. The hand datasets and performance datasets contain structured, labeled information. To derive a success criterion, the goal settings show that it is possible to use more ML approximations if it is organized as cross-sectional datasets. Moreover, these same datasets have been used to predict a group of participants successfully with an accuracy of 84.6% [69]. As a result, the ML model's success can be defined as the ability to predict a group of participants' surgical skill level with an accuracy higher than 84%, using strictly hand movement-based information.

5.3 Data Preprocessing

The data analysis had been started with data exploration (see Section 5.1 Data Exploration). A file explorer dictionary was built that holds a file list of each participant (see Figure 5.1). A parent class called "MainDataFrame" has been defined with a child class called "FrameBuilder" for this purpose. A main class object keeps two values created by frame builder child class, a full experimental data which hold whole datasets and a user experience dictionary which holds the predefined experience levels. A unique short experiment name is constructed for each experiment. For instance, the file name "n01__02_ECES0101_14-9_D_HandData" turned into "ECES0101_D_HandData" as a key value of dictionary. Similarly, "ECES0101_D" is a sample performance data set of an experiment.

The dictionary holds 10 hand text files and 10 performance text files for each participant. In total, 280 text files available as hand dataset. FrameBuilder makes the any necessary changes to build an all-in-one dictionary for each experiment using the explorer script. This will be handled with a following call:

TargetDataFrame = FrameBuilder (path)

After running the above code, whole data set are ready for processing. For instance, a full performance dataset is a target dictionary value for each experiment and can be reached like below:

TargetDataFrame['ECES0101_D']

Each data point means a task in a performance dataset (see Figure 5.4). Eight experiments have ten tasks for each scenario design (80) except the 4th scenario that includes 15 tasks. Therefore, the experiments ECES0401 and ECES0402 have 15 datapoints per participant (30 total tasks). The "time" values are available in all performance datasets. It shows that how much time it takes to complete a specific task.

```
['ECES0101_D',
 'ECES0101_D_HandData',
 'ECES0301_D',
 'ECES0301_D_HandData',
 'ECES0301_N',
 'ECES0301_N_HandData',
 'ECES0201_D',
 'ECES0201_D_HandData',
 'ECES0201_N',
 'ECES0201_N_HandData',
 'ECES0401_D',
 'ECES0401_D_HandData',
 'ECES0401_N',
 'ECES0401_N_HandData',
 'ECES0102_D',
 'ECES0102_D_HandData',
 'ECES0302_D',
 'ECES0302_D_HandData',
 'ECES0402_D',
 'ECES0402_D_HandData']
```

Figure 5.3 Key Values of Full Dictionary

The "distance" feature shows the path length that a user hands go through with the endoscopic tool for each task. In the experimental performance datasets, these features are available with exact the same meaning (see Table 5.1). Whence, the other features will be removed and each file will be used create a unified dataset. The complete dataset will be used as a base to extract new features for the hand movement data.

	time	distance	CameraDistance	isSucceed	CurrentDate
0	2.080288	151.962600	0.0	True	2015-09-14 13:29:02
1	2.782617	369.951800	0.0	True	2015-09-14 13:29:05
2	2.498920	245.759811	0.0	True	2015-09-14 13:29:07
3	2.639439	270.080719	0.0	True	2015-09-14 13:29:10
4	2.077340	205.663500	0.0	True	2015-09-14 13:29:12
...	...	...	...	...	...
275	3.483709	303.515800	0.0	True	2015-10-07 17:41:19
276	2.183658	198.647934	0.0	True	2015-10-07 17:41:21
277	2.015665	200.114777	0.0	True	2015-10-07 17:41:23
278	2.333885	209.127930	0.0	True	2015-10-07 17:41:26
279	3.064451	240.511887	0.0	True	2015-10-07 17:41:29

280 rows × 7 columns

Figure 5.4 Full Performance Data Frame for “ECES0101_D”

Hand movement dataset have datapoints saved frame by frame. It means that a C# script is fired with a change in hand movement and saves the related information. For instance, DataTime feature (see Figure 5.5) shows the elapsed time from the beginning of the experiment to the related datapoint.

Positional features are however used to record the exact position of an endoscopic tool, camera or light when the data is saved. The positional data is not applicable for now because it needs an implementation in unity environment which will be mentioned in the following sections. Therefore, positional data will be removed from the data frame.

Time dependency will also be removed as we mentioned previously (see Section 5.2 Goal Settings). It means that there is only one data feature and it is DataTime. The positional data will be handled separately in feature engineering process.

	DataTime	ToolPosition	ToolRotationEuler	ToolRotationQuaternion	CamPosition	CamRotationQuaternion	CamRotationEuler	CurrentDate
0	0.100657	(-0.1, 12.4, -42.1)	(351.9, 0.1, 323.7)	(-0.1, 0.0, -0.3, 0.9)	(1.0, 43.5, -130.9)	(-0.1, 0.0, 0.0, 1.0)	(351.2, 1.1, 355.2)	2015-09-14 13:44:26
1	0.200162	(-1.1, 18.2, -45.8)	(348.8, 359.7, 324.0)	(-0.1, 0.0, -0.3, 0.9)	(1.0, 43.5, -130.9)	(-0.1, 0.0, 0.0, 1.0)	(351.3, 1.1, 355.2)	2015-09-14 13:44:27
2	0.300273	(-2.8, 17.6, -45.5)	(348.3, 356.8, 324.3)	(-0.1, -0.1, -0.3, 0.9)	(1.0, 43.5, -130.9)	(-0.1, 0.0, 0.0, 1.0)	(351.3, 1.1, 355.2)	2015-09-14 13:44:27
3	0.400473	(-4.0, 15.0, -43.2)	(349.1, 356.6, 324.5)	(-0.1, -0.1, -0.3, 0.9)	(1.5, 43.5, -130.9)	(-0.1, 0.0, 0.0, 1.0)	(351.2, 1.2, 355.2)	2015-09-14 13:44:27
4	0.500921	(-8.9, 7.4, -29.1)	(352.9, 357.9, 324.5)	(-0.1, 0.0, -0.3, 1.0)	(4.4, 42.8, -130.1)	(-0.1, 0.0, 0.0, 1.0)	(351.1, 0.9, 354.7)	2015-09-14 13:44:27
...	...	...	...	...	...	...	...	...
10628	43.732640	(-8.1, -27.1, -33.3)	(343.6, 9.4, 339.7)	(-0.2, 0.1, -0.2, 1.0)	(-41.1, 17.8, -120.5)	(0.0, 0.1, -0.1, 1.0)	(359.4, 7.3, 350.3)	2015-10-07 17:49:47
10629	43.833225	(-9.2, -29.4, -31.1)	(343.0, 9.4, 339.7)	(-0.2, 0.1, -0.2, 1.0)	(-41.1, 18.0, -120.7)	(0.0, 0.1, -0.1, 1.0)	(359.5, 7.3, 350.5)	2015-10-07 17:49:47
10630	43.934150	(-11.3, -33.0, -27.6)	(342.3, 8.9, 339.6)	(-0.2, 0.0, -0.2, 1.0)	(-41.1, 18.1, -120.7)	(0.0, 0.1, -0.1, 1.0)	(359.5, 7.3, 350.5)	2015-10-07 17:49:47
10631	44.032810	(-11.5, -35.4, -21.5)	(342.5, 9.4, 339.6)	(-0.2, 0.1, -0.2, 1.0)	(-40.7, 17.8, -120.5)	(0.0, 0.1, -0.1, 1.0)	(359.5, 7.4, 350.6)	2015-10-07 17:49:47
10632	44.132970	(-11.5, -37.3, -16.1)	(344.4, 9.7, 339.6)	(-0.1, 0.1, -0.2, 1.0)	(-40.7, 17.5, -120.4)	(0.0, 0.1, -0.1, 1.0)	(359.4, 7.3, 350.6)	2015-10-07 17:49:47

Figure 5.5 Full Hand Data Frame for "ECES0101_D_Hand"

5.3.1 Creating Positional Datasets

The collected positional dataset is not applicable as is (see Section 5.2 Goal Settings). The positional features of the hand movement dataset should be turned into a proper format.

Table 5.3 Calculated Positional Data Features

Experiment	F#	DT	AVx	AVy	AVz	AAx	AAy	AAz	RV
ECES0101_D	*	*	*	*	*	*	*	*	*
ECES0101_N	*	*	*	*	*	*	*	*	*
ECES0301_D	*	*	C*	C*	C*	C*	C*	C*	C*
ECES0301_N	*	*	C*	C*	C*	C*	C*	C*	C*
ECES0201_D	*	*	C*	C*	C*	C*	C*	C*	C*
ECES0201_N	*	*	C*	C*	C*	C*	C*	C*	C*
ECES0401_D	*	*	C*	C*	C*	C*	C*	C*	C*
ECES0401_N	*	*	C*	C*	C*	C*	C*	C*	C*
ECES0102	*	*	D*	D*	D*	D*	D*	D*	D*
ECES0202	*	*	D*	D*	D*	D*	D*	D*	D*
ECES0302	*	*	D*	D*	D*	D*	D*	D*	D*
ECES0402	*	*	D*	D*	D*	D*	D*	D*	D*

F#: Frame number
DT: DataTime
AVx, AVy, AVz: Angular Velocity for each coordinate
AAx, AAy, AAz: Angular Acceleration for each coordinate
RV: Rotational Velocity
***:** Tool position data (fixed cam, fixed light)
C*: Cam position data (Cam as a tool)
D*: Calculation is conducted two times (Double hand)

For this purpose, a small Unity project is designed to re-implement the raw positional hand data points. Using the change in positional coordinates depending on data time,

it is possible to create features like angular velocity. Moreover, it is possible to derive more features using the angular velocity.

	0	1	2	3	4	5	6	7	8	9
0	Frame	0	0.014582	1.343041e+06	1.373690e+06	1363473.0	92105100.0	94206980.0	93506350.0	2.355810e+06
1	Frame	1	0.028444	-1.653294e+03	0.000000e+00	0.0	-97005740.0	-99097470.0	-98360470.0	1.653294e+03
2	Frame	2	0.031147	-1.059633e+04	-2.119395e+03	0.0	-500685400.0	-508886400.0	-504323600.0	1.080620e+04
3	Frame	3	0.040464	-1.229823e+03	0.000000e+00	0.0	-144283500.0	-147441100.0	-146344500.0	1.229823e+03
4	Frame	4	0.050014	-1.199958e+03	0.000000e+00	0.0	-140755000.0	-143838600.0	-142768900.0	1.199958e+03
...	...	...	...	...	...	...	...	...	...	...
2246	Frame	2246	22.470760	-9.170123e+02	0.000000e+00	0.0	-215085700.0	-219844000.0	-218209000.0	9.170123e+02
2247	Frame	2247	22.480300	-1.201229e+03	-6.007064e+02	0.0	-140925800.0	-144076000.0	-142942000.0	1.343056e+03
2248	Frame	2248	22.492180	-9.640989e+02	4.820495e+02	0.0	-113068900.0	-115525700.0	-114706800.0	1.077895e+03
2249	Frame	2249	22.500820	-1.990608e+03	0.000000e+00	0.0	-155772900.0	-159091900.0	-157908700.0	1.990608e+03
2250	Frame	2250	22.510730	0.000000e+00	1.156149e+03	0.0	-135515800.0	-138491700.0	-137577500.0	1.156149e+03

2251 rows × 16 columns

Figure 5.6 Positional Data Preview

Angular velocity is a measure that can be calculated using the difference in rotation angles between two frames, as well as the time difference between those frames. A sample Unity C# script is used to import the text file with the formatted as JSON. To calculate the angular velocity the toolRotationEuler values is enough. Quaternion calculations are handled by Unity engine behind the scenes (see Section 5.2 Goal Settings) so there is no need to import them. The Rotation Euler values prepared as a list that contains the Tool Rotational Euler coordinates for each frame. The DataTime feature can be used to reach the time difference between each consecutive value. A function iterates through the list of Rotational Euler values and calculates the necessary data. The calculation of angular velocity (AV), angular acceleration (AA) and rotational velocity (RV) between each frame contains a couple of steps:

- Calculate the time difference between each frame using DataTime:

  deltaTime = current time – previous time
- Calculate the change in rotation angle between each frame using Rotational Euler values:

  deltaRotation = current rotation – previous rotation
- Calculate AV between each frame in degrees per second:

$$AV = deltaRotation / deltaTime$$

- Calculate AA in degrees per second squared:

$$AA = (Current\ AV - Previous\ AV) / deltaTime$$

- Calculate RV:

$$RV = AV.magnitude$$

AV and AA calculations results in three values for each coordinate as AVx, Avy and AVz (see Table 5.2). A data preview is prepared from the output of positional dataset (see Figure 5.6). These datasets do not need a feature engineering activity. The data is complete and organized as is except same missing experimental files (see Section 5.1.1 Missing Files) so they will be extracted from the final calculations. It is important to notice that the positional datasets will be handled in a separate section.

ECES0101_D_HandData

	DataTime
0	0.014582
1	0.028444
2	0.031147
3	0.040464
4	0.050014
...	...
74306	24.185385
74307	24.187498
74308	24.235850
74309	24.237791
74310	24.285671

74311 rows × 1 columns

ECES0101_D

	time	distance
0	2.080288	151.962600
1	2.782617	369.951800
2	2.498920	245.759811
3	2.639439	270.080719
4	2.077340	205.663500
...	...	...
275	3.483709	303.515800
276	2.183658	198.647934
277	2.015665	200.114777
278	2.333885	209.127930
279	3.064451	240.511887

280 rows × 2 columns

Figure 5.7 Hand and Performance Data Preview for ECES0101

5.3.2 Data Alignment

In this section, hand datasets have been handled in a way to accommodate to the goal of turning time series data into a supervised learning model. All experimental data shows similar characteristics by means of 3D scenario design. Each collected hand datasets have DataTime feature for the hand movement data that controls a virtual

endoscopic tool. It gives us the ability to align the data through experimental model by means of scenario design. The experiment ECES0101 and ECES0402 are very different experiments by design. In ECES0101, the participant uses dominant or non-dominant hand to control an endoscopic tool in a room. In ECES0402, participant uses one hand to control the cautery tool while other hand controls the camera as a light source in a nose model. However, the hand movement and performance datasets are conjugate.

ECES0402_D_HandData

	DataTime
0	0.101053
1	0.434386
2	0.439350
3	0.442235
4	0.500160
...	...
21752	99.446830
21753	99.546640
21754	99.647920
21755	99.746350
21756	99.800520

21757 rows × 1 columns

ECES0402_D

	TotalTime	TotalDistance
0	3.678089	72.555470
1	2.801212	30.102919
2	5.257673	64.238930
3	1.172193	11.524234
4	2.559716	24.838265
...	...	...
415	14.181090	95.114815
416	8.327601	45.838303
417	13.395160	67.029000
418	11.777038	83.089090
419	4.581527	42.461258

420 rows × 2 columns

Figure 5.8 Hand and Performance Data Preview for ECES0402

Both hand datasets show the changes in time while surgical tool moves in a virtual model during the scenario play (see Figure 5.7 and Figure 5.8, HandData). The performance dataset of ECES0402 has a TotalTime (see Figure 5.8) attribute that holds the exact the same information of ECES0101's time attribute (see Figure 5.7, Performance Data), total completion time of a task.

The data alignment gives us also the ability to combine different datasets as a one complete dataset if necessary. It is now possible to append the ECES402 datasets to

ECES0101 dataset with a categorical feature experiment name or id. This approximation helps us to increase the probability of more ML algorithms applicability as well as eliminating the time dependency.

5.4 Data Analysis

The data analysis has been started with this chapter and it will continue with the Feature Engineering activities because the data is still not complete. Next chapter will introduce many features to improve the dataset more. However, the TimeElapsed feature is added to the data frame for demonstration purposes. The below graphs (see Figure 5.9) show elapsed time for each data point for all participants per experiment. It is important to notice that long sticks shows that participant' concentrated on a task that is hard and takes time if it is not a problematic record. It was anticipated that the difficulty of scenario is low in ECES0101 and high in ECES0402. The results clearly support this opinion.

In data preparation phase, it may be possible to clean the outliers to improve the model quality. However, to decide on the outlier is hard with hand movements. It is observed that sometimes participant tries hard to reach a target in a limited time frame. Also, they may lose the feeling of depth perception while moving through the virtual model. Therefore, the spikes in graph are evaluated as a normal procedural side effects and any data cleaning for outlier data is not conducted.

To conclude, the datasets has been investigated with clearly specified goals in this section. In any ML application, it is important to match up the ML algorithm with the data. For this reason, the success criteria were defined, data was prepared, data alignment, data preprocessing, and data exploration were handled.

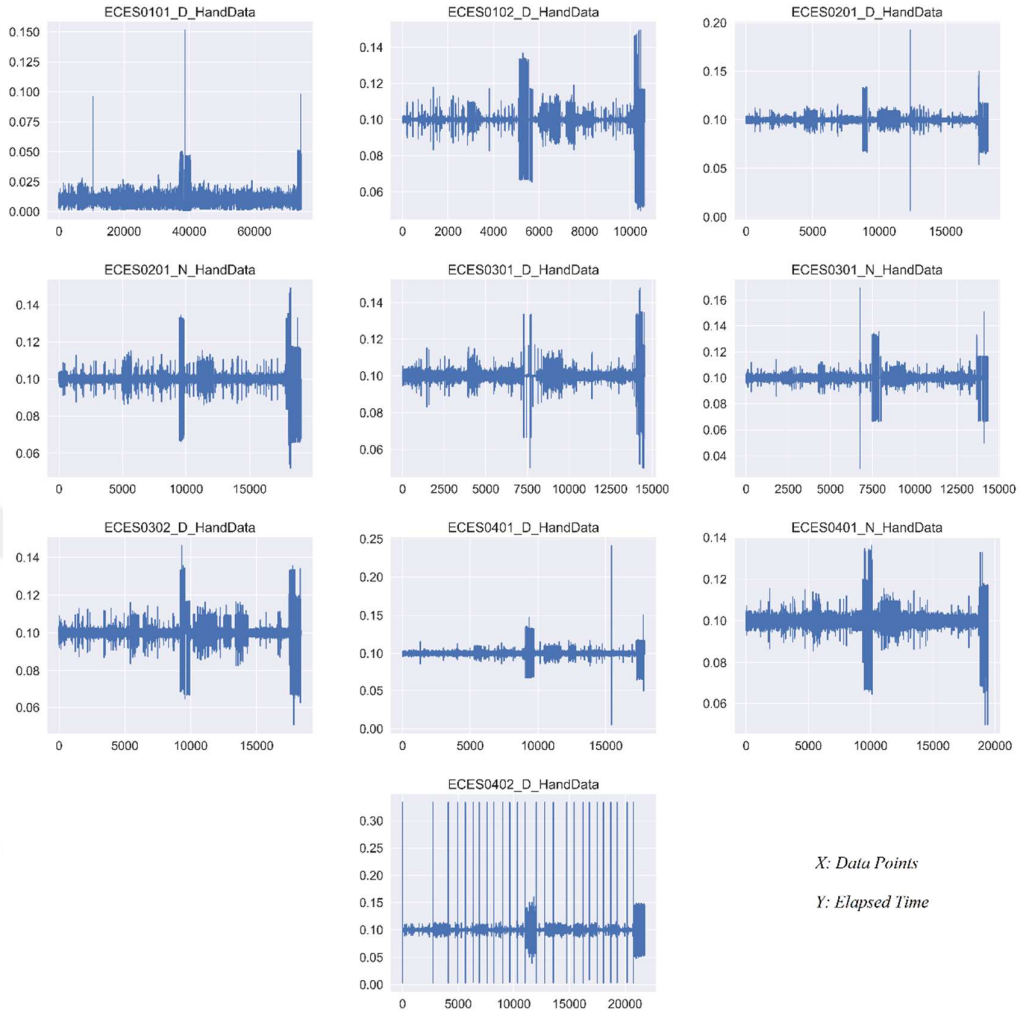


Figure 5.9 “TimeElapsed” Feature per Data Point for All Experiments

CHAPTER 6

FEATURE ENGINEERING AND ML APPLICATION

In this chapter, the aim is to finalize and to create richer feature set so that the probability of ML model's success is increased. Moreover, selected ML algorithm will be applied with these finalized datasets. Therefore, there are two main activities in this chapter, data analysis and the application of ML model.

It may be said that the feature extraction has already been started in previous section because it is possible to observe some common data attributes in the existing datasets. In addition, it is realized that some categorical data features can be extracted just using the name of the hand dataset files. For example, "n01__02_ECES0101_14-9_D_HandData" contains the participant identification code "N01" and handedness case; "D". Therefore, we know that this dataset belongs to the participant N01 who plays the experimental scenario with dominant hand. Such new features are added to both performance and hand datasets with other known categorical features like a distinct experiment name or the surgical skill level of participant.

It was stated previously that the performance data can help us to extract some more features (see Figure 5.4). The DataTime that is common feature in all hand datasets (see Figure 5.5) shows how much time it took up to the current datapoint. Hence, this similarity can be used to calculate a feature that holds the elapsed time for each datapoint to complete. TimeElapsed is the first feature that can help the idea of extracting more features.

The time in performance datasets shows how much it takes to complete a task. Now it is possible to go through all datasets by summing each DataTime value up and check if the performance task completion time of is reached or not. The TaskID feature has been added to the hand datasets using this approximation. However, it is detected that

there are idle times that do not belong to any task. These idle times detected during task changes as well as the beginning and the end of the experiments. Idle tasks are labeled as “0” (see Figure 6.1, TaskID). The idle time values will be not be used and removed for this research study.

Participant	Handedness	Experiment	Skill	TimeElapsed	TaskID	distance
N01	D	ECES0101_D_HandData	1	0.014582	1.0	0.734119
N01	D	ECES0101_D_HandData	1	0.013862	1.0	0.734119
N01	D	ECES0101_D_HandData	1	0.002704	1.0	0.734119
N01	D	ECES0101_D_HandData	1	0.009317	1.0	0.734119
N01	D	ECES0101_D_HandData	1	0.009550	1.0	0.734119
...	...	...	...	...	...	...
N28	D	ECES0101_D_HandData	0	0.047724	0.0	0.000000
N28	D	ECES0101_D_HandData	0	0.002113	0.0	0.000000
N28	D	ECES0101_D_HandData	0	0.048352	0.0	0.000000
N28	D	ECES0101_D_HandData	0	0.001941	0.0	0.000000
N28	D	ECES0101_D_HandData	0	0.047880	0.0	0.000000

Figure 6.1 Preview for Extracted Features

Similarly, the distance value can also be extracted from performance datasets to adapt to the hand datasets. The distance of a task simply means that a path length of a participant hand took for each specific task. However, there is no recorded distance data in hand datasets. Therefore, a distance approximation should be developed. The TaskID is known for each datapoint in left side of Figure 5.8. The distance variable will be divided the number of total datapoints that belong to a specific task and resulted value will be a fixed distance variable. It can be argued that a fixed distance for a task will or will not be a meaningful addition to the hand dataset. However, the distance is proportional to the total number of datapoints as well as the time spent on that

datapoint. Each datapoint stands for a frame meaning that the movement in the virtual environment trigger the saving operation. Therefore, such advantages motivated us to integrate the approximate distance variable to the hand datasets without underestimate the shortcomings. The next important task is to decide on the ML model.

6.1 ML Model Selection

To select a ML model, this information will be used to realize basic needs. In addition, the aim of this thesis and the ML model is defined in the related sections. The dataset contains time series data that turned into a supervised labeled cross-sectional data. The aim is to predict the surgical skill level of a participant that is either intermediate or novice by classifying their surgical skill levels depending on their hand movement datasets.

The selection of a category is the hardest part of the ML model application. It is now possible to say the exact goal is prediction of a category. The data sizes are clear and there is a need for a supervised learning algorithm for these structured datasets. The Random Forest Classifier is detected as a suitable choice.

6.1.1 Random Forest Classifier

A Random Forest Classifier (RFC) is an estimator based on decision trees. Number of different decision trees makes prediction, and random forest classifiers uses these prediction power to estimate the average value of this prediction pool. This maximizes the accuracy while controlling the over-fitting issue. One important hyperparameter of RFC is bootstrap and the RFC model tries to build decision tree with whole dataset if the bootstrap value is false. Else the user has to be specified the max_sample hyperparameter that limits the number of samples.

The hand data set contains values that look like outliers and this situation made RFC a proper choice. It was discussed that the nature of hand movement data and experimental design causes this issue. For instance, a participant plays a scenario while controlling a haptic device. During this period, there is a decision-making process going on. The hand movement may be slow at those times and fast if the decision is

made. Another example may be the loosing depth perception during the experimental run. It may create fast movements while trying to find the right angle to see the scene properly. It is observed that the velocity approximation has very large values over thousand as well as small values. RFC prediction relies on a consensus of decision trees while selecting the output class. This reduces the sensitivity of RFC to outliers because the outliers will be in the far side of this forest that has little or no effect in the quality of decision-making process. As a result, there is no normalization process in the data preprocessing section. The hand datasets also have important categorical values that is extracted during the feature engineering process. RFC uses categorical value without problem and it even do not need to encode the categorical values.

6.1.2 Time Series Data as Supervised Learning

In this thesis, the data have a time dependency. The time dependency is reduced with a number of labeled, structured data that help us to phrase the time series data as supervised learning. To increase the strength of the supervised learning process, the nature of time series can be used and provide more insights. In a sequence, a current data point and the following data points has a natural relation. This relation can be used as a feature set such as the future sample will be like an output constructed with previous sample. This approximation makes a time series dataset a proper supervised learning problem because it will have a power to reveal more insights.

6.2 Feature Engineering

The data set now contains a distance approximation for each data point and the elapsed time. An approximate velocity value was derived using this information and added to the feature set. The approximation of velocity is expected to provide better insights. The data set has provided some ideas about how to handle feature engineering process. A participant is expected to have some distinct individual hand movement characteristics. Another expectation would be that all participants should derive hand movement patterns for each task. For example, a trained surgical actor is expected to exhibit more steady hand movements in case of a critical task. Moreover, this trained hand movements are expected to be stable between tasks throughout the scenario play. These expectations can be used to state such hypothesis, and produce many metric

sets. However, the aim this research study is to create data science framework. Therefore, the participant based and task-based rolling windows analysis are employed to create data frames that exhibits such defined characteristics of hand movements using Pandas' rolling function.

6.2.1 Feature Production

Participant-based Rolling Window with Velocity-based Features (PV)

The aim of this process is to create a new data frame using a number of existing features.

	Participant	TimeElapsed	Velocity	PartWStd	PartWMean	PartWVar	logs_div	logs_subt
550	1	0.099900	9.032157	33.948103	12.480908	1152.473667	0.994525	-0.000550
551	1	0.101457	8.893520	33.917626	12.474409	1150.405377	1.015589	0.001557
552	1	0.097965	9.210532	33.887174	12.468507	1148.340573	0.965582	-0.003492
553	1	0.101017	8.932248	33.856854	12.462124	1146.286581	1.031155	0.003052
560	2	0.100040	19.202593	NaN	19.202593	NaN	NaN	NaN
561	2	0.100783	19.061151	0.100015	19.131872	0.010003	1.007420	0.000742
562	2	0.100009	19.208663	0.083469	19.157469	0.006967	0.992321	-0.000774
563	2	0.101909	18.850527	0.167923	19.080733	0.028198	1.018999	0.001900
564	2	0.098087	19.584980	0.268331	19.181583	0.072002	0.962499	-0.003822
565	2	0.100952	19.029150	0.247939	19.156177	0.061474	1.029209	0.002865
566	2	0.099903	19.229054	0.228007	19.166588	0.051987	0.989604	-0.001049
567	2	0.099251	19.355258	0.221382	19.190172	0.049010	0.993480	-0.000651
568	2	0.099947	19.220603	0.207332	19.193553	0.042986	1.007006	0.000695
569	2	0.099490	19.308719	0.198838	19.205070	0.039536	0.995436	-0.000456
570	2	0.100753	19.066707	0.193192	19.192491	0.037323	1.012693	0.001263

Figure 6.2 A Preview for a Participant-Velocity based Data frame

The extracted additional features and their explanations can be seen as below:

- PartWStd: Calculation of standard deviation of Velocity value from the beginning of the related participant's data record. The window length is the length of the participant's data record.

- PartWMean: Calculation of mean of Velocity value from the beginning of the related participant's data record. The window length is the length of the participant's data record.
- PartWVar: Calculation of variance of Velocity value from the beginning of the related participant's data record. The window length is the length of the participant's data record.
- logs_div: Calculation of the division of the current time elapsed value with the previous datapoint. The window length is 2.
- logs_subt: Calculation of the subtraction of the previous datapoint from the current time elapsed value. The window length is 2.

The Figure 6.2 shows the change in dataset from participant 1 to participant 2. Some values are "NaN" value that will be changed to zero. Calculations start over whenever a new participant's data record is started.

Participant-based Rolling Window with Time-based Features (PT)

The aim of this process is to create a new data frame depending on elapsed time data. The extracted additional features and their explanations can be seen as below:

- PartDurWsum: Calculation of sum of TimeElapsed value from the beginning of the related participant's data record. The window length is the length of the participant's data record.
- PartDurWstd: Calculation of standard deviation of TimeElapsed value from the beginning of the related participant's data record. The window length is the length of the participant's data record.
- PartDurWmean: Calculation of mean of TimeElapsed value from the beginning of the related participant's data record. The window length is the length of the participant's data record.
- PartDurWvar: Calculation of variance of TimeElapsed value from the beginning of the related participant's data record. The window length is the length of the participant's data record.
- logs_div & logs_subt: Calculation is same with the previous section.

This data frame also has the “NaN” values that means that this value is also changed to zero for ML model application. All additional features are reset whenever a new participant’s data record is started.

One Second Rolling Window with Velocity-based Features (1SV)

The aim of this process is to create a new specific data frame with a different window. The window is now a one second time frame. The extracted additional features and their explanations can be seen as below:

- SecWstd: Calculation of standard deviation of Velocity value from the beginning of the related data record up to the one second. The window the length is one second.
- SecWmean: Calculation of mean of Velocity value from the beginning of the related data record up to the one second. The window the length is one second.
- SecWvar: Calculation of variance of Velocity value the beginning of the related data records up to the one second. The window the length is one second.

All additional features are reset whenever a new one second long of data record has completed.

One Second Rolling Window with Time-based Features (1ST)

The aim of this process is to create a new specific data frame depending on elapsed time with a one second rolling window. The extracted additional features and their explanations can be seen as below:

- TimeDursum: Calculation of summation of TimeElapsed value from the beginning of the related data record up to the one second. The window length is one second.
- TimeDurstd: Calculation of standard deviation of TimeElapsed value from the beginning of the related data record up to the one second. The window the length is one second.
- TimeDurmean: Calculation of mean of TimeElapsed value from the beginning of the related data record up to the one second. The window the length is one second.

- TimeDurvar: Calculation of variance of TimeElapsed value from the beginning of the related data record up to the one second. The window the length is one second.

Task-based Rolling Window with Velocity-based Features (TV)

The aim of this process is to create a new specific data frame similar to section "Participant-based Rolling Window with Velocity-based Features".

	Participant	TimeElapsed	Velocity	TaskWStd	TaskWMean	TaskWVar	logs_div	logs_subt
200	1	0.010014	73.308812	19.836490	76.098783	393.486342	1.007725	0.000077
201	1	0.012012	61.117846	19.815139	76.024620	392.639730	1.199467	0.001997
202	1	0.011211	65.479735	19.779882	75.972675	391.243726	0.933386	-0.000800
203	1	0.006771	108.415753	19.861420	76.131710	394.475992	0.603969	-0.004440
204	1	0.009993	73.466249	19.813555	76.118708	392.576943	1.475722	0.003221
205	1	0.011111	66.069810	19.777566	76.069927	391.152130	1.111949	0.001119
206	1	0.011203	65.529884	19.743100	76.019008	389.790012	1.008239	0.000092
207	1	0.007840	169.730076	NaN	169.730076	NaN	NaN	NaN
208	1	0.009896	134.468610	24.933622	152.099343	621.685505	1.262228	0.002056
209	1	0.011342	117.327442	26.718367	140.508709	713.871114	1.146097	0.001446
210	1	0.008779	151.580187	22.506854	143.276578	506.558481	0.774029	-0.002563

Figure 6.3 A Preview for a Task based Data frame

This time the window is each task. The extracted additional features and their explanations can be seen as below:

- TaskWStd: Calculation of standard deviation of Velocity value from the beginning of a task to the end of that task. The window length is the length of the related task.
- TaskWMean: Calculation of mean of Velocity value from the beginning of a task to the end of that task. The window length is the length of the related task.
- TaskWVar: Calculation of variance of Velocity value from the beginning of a task to the end of that task. The window length is the length of the related task.
- logs_div: Calculation of the division of the current time elapsed value with the previous datapoint. The window length is the length of the related task.

- logs_subt: Calculation of the subtraction of the previous datapoint from the current time elapsed value. The window length is the length of the related task.

The Figure 6.3 shows the change in dataset from TaskID 1 to TaskID 2. Some values are “NaN” that means the minimum calculation of the related feature needed to be more than one. All additional feature calculations start from scratch whenever a new TaskID is started.

Task-based Rolling Window with Time-based Features (TT)

The aim of this process is to create a new specific data frame depending on elapsed time data. The extracted additional features and their explanations can be seen as below:

- TaskDurWsum: Calculation of sum of TimeElapsed value from the beginning of a task to the end of that task. The window length is the length of the related task.
- TaskDurWstd: Calculation of standard deviation of TimeElapsed value from the beginning of a task to the end of that task. The window length is the length of the related task.
- TaskDurWmean: Calculation of mean of TimeElapsed value from the beginning of a task to the end of that task. The window length is the length of the related task.
- TaskDurWvar: Calculation of variance of TimeElapsed value from the beginning of a task to the end of that task. The window length is the length of the related task.
- logs_div & logs_subt: Calculation is same with the previous section.

Two Second Rolling Window with Velocity-based Features (2SV)

The aim of this process is to create a new specific data frame depending on a different window. The window is now a two second time period. The extracted additional features and their explanations can be seen as below:

- SecTaskWStd: Calculation of standard deviation of Velocity value from the beginning of the related data record up to the two second. The window the length is two second.
- SecTaskWMean: Calculation of mean of Velocity value from the beginning of the related data record up to the two second. The window the length is two second.
- SecTaskWVar: Calculation of variance of Velocity value from the beginning of the related data record up to the two second. The window the length is two second.

All additional features calculations starts over again whenever a new two second time frame has completed.

Two Second Rolling Window with Time-based Features (2ST)

The aim of this process is to create a new specific pandas data frame depending on elapsed time with a two second rolling window. The extracted additional features and their explanations can be seen as below:

- SecTaskWSum: Calculation of the variance of the "TimeElapsed" value from the beginning of the related data record up to the two-second length of time The window length is two seconds.
- SecTaskWstd: Calculation of the variance of the "TimeElapsed" value from the beginning of the related data record up to the two-second length of time The window length is two seconds.
- SecTaskWmean: Calculation of the variance of the "TimeElapsed" value from the beginning of the related data record up to the two-second length of time The window length is two seconds.
- SecTaskWvar: Calculation of the variance of the "TimeElapsed" value from the beginning of the related data record up to the two-second length of time The window length is two seconds.

6.3 ML Application

The RFC is applied using the designed data pipeline that is triggered via a “FullMLPipeline(True)” call. The train and test data has been constructed using the

SciKit-Learn's default splitter. The test size is used as 30% for all training jobs. A processed data frame is ready for RFC application except categoric features. All categoric features (Participant, TimeElapsed, Velocity, TaskID, Experiment) removed. Moreover, the none values replaced with zero values. The RFC algorithm has a parameter called n_estimator. The n_estimator is the number of decision trees. In other words, user can decide on the depth of the tree. The default value is 100 and it is used as is.

	index	Skill	PartWStd	PartWMean	PartWVar	logs_div	logs_subt
0	0	1	0.000000	50.345527	0.000000	0.000000	0.000000
1	1	1	1.848054	51.652299	3.415303	0.950650	-0.000720
2	2	1	126.957238	124.947211	16118.140200	0.195034	-0.011158
3	3	1	106.197663	113.409047	11277.943727	3.446140	0.006613
4	4	1	93.410319	106.101108	8725.487674	1.025045	0.000233
...	...	...	...	...	...	...	...
74173	74301	0	455.054984	491.227017	207075.038896	0.037334	-0.046574
74174	74302	0	455.053515	490.793653	207073.701378	26.702026	0.046423
74175	74303	0	454.966063	491.129484	206994.118897	0.044492	-0.046083
74176	74304	0	454.964311	490.697318	206992.523916	22.301892	0.045710
74177	74305	0	454.959292	491.125779	206987.957270	0.040175	-0.045933

74178 rows × 7 columns

Figure 6.4 A Preview for Ready Data Frame

The Skill column is used as a target set (see Figure 6.4) and the remaining columns (PartWStd, PartWMean, PartWVar, logs_div, logs_subt) will be used to create test and training sets. A statistical description of the data frame can be seen in Figure 6.5. To split the data frame, "train_test_split" method of the sklearn library's model selection module is used. This method returned the number of sub-data frames named x_training, x_test, y_training, y_test. The 30% of the 74178 rows is used as test set. Therefore, the length of the x_test and the y_test frames is 22254. Training frames has 51924 rows. Now the data frame is ready to fit the RFC model to our data.

	PartWStd	PartWMean	PartWVar	logs_div	logs_subt
count	74150.000000	74178.000000	74150.000000	74150.000000	74150.000000
mean	52.855121	106.780248	7698.182458	1.360288	-0.000002
std	70.032741	69.336259	25896.911384	2.412142	0.008117
min	0.038307	25.563265	0.001467	0.024087	-0.120343
25%	23.395511	79.024661	547.349942	0.855366	-0.001502
50%	30.525322	92.598522	931.795268	1.024565	0.000236
75%	39.613780	107.803135	1569.251578	1.196779	0.001772
max	506.483416	559.937471	256525.450850	51.373110	0.148399

Figure 6.5 Description of Sample Data Frame

6.3.1 Target

The target data set was prepared prior to the experimental application, as mentioned. The number of monitored, assisted, and performed operations is used to define the participants as novices or intermediates. The target dataset contains 12 intermediates with at least one performed operation, and the remaining participants are defined as novices with no performed operations. Accordingly, intermediates are encoded as 1 and novices as 0.

6.3.2 Implementation of Positional Data

The positional datasets are complete and coherent structured datasets. Therefore, they are ready to implement with some data alignment efforts. This time a big complete data frame with 335439 rows and 10 columns will be used (see Figure 6.6). Data alignment depends on one information.

All sub data frames, the positional data record of a virtual surgical tool is the common attribute. In some experiments, the camera may be used as a tool. In some settings, an independent tool is used with a fixed camera setting. Such differences in scenario design will not change the state of participant that is playing the scenario to control a virtual surgical device through a haptic interface. The angular velocity, angular acceleration and the rotational velocity data features derived from this common attribute.

	DataTime	AVx	AVy	AVz	AAx	AAy	AAz	RV	Participant	Skill
0	0.014582	1.343041e+06	1.373690e+06	1.363473e+06	92105100.0	9.420698e+07	93506350.0	2.355810e+06	N01	1
1	0.028444	-1.653294e+03	0.000000e+00	0.000000e+00	-97005740.0	-9.909747e+07	-98360470.0	1.653294e+03	N01	1
2	0.031147	-1.059633e+04	-2.119395e+03	0.000000e+00	-500685400.0	-5.088864e+08	-504323600.0	1.080620e+04	N01	1
3	0.040464	-1.229823e+03	0.000000e+00	0.000000e+00	-144283500.0	-1.474411e+08	-146344500.0	1.229823e+03	N01	1
4	0.050014	-1.199958e+03	0.000000e+00	0.000000e+00	-140755000.0	-1.438386e+08	-142768900.0	1.199958e+03	N01	1
...	...	...	...	...	...	...	...	...	...	...
335434	99.446830	7.864220e+01	0.000000e+00	0.000000e+00	-1029652.0	-1.797018e+04	-1037557.0	7.864220e+01	N28	0
335435	99.546640	5.740965e+01	5.739213e+01	-1.148193e+02	-1503527.0	-2.566182e+04	-1516006.0	1.406172e+02	N28	0
335436	99.647920	0.000000e+00	-5.655774e+01	5.657500e+01	-1482235.0	-2.641382e+04	-1492273.0	7.999693e+01	N28	0
335437	99.746350	0.000000e+00	5.819283e+01	1.164212e+02	-1525087.0	-2.601170e+04	-1534807.0	1.301549e+02	N28	0
335438	99.800520	0.000000e+00	1.057793e+02	-6.346436e+02	-2771362.0	-4.638953e+04	-2802890.0	6.433986e+02	N28	0

335439 rows × 10 columns

Figure 6.6 Positional Data Frame

As a result, the positional data frame is implemented with the RFC model with the help of similar technologies to train the machine with a data frame containing 334439 rows and 9 columns.

CHAPTER 7

RESULTS AND ANALYSIS

In this chapter, the application will be investigated, and the results will be presented in detail as well as the ML model. In the first part, the output of the ML application is handled. Then the ML framework and this research study as a whole will be evaluated. There are nine data frames designed in the feature engineering section. Those data frames were handled with RFC model.

7.1 Results

To summarize the calculated values, it has to be realized that the rolling windows methods creates a periodic observation using data. For instance, a participant with a 1000 datapoint exist in the dataset. The methodology starts with the datapoint 2. Standard deviation of two datapoints is recorded in the related cell. It goes all the way to the 1000 datapoint. In each datapoint the standard deviation calculation between 0 and current datapoint is recorded to the related cell in case of PartWStd. The process is repeated with mean and the variance calculations. The features PartWStd, PartWMean, PartWVar are used in PV data frame. In case of PT data frame, the same calculations are applied with the Time Elapsed value instead of Velocity.

The rolling windows size is fixed to 2 in length in case the features log_div and the log_subt. These features depend on two simple methods that takes two input and returns the calculation as below:

- log_div: Old TimeElapsed Value / Current TimeElapsed Value
- log_subt: Old TimeElapsed Value – Current TimeElapsed Value

Another feature set depends on the task based rolling windows. Each TaskID is filtered and used as a rolling window. The methodology starts with the datapoint 2 and goes all the way to the end of the task. For the features, TaskWStd and TaskWMean and TaskWVar, each datapoint between 0 and current datapoint's statistical observation is calculated and recorded to the related cell. These features are used in TV data frame. The features of TT data frame are named as TaskDurWSum, TaskDurWstd, TaskDurWmean and the TaskDurWvar. These features use the same rolling window approximation. This time the calculations are conducted using the Time Elapsed value instead of Velocity. In addition, log_div and log_subt features are calculated same as above formulas for each task observation.

The time-based rolling windows uses a fixed time period, 1 second as a length of window. In 1SV data frame, SecWstd, SecWmean and SecWvar features are calculated using Velocity feature for each time dependent periodic windows. In 1ST the Time Elapsed value is used instead of Velocity value. The 2SV and 2ST data frames contain same features that is calculated based on 2 second long of window for rolling statistics.

The first results were gathered from the application of ML model on the nine different data frames with the engineered feature sets. The overall data frame contains the each and every feature engineered and perfect result looks like a sign of overfitting. It will be ignored and will not be investigated as highest value because it needs to be investigated further. All accuracy results are rounded as they are long floating-point numbers.

PV has the highest accuracy while 1ST has lowest, 0.57. The approximate velocity calculation injected to the datasets looks like the strongest addition because average values of the velocity-oriented feature sets (PV, 1SV, TV) are the highest. 1SV data frame reaches the higher accuracy among other time dependent window approximations (1ST, 2SV, 2ST).

In addition, the positional dataset is implemented with the RFC model. The accuracy score is calculated as 94%. It is important to notice that the positional data frame do not exist in Table 7.1. It has the all-experimental data in a big complete dataset. Therefore, the structure of the dataset is not aligned with other individual data frames. It will be investigated in the following chapters.

Table 7.1 Accuracy Results for Engineered Data Frames

Experiment	PV	PT	1SV	1ST	TV	TT	2SV	2ST	overall
ECES0101_D	0.98	0.73	0.83	0.59	0.89	0.85	0.65	0.61	0.99
ECES0301_D	0.94	0.68	0.84	0.57	0.85	0.76	0.66	0.58	0.99
ECES0301_N	0.95	0.71	0.84	0.60	0.88	0.80	0.68	0.61	0.99
ECES0201_D	0.95	0.72	0.88	0.58	0.90	0.81	0.70	0.62	0.99
ECES0201_N	0.97	0.76	0.88	0.60	0.93	0.86	0.70	0.63	0.99
ECES0401_D	0.89	0.68	0.86	0.58	0.84	0.75	0.64	0.62	0.99
ECES0401_N	0.91	0.66	0.86	0.58	0.85	0.74	0.64	0.59	0.99
ECES0102	0.95	0.74	0.84	0.62	0.86	0.80	0.71	0.64	0.99
ECES0302	0.95	0.73	0.85	0.63	0.90	0.81	0.72	0.66	0.99
ECES0402	0.87	0.70	0.70	0.65	0.83	0.77	0.73	0.66	0.99
Range (Max-Min)	0.11	0.10	0.18	0.08	0.10	0.12	0.09	0.08	0
Average	0.94	0.71	0.84	0.60	0.87	0.79	0.68	0.62	0.99
Total Feature	5	6	3	4	5	6	3	4	36

*P: Participant-based Rolling Window
PV: Velocity-based Features
PT: Time-based Features
1S: One Second Rolling Window
1SV & 1ST: Velocity-based & Time-based Features

*T: Task-based Rolling Window
TV: Velocity-based Data frame
TT: Time-based Data frame
2S: Two Second Rolling Window
2SV & 2ST: Velocity-based & Time-based Features

7.2 Analysis

A ML model can even be evaluated using its own training set. It could provide valuable idea to improve the ML approximation itself. However, the output would not be valid for analysis or making predictions. It is because of the ability to learn, which is basically the heart of the learning machines. In other words, the learning activity itself includes a kind of memorizing process. This puts the time dependency in a critical position. It creates a natural relationship between each observation. Any model that realizes this pattern perfectly would make the perfect decision. The feature engineering activities in this thesis study is an attempt to turn the weakness into a useful entity to work on a time series data. Now, it is assumed that the input and output observations can be used to predict the new output using the input variable. However, the relation between the selected ML approximations and how the data will be used to fit and

evaluate the model are the main issues that should be faced. For instance, RFC require two arrays to be fitted like some other classifiers. Training samples keeps the training data with some specific dimension depending on the number of features. The target values hold the target values. Hence, some general evaluation approximations will be used to investigate how the applied ML model works.

7.2.1 Evaluation

In this section, the experiments, ECES0301_D and the ECES0201_N, were selected along with the positional dataset for demonstration purposes. The selection depends on the average value of highest and lowest performing data frames. ECES0301_D is exactly having the average accuracy value of the highest performing data frame, PV. The ECES0201_N's accuracy value is also same with the average value on the 1ST data frame that has the lowest average accuracy score. These two data frames are selected for these reasons. Moreover, the positional dataset gave the accuracy score 94% and it exists in the Figure 7.1 as it provides a different perspective of the same experimental output.

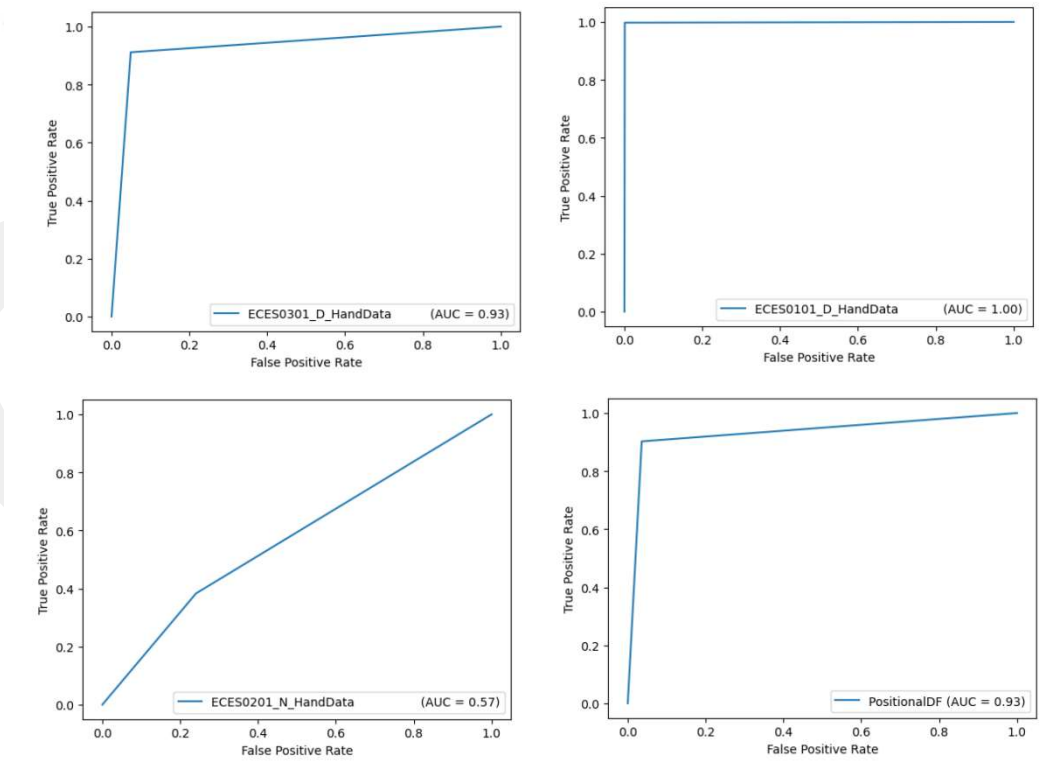


Figure 7.1 ROC Curve and AUC Score

Receiver Operating Characteristic (ROC) Curve and Area under the Curve (AUC) score

An ROC curve (receiver operating characteristic curve) is a graph that presents the performance of a classification model at all classification thresholds. The True Positive Rate (TPR) and False Positive Rate (FPR) is used to create a ROC curve. AUC is a measure that shows the area under ROC curve to calculate the points that classifies more item as positive. Therefore, the true and false positive rates is expected to be maximized. The ROC curve and AUC score is perfect in ECES0101_D as expected (see Figure 7.1). PV of ECES0301_D and PositionalDF shows near perfect roc curve and 0.93 AUC score. The ECES0201_N's AUC metric is 0.57. It means that the model cannot distinguish either positive or negative value.

Table 7.2 Classification Report for PositionalDF

PositionalDF	Precision	Recall	F1-score	Support
Novice - 0	0.92	0.96	0.94	41032
Intermediate - 1	0.94	0.88	0.91	27527
Accuracy			0.93	68559
Macro Average	0.93	0.92	0.93	68559
Weighted Average	0.93	0.93	0.93	68559

Classification Scores

The precision score gives a hint about model's prediction ability of positives as true. It can be seen that PositionalDF has high precision score so it means that the model produces less false positive for intermediates (see Table 7.2).

Table 7.3 Classification Report for ECES0301_D

ECES0301_D - PV	Precision	Recall	F1-score	Support
Novice – 0	0.94	0.95	0.95	2578
Intermediate – 1	0.93	0.91	0.92	1749
Accuracy			0.93	4327
Macro Average	0.93	0.93	0.93	4327
Weighted Average	0.93	0.93	0.93	4327

The prediction power looks like in a good level with an accuracy of 94%. The Table 7.3, precision value shows smaller difference than the PositionalDF. In Table 7.4, the

probability of predicting intermediates as true is low. Therefore, false positive rate is expected to be high.

Recall is related with false negatives. It is 1 if the model produces no false negative. Therefore, the probability of the model to predict the surgical skill as a true novice is high. The sample size of novice is bigger than intermediate, and this situation supports the high recall value. Table 7.4 recall value is the highest metric value in that table. The support value which the model trained shows that there were 3490 novices while intermediates are 2182. The novice samples are higher than intermediate. F1-score is better in ECES0301_D and PositionalDF. It is important to notice that the sample size is related with the quality of the applied model. The imbalanced distribution between classes is a cause of low quality and it is more obvious in Table 7.4.

Table 7.4 Classification Report for ECES0201_N

ECES0201_N – 1ST	Precision	Recall	F1-score	Support
Novice – 0	0.66	0.76	0.71	3490
Intermediate - 1	0.50	0.38	0.43	2182
Accuracy			0.61	5672
Macro Average	0.58	0.57	0.57	5672
Weighted Average	0.60	0.61	0.60	5672

The support metric is the count of the sample set, as we mentioned. For instance, there are 2578 novices and 1749 intermediates in ECES0301_D. The macro average is calculated using the precision, recall, and f1-score values between classes and reveals the class imbalances more accurately (see Table 7.4). The weight is calculated using the count of class samples in the weighted average value. A high value means the model favors the majority class, depending on the sample size.

Improve the model with Hyperparameter Tuning

RFC model like other ML approximations have parameters. These parameters set the way the model learn. For instance, n_estimator parameter was mentioned as the number of decision trees while building the RFC model.

The feature engineering and preparing a good data structure that accommodates the selected ML model perfectly is the core idea in this research study. However, selection of hyperparameters is a crucial process that have been overlooked at the expense of focusing on the main subject. It is a configuration of the selected ML approximation that may be the part of the ML model designing phase. To improve the ML model (see Figure 4.1), PositionalDF, PV data frame of ECES0301_D and the 1ST data frame of ECES0201_N have been tuned with hyperparameters to investigate further possibilities to improve the accuracy score measure.

The RFC model contain many parameters. All previous accuracy estimations have been calculated with the default parameter set of RFC of Scikit-Learn. The n_estimator parameter has been discussed before. The below definitions are taken from Scikit-Learn's documentation [114]:

- max_depth: The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- min_samples_split: It is the minimum number of samples required to split an internal node. The default value is an integer or float 2.
- min_samples_leaf: The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least
- min_samples_leaf: Training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression. The default value is an integer or float, 1.

RandomizedSearchCV is a search approximation to find the best set of hyper parameters randomly. It takes a sample, fits it and check score unless configured for further purposes with additional parameters. The optimization is a cross-validation search with fixed number of sample if they are presented as a list. The number of iterations through this search is decided by the user with n_iter attribute.

Accordingly, the parameter list for the RFC model is prepared:

- n_estimators: [10, 60, 110, 160, 210, 260, 310, 360, 410, 460, 510, 560, 610, 660, 710, 760, 810, 860, 910, 960]
- max_depth: [None, 3, 5, 10]
- min_samples_split: [2, 4, 6, 8, 10, 12, 14, 16, 18]
- min_samples_leaf: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

This parameter dictionary is used as a parameter distribution for randomized search. The other attributes provided to the RandomizedSearchCV are n_iter as 20 and CV as 5. The results are provided in Table 7.5.

First, it is important to notice that the previous results are gathered from Table 7.1. Those values are rounded accuracy results. The accuracy score of ECES0301_D - PV is changes between 0.93 and 0.94 with the default settings of RFC. Whence, the results show that the tunning procedure has no effect in it.

Table 7.5 Hyperparameter Tunning

Experiment	Previous Accuracy	Accuracy After Tunning
ECES0301_D – PV	0.94	0.93
ECES0201_N – 1ST	0.60	0.66
PositionalDF	0.94	0.98

The hyperparameter tunning created a positive effect on ECES0201_N – 1ST and PositionalDF. First, it is important to notice that the previous results are gathered from Table 7.1. Those values are rounded accuracy results. The accuracy score of ECES0301_D - PV is between 0.93 and 0.94 with the default settings of RFC. Whence, the results show that the tunning procedure has no much effect in it.

7.3 Further Experiments

The ML model (see Figure 4.1) integrity is evaluated in the previous section. All the steps of this research have been examined up to this point. Now, let's try to observe the coherence of our work by bringing these pieces together. The data has been prepared so a results table is prepared with the accuracy of engineered data frames using RFC model (see Table 7.1). In that table there are 9 ready datasets for each

experiment. These results from 90 data frames reveals different information. However, PositionalDF (See Appendix A.1) could not be fitted into that table because it is a complete data frame that has the all-experimental data unlike other datasets. Moreover, the each of the 9 data frames have the exact same feature sets. It means that the remaining data set can be aligned to compare with the positional data results. The accuracy results (see Table 7.6, ACC) are the basic evaluation metric. The datasets are fitted to the ML model with tunned hyperparameters using Randomized CV search.

Table 7.6 Full Data Frame Results

Experiment	F#	R#	PAA	ACC	AUC
FullPV	5	227036	0.94	0.90	0.97
FullPT	3	227036	0.71	0.69	0.74
Full1SV	6	227036	0.84	0.77	0.85
Full1ST	4	227036	0.60	0.64	0.63
FullTV	5	227036	0.87	0.82	0.91
FullTT	3	227036	0.79	0.76	0.84
Full2SV	6	227036	0.68	0.66	0.70
Full2ST	4	227036	0.62	0.65	0.66
FullComplete	36	227036	0.99	0.99	0.99
PositionalDF	7	228530	0.94	0.94	0.98
Range (Max-Min)	33	-	0.37	0.30	0.35
Average	7.9	-	0.80	0.78	0.81
Total	79	-	-	-	-
F#: Number of Features R#: Number of Rows		PAA: Previous Average Accuracy ACC: Accuracy AUC: Area Under the ROC Curve			

The accuracy results show the ability to successfully predict a surgeon skill level in a certain percentage. In the same table, PAA shows the previous accuracy score calculated by taking the average value of PV data frame of 10 experiments (see Table 7.1, Average). Therefore, all data frames have less accuracy than the previous averages except Full1ST and the Full2ST. It may be evaluated as an expected result because these data frames have fixed time windows. The increase in the number of windows affected them positively. There is no detected data leak but the FullComplete data frame shows signs of overfitting. It looks like that the RFC memorizes the relation completely when the 36 features together fitted to the model. This data frame will be excluded from the comparison, will not be used as a best score. Other experiments are expected to be in a value close to the PAA since the data contain all experiments. There

are small differences between PAA and ACC values in all the datasets so the results look like close to the expectations.

The FullPV (94%) is again the best performing data frame among engineered data frames followed by FullTV (71%). Both of them provides the statistical observations based on Velocity based calculations rolled in a participant-based windows.

However, it should be noted that the FullPV contains two additional attribute logs_div and logs_subt. To measure the power of these two features, the data are fitted to the ML model again without them. The optimal parameters for the data frame fullPV were ready; "n_estimators" is 660, "min_samples_split" is 10, "min_samples_leaf " is 5 and the "max_depth" is None. After fitting the data to the RFC model using these parameters, the accuracy score is 0.90. It shows that these two features have positive affect on accuracy of the model 0.4%.

Now it is possible to wonder that what will be the accuracy of the model if it has only the features logs_div and logs_subt. In the same conditions, the data frame that contain only these two features has an accuracy score 0.59. This shows that the velocity approximation is a vital feature that provides a strong observation sequence to successfully distinguish the skill level of the subjects.

FullPT with an accuracy value 0.69, and Full TT with an accuracy value 0.76 are other data frames that has a feature set calculated based on Time Elapsed observations for each rolling windows. Their rolling window length is the task length. The accuracy values are lower than velocity-based feature set previously mentioned. Therefore, it shows that the approximate Velocity is a more successful measurement to distinguish the surgical skill than Time Elapsed value.

The time-based data frames are based on two time periods: 1 second (1ST, 1SV) and 2 seconds (2SV, 2ST). The feature sets of the 1SV and 2SV data frames are calculated using the approximate velocity measure between 1-second-long and 2-second-long records.

In time-based division of the data set creates lower observation of the data sequence to make meaningful inferences. The number of windows is higher than other windows. For instance, a task's completion time is 2.08 seconds (see Figure 5.4, time), and there are 3 windows in that task. However, the window count will be 1 if it is task-based and 0 if it is a participant-based rolling window. The low number of periods between the rolling windows looks like causes less accurate detection of the surgical skills. 1ST and 2ST are two data frames that uses the elapsed time measurement. Similar to the previous results, elapsed time-based calculations of the feature set give lower accuracy than velocity-based measurements.

To summarize, the results are presented in detail to further analyze the dataset characteristics using different measurement tools, metrics. The PositionalDF data frame has the best accuracy, 94%.

7.3.1 Evaluation

First evaluation metric has been provided in Figure 7.6, AUC score. AUC score is a good measure for the analysis. AUC takes all possible threshold into account to measure the performance of all-possible classification condition. The rank is more important in AUC calculation than the value itself. Moreover, it can evaluate the quality of the model in any classification threshold. The highest AUC score is FullPV's score and it nearly equal to PositionalDF.

AUC score is the area under the ROC curve as it is mentioned previously. The ROC curve provides a visual solution to investigate the model's true positive rate (tpr) versus false positive rate (fpr) among those prediction results. For instance, predicting the intermediate values correctly among prediction results, while the real value is intermediate is called true positives. The prediction of skill as intermediate while it actually is novice is called false positives. True positive is the prediction of intermediate skill of a participant while the truth is also the same. False negative means the model's prediction of novice is a wrong answer.

It looks like investigation of predictions and the probability distribution is the key statistical perspectives to evaluate a ML model. There is a function for this purpose the "predict_proba" ready to be used with the trained model like below:

- Model.predict_proba(X_test)

It returns an array of prediction probabilities for all data points in target by comparing it with the predictions. These probabilities can be used to create confusion matrix for all data frames (see Appendix B, Confusion Matrix Heat Map).

Table 7.7 Classification Report

Experiment	P		R		F1		Macro	Weighted
	1	0	1	0	1	0		
FullPV	0.89	0.90	0.85	0.93	0.87	0.92	0.89	0.90
FullPT	0.64	0.71	0.50	0.81	0.76	0.56	0.66	0.68
Full1SV	0.76	0.77	0.61	0.87	0.67	0.82	0.75	0.77
Full1ST	0.68	0.63	0.18	0.94	0.29	0.76	0.58	0.62
FullTV	0.81	0.83	0.73	0.89	0.77	0.86	0.81	0.82
FullTT	0.73	0.78	0.63	0.84	0.81	0.68	0.74	0.76
Full2SV	0.67	0.63	0.37	0.85	0.47	0.75	0.62	0.65
Full2ST	0.69	0.64	0.22	0.93	0.34	0.76	0.60	0.63
FullComplete	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
PositionalDF	0.94	0.93	0.88	0.96	0.90	0.94	0.93	0.93
Range (Max-Min)	0.30	0.30	0.70	0.15	0.61	0.38	0.35	0.31

P: Precision
R: Recall
F1: F1-score
1, 0: Intermediate, Novice
Macro: Mean Macro Average
Weighted: Mean Weighted Average

These classification metrics have been applied in section 7.2.1 for individual data frames. The exact results of these metrics can easily be calculated with the defined probabilities using below equations:

- Precision = true positives / (true positives + false positives)
- Recall = true positives / (true positives + false negatives)
- F1 score = 2 * (Precision * Recall) / (Precision + Recall)

The sample number has a clear effect in the classification metrics. The total number of intermediate samples is 27225 and novice sample is 40886 for FullPV (see Appendix B, Confusion Matrix Heat Map). The PositionalDF's sample size is 68559 which contains 27527 novices, and 41032 intermediates. In Full1ST, Full2SV and Full2ST data frames, the Recall value is below 0.5. It means that he RFC classifier has a high number of False negatives. One solution can be a reorganization of hyperparameters. Elimination of some participants can be a solution if the problem is class balance. However, the class distributions are same between all engineered data frames. The windows in these data frames are determined using time frames as 1 second or 2 seconds. It causes a higher number of windows comparing with the task or participant based rolling window calculations. It also shows that the purely time dependent analysis (Full1ST, Full2ST) provides less insights about hand movement data. Recall score effects the F1 score also. The results look like balanced for other data frames.

Feature Importance

The importance of logs_div and logs_subt have been evaluated in the previous section by removing the other features and applying the model again. Once the most important feature is calculated, they can be used to find out how model perform without others.

Table 7.8 Accuracy for Important Features

Experiment	PA	ACC
FullPV	0.90	0.93
FullPT	0.69	0.69
Full1SV	0.77	0.79
Full1ST	0.64	0.58
FullTV	0.82	0.88
FullTT	0.76	0.80
Full2SV	0.66	0.62
Full2ST	0.65	0.61
FullComplete	0.99	0.92
PositionalDF	0.94	0.85

PA: Previous Accuracy
ACC: Accuracy

Now it is possible to test the model's integrity using the most important two features to calculate the accuracy for each data frame using RFC model. The accuracy results

show that the accuracy can be increased by this process. The velocity-based features' accuracy increased (FullPV, FullTV). Features created using task-based window have little or no change (FullPT, FullTT). All data frames with time-based rolling window have less accuracy than before except 1SV. Finally, the accuracy of complete set and positional data set is decreased by this change.

To summarize, it is possible to create more experiments to investigate the data structure as well as the integrity of the model with a properly created pipeline. The results show that the hand movement data sets contain meaningful patterns in their motion based, characteristic measures and results support this with an acceptable level of proof.

CHAPTER 8

DISCUSSIONS AND CONCLUSION

This research study has been started with a background analysis (see Chapter 2, Background of the Study) that is an attempt to reveal the lack of systematic data handling with a currently available rich set of open-source data science tools as a current challenge that the research community should focus on. To overcome this challenge, it is necessary to integrate data science tools into the simulation design (see Section 3.1 Experimental Application) that will probably improve the surgical education programs by providing more sophisticated assessment capacity. This thesis study basically tries to understand the specific surgical skill differences between novice and intermediate residents (see Chapter 7, Results and Analysis) using experimental data produced by a computer-based VR simulator (ECE). For this purpose, hand movement data (see Section 5.1 Data Exploration) collected through haptic devices while 28 participants worked on tasks as part of surgery scenarios (see Section 3.1.1 Experimental Setup and Procedure) has been used. The dataset collection contains hand movement data as well as performance-based measurements like task duration. Firstly, computer simulation-based surgical training environment data collection and the suitable set of data science tools and environments (see Section 4.3 Choosing the Right Toolbox) were analyzed. Then, a data preprocessing phase is conducted to architect the datasets according to the ML model. Such processed datasets can be used as a guide to make the necessary changes to the environment design in order to create a dataset ready for ML model implementation. This practice can standardize a framework for how a data science pipeline can be built on top of a specific virtual reality environment. Moreover, feature engineering methodology is presented with the help of the created data structure. Hand movement-based analysis is extended to reach high accuracy, and the integrity of the results is investigated with the help of a data science toolset.

The applied ML algorithm is RFC. RFC is based on decision trees. Decision trees are top-down approach. There is a root node at the top. The following nodes goes through the different layers up to the leaves that located at the bottom of the decision trees. This provides a rigid structure. For instance, it is possible to use a decision tree algorithm to divide the data into two, intermediate and novice to calculate the total numbers. This creates a hierarchy that inherits some information from upper layers. This information flow ends in the bottom nodes called leaves. The number at leaves reveals the total number of categories in the case of regression problem. The tree uses the existing pattern and can make decision for new observations. In other words, the tree can memorize the information through this organizational structure. In this manner, the order of the node organization is important. A decision tree tries to make categorization in this rigid structure with yes or no questions during the classification task.

This architectural view limits the usability of the decision trees in ML. This need causes new approximations like Ensemble Learning (EL). It is an attempt to remove such shortcomings of using a rigid structure and provides better generalizing ability. The core idea is the ability to use many outputs in combination to find an answer. It even makes it possible to employ many algorithms together to accomplish a work. The selected RFC model is a popular EL model. The Forest means the forest of decision trees that are built by using different subset of features. All or some of the decision trees make a prediction for the new data and the final result is chosen based on the number of decision count. This process is called majority voting mechanism of RFC during the prediction. SciKit-Learn has its own RFC library under its ensemble module. The basic implementation of RFC contains the workflow as below:

- Create a decision tree based on a random node with a subset of random data records (features).
- Repeat the previous process according to the user specified parameters for the RFC (number of decision trees, etc.).
- All decision trees make a prediction for a new record and classifies the new data depending on the majority vote accordingly.

The RFC algorithm has the capability to have a diverse representation of knowledge. It is vital to recognize that feature engineering and data processing increase the chance of building up that basic structure to achieve an accurate model. A high number of trees makes the model diverse in terms of the probability; hence the model may provide better insights. However, this may be a curse if the data lacks diverse knowledge. It means that a similar subset of the features always wins the majority voting mechanism to limit the chance of a fair distribution of that voting power. This causes a skewed ensemble that overfits the training dataset (see Figure 7.6; Experiment Fullcomplete). In such cases, the generalization ability will be damaged. The ML model may have difficulties on identifying the target set. The confusion matrix provides a visually satisfying graphics to observe the sample ratio predicted right or wrong clearly (APPENDIX B. Confusion Matrix Heat Map). For instance, the PositionalDF gives a 94% accuracy with 39417 true negative, 1615 false negative 3218 false positive and 24309 true positive values. Such information gives an idea to evaluate the model's integrity successfully.

The accuracy achieved by our random forest classifier with only two features' ranges from 0.94 to 0.64 (see Table 7.6). 94% accuracy means that the generalization ability gets most of the results right. This is high accuracy. It can be improved in a number of different ways, but the need for accuracy can change. Let's think about a ML pipeline developed for a dynamic evaluation framework that will work as a live feedback system. The stream of data is processed, and the skill feedback label turns into intermediate at a specific time with 64% accuracy. It means that the feedback system works with the help of available data. Therefore, the need for accuracy may also be thought of as a relative property. It is important to notice that the prediction is conducted to estimate the surgical skill level of each micromovement of the hands.
In the classification report (see Table 7.7), the FullPT data frame's recall value is high, with a value of 0.81 for the novice class. However, it is low for the intermediate class at 0.50. The same data frame has a precision metric of 0.64 for intermediates, and novices have a low f1 score of 0.56. The F1 score is a mirror of the precision and recall results of two classes (see Section 7.3.1 Evaluation). The model is performing better on class 0s (intermediates) and is not bad for class 1. However, the data frames Full2ST,

Full2SV, and Full1ST have a bad recall score for class 1. It looks like a combination of low-performing time-based rolling windows with a low number of samples. In these model applications, the count of intermediates is low (27225), and the novices have a higher sample rate (40886).

To recapture the process, the data preprocessing has been started with the investigation of distinct databases for hand movement (see Figure 5.5) and performance data (see Figure 5.4). The hand movement data set turned into a rich data set using one feature, DataTime, in combination with the distance attribute of the performance dataset. It may be possible to take advantage of the time series interaction of each frame for such a data structure. However, these datasets adapted to a different architectural perspective, and this process itself was also a useful practice. The results show that this approximation with a velocity measure gives results with high accuracy.

The findings in this thesis study show that confidence in using hand movement as an objective measurement tool is high. Moreover, the most important features of each data frame (see Appendix C, Feature Importance) are also a way to explore the data set and its features, as well as the integrity of the data science pipeline. For example, the most important features of the positional data frame are detected as AAz and AAy (see Appendix Figure C.3, PositionalDF Feature Importance). Angular acceleration in these specific directions reveals strategic knowledge about hand movement. The importance of the acceleration direct to the z coordinate shows that the depth perception of the participants provides clear signs to connect this movement with the surgical expertise. In addition, it can be inferred that the acceleration direct to the y coordinate may be a proof the relation of rotational speed with surgical skill. Another important gain for this research study is the ability to create, measure, and evaluate this pipeline from raw data to the Unity environment. It means that it is possible to measure the success of features prior to the environment development. The high accuracy of the positional data frames shows that the application development and design strategies can take advantage of data science framework like this study. It will result in a more accurate recording strategies, data structure and game design as well as robust the effect of the outputs.

The most important achievement of this thesis study is the ability to connect the raw micromovement of hand data to surgical skill. The results show that a proper data pipeline can be integrated into a virtual environment with an accurate, objective measurement framework. The accuracy results show that it is possible to reach high accuracy results with a proper framework constructed using the data science literature, surgical skill assessment corpus, and other knowledge bases.

In conclusion, the proper sets of tools have been employed to build a deep understanding of the raw data. The surgical skills and knowledge required to establish surgical competency and the approaches to addressing the limitations have been investigated in the literature. This study contributes to the field of surgical education by implementing the theory and methods of software engineering to better classify intermediate and novice surgical residents. This data-oriented process is designed to improve the assessment of surgical training programs and the quality and feedback system of VR-based surgical training programs. The quality measurement techniques and discovering important features to improve skill level estimation accuracy required for the development of surgical capabilities have been analyzed in depth with the help of existing studies. The data analysis effort has been built up to explore the ML methods and feature extraction methods, how they are related to hand movements or skill assessment concepts, and how they are applied using haptic-like technologies. The previous efforts to measure hand movements have been improved to a 94% accuracy level. The authors think that this attempt shows that it is possible to connect the surgical instinct with the latest software engineering technologies and, finally, to explore the behavioral characteristics of surgical hand movements.

CHAPTER 9

LIMITATIONS AND FUTURE WORK

In this research study, the experimental dataset made it hard to adapt the many deep learning approximations, mainly because of the data volume. It is possible to increase the participant number to cover more surgical skill levels and increase the diversity of the dataset. Another issue is that the design of the software shows no sign of a data-oriented perspective in a systematic way. In the future, the pre-analysis can be adapted to the software development lifecycle to improve the quality of the datasets and even adapt the analysis tool to the game environment.

In future, the temporal nature of the hand movement dataset can be analyzed with the additional experimental design studies. This framework can be used as a guide to build a novel VR environment that is capable to measure the surgical skill. Moreover, the data pipeline of this thesis can be integrated to a VR environment to provide a dynamic feedback system. Such dynamic feedback system can be used to realize the surgical trainee's skill dynamically and provide the proper experimental scenarios according to the detected skill level information.

REFERENCES

[1] H. Wenglinsky, "Does It Compute? The Relationship between Educational Technology and Student Achievement in Mathematics," Policy Information Center, Mail Stop 04-R, Educational Testing Service, Rosedale Road, Princeton, NJ 08541-0001; Tel: 609-734-5694; e-mail: pic@ets, Sep. 1998. Accessed: Mar. 01, 2023.

[2] G. Tonbul, D. Topalli, and N. E. Cagiltay, "A systematic review on classification and assessment of surgical skill levels for simulation-based training programs," *International Journal of Medical Informatics*, vol. 177, p. 105121, Sep. 2023.

[3] T. R. Gadacz and M. A. Talamini, "Traditional versus laparoscopic cholecystectomy," *The American Journal of Surgery*, vol. 161, no. 3, pp. 336–338, Mar. 1991.

[4] A. Darzi, V. Datta, and S. Mackay, "The challenge of objective assessment of surgical skill," *The American Journal of Surgery*, vol. 181, no. 6, pp. 484–486, Jun. 2001.

[5] "Fundamentals of Laparoscopic Surgery - A SAGES Wiki Article," *SAGES*, Jun. 10, 2023. https://www.sages.org/wiki/fundamentals-laparoscopic-surgery/ (accessed Jun. 13, 2023).

[6] G. B. Hunt, "Principles of operative technique," in *BSAVA Manual of Canine and Feline Surgical Principles*, British Small Animal Veterinary Association, 2018, pp. 264–276.

[7] K. Giacomino, R. Caliesch, and K. M. Sattelmayer, "The effectiveness of the Peyton's 4-step teaching approach on skill acquisition of procedures in health professions education: A systematic review and metaanalysis with integrated meta-regression," *PeerJ*, vol. 8, 2020.

[8] Z. Su, Y. Liu, W. Zhao, Y. Bai, N. Jiang, and S. Zhu, "Digital technology for orthognathic surgery training promotion: a randomized comparative study," *PeerJ Computer Science*, vol. 10, p. e13810, 2022.

[9] V. Garousi and M. V. Mäntylä, "A systematic literature review of literature reviews in software testing," *Information and Software Technology*, vol. 80, pp. 195–216, 2016.

[10] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.

[11] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[12] Y. Shakeel, J. Krüger, I. V. Nostitz-Wallwitz, G. Saake, and T. Leich, “Automated Selection and Quality Assessment of Primary Studies: A Systematic Literature Review,” *Journal of Data and Information Quality (JDIQ)*, vol. 12, no. 1, pp. 1–26, 2019.

[13] C. Richards, J. Rosen, B. Hannaford, C. Pellegrini, and M. Sinanan, “Skills evaluation in minimally invasive surgery using force/torque signatures,” *Surgical endoscopy*, vol. 14, no. 9, pp. 791–798, 2000.

[14] J. Rosen, B. Hannaford, C. G. Richards, and M. N. Sinanan, “Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills,” *IEEE transactions on Biomedical Engineering*, vol. 48, no. 5, pp. 579–591, 2001.

[15] M. K. Chmarra, W. Kolkman, F. W. Jansen, C. A. Grimbergen, and J. Dankelman, “The influence of experience and camera holding on laparoscopic instrument movements measured with the TrEndo tracking system,” *Surgical endoscopy*, vol. 21, no. 11, pp. 2069–2075, 2007.

[16] S. Yamaguchi *et al.*, “Construct validity for eye–hand coordination skill on a virtual reality laparoscopic surgical simulator,” *Surgical endoscopy*, vol. 21, no. 12, pp. 2253–2257, 2007.

[17] J. L. Moyano-Cuevas *et al.*, “Validation of SINERGIA as training tool: a randomized study to test the transfer of acquired basic psychomotor skills to LapMentor,” *International journal of computer assisted radiology and surgery*, vol. 6, no. 6, pp. 839–846, 2011.

[18] J. Takeda, I. Kikuchi, A. Kono, R. Ozaki, J. Kumakiri, and S. Takeda, “Efficacy of short-term training for acquisition of basic laparoscopic skills,” *Gynecology and Minimally Invasive Therapy*, vol. 5, no. 3, pp. 112–115, 2016.

[19] J. Jeekel, “Crucial Times for General Surgery,” *Annals of Surgery*, vol. 230, no. 6, p. 739, 1999.

[20] R. J. Gray, K. Kahol, G. Islam, M. Smith, A. Chapital, and J. Ferrara, “High-fidelity, low-cost, automated method to assess laparoscopic skills objectively,” *Journal of surgical education*, vol. 69, no. 3, pp. 335–339, 2012.

[21] A. Buia, F. Stockhausen, N. Filmann, and E. Hanisch, “3D vs. 2D imaging in laparoscopic surgery—an advantage? Results of standardised black box training in laparoscopic surgery,” *Langenbeck’s archives of surgery*, vol. 402, no. 1, pp. 167–171, 2017.

[22] N. V. Patel, J. M. Robbins, and C. J. Shanley, “Low-fidelity exercises for basic surgical skills training and assessment,” *The American journal of surgery*, vol. 197, no. 1, pp. 119–125, 2009.

[23] J. Cifuentes, M. T. Pham, P. Boulanger, R. Moreau, and F. Prieto, “Towards a classification of surgical skills using affine velocity,” *IET Science, Measurement & Technology*, vol. 12, no. 4, pp. 548–553, 2018.

[24] A. G. Gallagher *et al.*, “Outlier experienced surgeon’s performances impact on benchmark for technical surgical skills training,” *ANZ journal of surgery*, vol. 88, no. 5, pp. E412–E417, 2018.

[25] J. Choque-Velasquez, R. Colasanti, J. Collan, R. Kinnunen, B. R. Jahromi, and J. Hernesniemi, “Virtual reality glasses and ‘eye-hands blind technique’ for microsurgical training in neurosurgery,” *World neurosurgery*, vol. 112, pp. 126–130, 2018.

[26] N. K. Prasad, C. Kvasnovsky, E. S. Wise, and S. M. Kavic, “The right way to teach left-handed residents: strategies for training by right handers,” *Journal of surgical education*, vol. 75, no. 2, pp. 271–277, 2018.

[27] J. Binkley, A. D. Bukoski, J. Doty, M. Crane, S. L. Barnes, and J. A. Quick, “Surgical simulation: markers of proficiency,” *Journal of the American College of Surgeons*, vol. 225, no. 4, p. S182, 2017.

[28] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang, “Current status, opportunities and challenges of augmented reality in education,” *Computers & education*, vol. 62, pp. 41–49, 2013.

[29] E. Klopfer, *Augmented Learning: Research and Design of Mobile Educational Games*. MIT Press, 2008.

[30] J. P. Richards, A. J. Done, S. R. Barber, S. Jain, Y. Son, and E. H. Chang, “Virtual coach: the next tool in functional endoscopic sinus surgery education,” in *International forum of allergy & rhinology*, Wiley Online Library, 2020, pp. 97–102.

[31] A. K. Bell, M. B. Saide, J. T. Johanas, G. G. Leisk, S. D. Schwaitzberg, and C. G. L. Cao, “Innovative dynamic minimally invasive training environment (DynaMITE),” *Surgical innovation*, vol. 14, no. 3, pp. 217–224, 2007.

[32] A. K. Bell, M. Zhou, S. D. Schwaitzberg, and C. G. L. Cao, “Using a dynamic training environment to acquire laparoscopic surgery skill,” *Surgical endoscopy*, vol. 23, no. 10, pp. 2356–2363, 2009.

[33] L. A. Haveran *et al.*, "Optimizing laparoscopic task efficiency: the role of camera and monitor positions," *Surgical endoscopy*, vol. 21, no. 6, pp. 980–984, 2007.

[34] R. M. Leung, J. Leung, A. Vescan, A. Dubrowski, and I. Witterick, "Construct validation of a low-fidelity endoscopic sinus surgery simulator," *American journal of rhinology*, vol. 22, no. 6, pp. 642–648, 2008.

[35] M. Bajka, S. Tuchschmid, M. Streich, D. Fink, G. Székely, and M. Harders, "Evaluation of a new virtual-reality training simulator for hysteroscopy," *Surgical endoscopy*, vol. 23, no. 9, pp. 2026–2033, 2009.

[36] T. Horeman, S. P. Rodrigues, F.-W. Jansen, J. Dankelman, and J. J. van den Dobbelsteen, "Force measurement platform for training and assessment of laparoscopic skills," *Surgical endoscopy*, vol. 24, no. 12, pp. 3102–3108, 2010.

[37] T. Horeman, S. P. Rodrigues, F. W. Jansen, J. Dankelman, and J. J. van den Dobbelsteen, "Force parameters for skills assessment in laparoscopy," *IEEE Transactions on Haptics*, vol. 5, no. 4, pp. 312–322, 2011.

[38] S. Yamaguchi *et al.*, "Objective assessment of laparoscopic suturing skills using a motion-tracking system," *Surgical endoscopy*, vol. 25, no. 3, pp. 771–775, 2011.

[39] J. B. Pagador *et al.*, "Decomposition and analysis of laparoscopic suturing task using tool-motion analysis (TMA): improving the objective assessment," *International journal of computer assisted radiology and surgery*, vol. 7, no. 2, pp. 305–313, 2012.

[40] J. Varas *et al.*, "Significant transfer of surgical skills obtained with an advanced laparoscopic training program to a laparoscopic jejunojejunostomy in a live porcine model: feasibility of learning advanced laparoscopy in a general surgery residency," *Surgical endoscopy*, vol. 26, no. 12, pp. 3486–3494, 2012.

[41] V. S. Arikatla *et al.*, "Face and construct validation of a virtual peg transfer simulator," *Surgical endoscopy*, vol. 27, no. 5, pp. 1721–1729, 2013.

[42] R. Hirayama *et al.*, "Training to acquire psychomotor skills for endoscopic endonasal surgery using a personal webcam trainer," *Journal of neurosurgery*, vol. 118, no. 5, pp. 1120–1126, 2013.

[43] E. F. Hofstad, C. Våpenstad, M. K. Chmarra, T. Langø, E. Kuhry, and R. Mårvik, "A study of psychomotor skills in minimally invasive surgery: what differentiates expert and nonexpert performance," *Surgical endoscopy*, vol. 27, no. 3, pp. 854–863, 2013.

[44] E. F. Hofstad, C. Våpenstad, L. E. Bø, T. Langø, E. Kuhry, and R. Mårvik, "Psychomotor skills assessment by motion analysis in minimally invasive surgery on an animal organ," *Minimally Invasive Therapy & Allied Technologies*, vol. 26, no. 4, pp. 240–248, 2017.

[45] C. Loukas, C. Rouseas, and E. Georgiou, "The role of hand motion connectivity in the performance of laparoscopic procedures on a virtual reality simulator," *Medical & biological engineering & computing*, vol. 51, no. 8, pp. 911–922, 2013.

[46] P. J. van Empel *et al.*, "Learning curve on the TrEndo laparoscopic simulator compared to an expert level," *Surgical endoscopy*, vol. 27, no. 8, pp. 2934–2939, 2013.

[47] A. Sánchez *et al.*, "Laparoscopic surgery skills evaluation: analysis based on accelerometers," *JSLS: Journal of the Society of Laparoendoscopic Surgeons*, vol. 18, no. 4, 2014.

[48] M. Uemura *et al.*, "Analysis of hand motion differentiates expert and novice surgeons," *journal of surgical research*, vol. 188, no. 1, pp. 8–13, 2014.

[49] M. Uemura *et al.*, "Procedural surgical skill assessment in laparoscopic training environments," *International journal of computer assisted radiology and surgery*, vol. 11, no. 4, pp. 543–552, 2016.

[50] M. Uemura *et al.*, "Feasibility of an AI-Based Measure of the Hand Motions of Expert and Novice Surgeons," *Computational and Mathematical Methods in Medicine*, vol. 2018, 2018.

[51] D. Xiao, J. J. Jakimowicz, A. Albayrak, S. N. Buzink, S. M. B. I. Botden, and R. H. M. Goossens, "Face, content, and construct validity of a novel portable ergonomic simulator for basic laparoscopic skills," *Journal of surgical education*, vol. 71, no. 1, pp. 65–72, 2014.

[52] F. P. Escamirosa, R. M. O. Flores, I. O. García, C. R. Z. Vidal, and A. M. Martínez, "Face, content, and construct validity of the EndoViS training system for objective assessment of psychomotor skills of laparoscopic surgeons," *Surgical endoscopy*, vol. 29, no. 11, pp. 3392–3403, 2015.

[53] V. Lahanas, C. Loukas, N. Smailis, and E. Georgiou, "A novel augmented reality simulator for skills assessment in minimal invasive surgery," *Surgical endoscopy*, vol. 29, no. 8, pp. 2224–2234, 2015.

[54] M. S. R. Prasad, M. Manivannan, and S. M. Chandramohan, "Effects of laparoscopic instrument and finger on force perception: a first step towards laparoscopic force-skills training," *Surgical endoscopy*, vol. 29, no. 7, pp. 1927–1943, 2015.

[55] G. Saggio *et al.*, "Objective surgical skill assessment: An initial experience by means of a sensory glove paving the way to open surgery simulation?," *Journal of surgical education*, vol. 72, no. 5, pp. 910–917, 2015.

[56] Y. Watanabe *et al.*, "Camera navigation and cannulation: validity evidence for new educational tasks to complement the Fundamentals of Laparoscopic Surgery Program," *Surgical endoscopy*, vol. 29, no. 3, pp. 552–557, 2015.

[57] D. Berger-Richardson *et al.*, "Description and preliminary evaluation of a low-cost simulator for training and evaluation of flexible endoscopic skills," *Surgical innovation*, vol. 23, no. 2, pp. 183–188, 2016.

[58] B. A. Fransson, C. Chen, J. A. Noyes, and C. A. Ragle, "Instrument motion metrics for laparoscopic skills assessment in virtual reality and augmented reality," *Veterinary surgery*, vol. 45, no. S1, pp. O5–O13, 2016.

[59] B. Genovese *et al.*, "Surgical hand tracking in open surgery using a versatile motion sensing system: are we there yet?," *The American Surgeon*, vol. 82, no. 10, pp. 872–875, 2016.

[60] M. A. Farcas, M. O. Trudeau, A. Nasr, J. T. Gerstle, B. Carrillo, and G. Azzie, "Analysis of motion in laparoscopy: the deconstruction of an intra-corporeal suturing task," *Surgical endoscopy*, vol. 31, no. 8, pp. 3130–3139, 2017.

[61] S. Choussein *et al.*, "Robotic assistance confers ambidexterity to laparoscopic surgeons," *Journal of minimally invasive gynecology*, vol. 25, no. 1, pp. 76–83, 2018.

[62] M. Schijven and J. Jakimowicz, "Construct validity," *Surgical Endoscopy and Other Interventional Techniques*, vol. 17, no. 5, pp. 803–810, 2003.

[63] F. Cavallo, G. Megali, S. Sinigaglia, O. Tonet, P. Dario, and A. Pietrabissa, "A biomechanical analysis of bi-manual coordination and depth perception in virtual laparoscopic surgery," in *Computer Assisted Radiology and Surgery (CARS)-10th Annual Conference of the International Society for Computer Aided Surgery*, 2006.

[64] M. Wilson, J. McGrath, S. Vine, J. Brewer, D. Defriend, and R. Masters, "Psychomotor control in a virtual laparoscopic surgery training environment: gaze control parameters differentiate novices from experts," *Surgical endoscopy*, vol. 24, no. 10, pp. 2458–2464, 2010.

[65] C. Loukas and E. Georgiou, "Multivariate autoregressive modeling of hand kinematics for laparoscopic skills assessment of surgical trainees," *IEEE transactions on biomedical engineering*, vol. 58, no. 11, pp. 3289–3297, 2011.

[66] J.-M. Luursema, M. M. Rovers, M. Groenier, and H. van Goor, "Performance variables and professional experience in simulated laparoscopy: a two-group learning curve study," *Journal of surgical education*, vol. 71, no. 4, pp. 568–573, 2014.

[67] V. Lahanas, C. Loukas, K. Georgiou, H. Lababidi, and D. Al-Jaroudi, "Virtual reality-based assessment of basic laparoscopic skills using the Leap Motion controller," *Surgical endoscopy*, vol. 31, no. 12, pp. 5012–5023, 2017.

[68] B. M. Finnerty, C. Afaneh, A. Aronova, T. J. Fahey, and R. Zarnegar, "General surgery training and robotics: Are residents improving their skills?," *Surgical endoscopy*, vol. 30, no. 2, pp. 567–573, 2016.

[69] N. E. Cagiltay, E. Ozcelik, G. Sengul, and M. Berker, "Construct and face validity of the educational computer-based environment (ECE) assessment scenarios for basic endoneurosurgery skills," *Surgical endoscopy*, vol. 31, no. 11, pp. 4485–4495, 2017.

[70] N. E. Cagiltay *et al.*, "The effect of training, used-hand, and experience on endoscopic surgery skills in an educational computer-based simulation environment (ECE) for endoneurosurgery training," *Surgical innovation*, vol. 26, no. 6, pp. 725–737, 2019.

[71] J. A. Sánchez-Margallo and F. M. Sánchez-Margallo, "Initial experience using a robotic-driven laparoscopic needle holder with ergonomic handle: assessment of surgeons' task performance and ergonomics," *International journal of computer assisted radiology and surgery*, vol. 12, no. 12, pp. 2069–2077, 2017.

[72] G. G. Menekse Dalveren and N. E. Cagiltay, "Insights from surgeons' eye-movement data in a virtual simulation surgical training environment: effect of experience level and hand conditions," *Behaviour & Information Technology*, vol. 37, no. 5, pp. 517–537, 2018.

[73] R. Prasad, M. Muniyandi, G. Manoharan, and S. M. Chandramohan, "Face and Construct Validity of a Novel Virtual Reality–Based Bimanual Laparoscopic Force-Skills Trainer With Haptics Feedback," *Surgical innovation*, vol. 25, no. 5, pp. 499–514, 2018.

[74] D. Topalli and N. E. Cagiltay, "Eye-hand coordination patterns of intermediate and novice surgeons in a simulation-based endoscopic surgery training environment," *Journal of Eye Movement Research*, vol. 11, no. 6, 2018.

[75] D. Topalli and N. E. Cagiltay, "Classification of intermediate and novice surgeons' skill assessment through performance metrics," *Surgical innovation*, vol. 26, no. 5, pp. 621–629, 2019.

[76] J. Farmer *et al.*, "Systematic approach for content and construct validation: Case studies for arthroscopy and laparoscopy," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 16, no. 4, p. e2105, 2020.

[77] M. K. O'Malley, M. D. Byrne, S. Estrada, C. Duran, D. Schulz, and J. Bismuth, "Expert surgeons can smoothly control robotic tools with a discrete control interface," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 388–394, 2019.

[78] A. A. Yilbas *et al.*, "The effect of playing video games on fiberoptic intubation skills," *Anaesthesia Critical Care & Pain Medicine*, vol. 38, no. 4, pp. 341–345, 2019.

[79] C. W. Ashley, K. Donaldson, K. M. Evans, B. Nielsen, and E. N. Everett, "Surgical cross-training with surgery naive learners: implications for resident training," *Journal of surgical education*, vol. 76, no. 6, pp. 1469–1475, 2019.

[80] S. Sadeghnejad *et al.*, "A validation study of a virtual-based haptic system for endoscopic sinus surgery training," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 15, no. 6, p. e2039, 2019.

[81] A. N. Karabanov *et al.*, "Getting to grips with endoscopy-Learning endoscopic surgical skills induces bi-hemispheric plasticity of the grasping network," *NeuroImage*, vol. 189, pp. 32–44, 2019.

[82] M. Silvennoinen, J.-P. Mecklin, P. Saariluoma, and T. Antikainen, "Expertise and skill in minimally invasive surgery," *Scandinavian Journal of Surgery*, vol. 98, no. 4, pp. 209–213, 2009.

[83] "Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper - Cisco," Jun. 02, 2021. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed Jun. 14, 2021).

[84] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, Jul. 1959.

[85] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, "AN OVERVIEW OF MACHINE LEARNING," in *Machine Learning*, Elsevier, 1983, pp. 3–23.

[86] D. Poole *et al.*, "Computational Intelligence A Logical Approach," 1998. Accessed: Jun. 13, 2021. [Online]. Available: http://www.oup-usa.org

[87] D. Wang, “Unsupervised Learning: Foundations of Neural Computation,” Jun. 2001. Accessed: Jun. 13, 2021. [Online]. Available: https://ojs.aaai.org/index.php/aimagazine/article/view/1565

[88] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[89] H. Almuallim, “An efficient algorithm for optimal pruning of decision trees,” *Artificial Intelligence*, vol. 83, no. 2, pp. 347–362, Jun. 1996.

[90] B. Settles, “Active Learning Literature Survey,” Computer Sciences Department, 2009. Accessed: Jun. 13, 2021. [Online]. Available: https://minds.wisconsin.edu/handle/1793/60660

[91] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, “Learning to Generalize: Meta-Learning for Domain Generalization,” Apr. 2018. Accessed: Jun. 13, 2021. [Online]. Available: www.aaai.org

[92] Q. Li, Z. Han, and X.-M. Wu, “Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning,” Apr. 2018. Accessed: Jun. 13, 2021. [Online]. Available: www.aaai.org

[93] S. Bozinovski, “Modeling mechanisms of cognition-emotion interaction in artificial neural networks, since 1981,” in *Procedia Computer Science*, Elsevier B.V., Jan. 2014, pp. 255–263.

[94] S. C. Gadanho and J. Hallam, “Robot learning driven by emotions,” *Adaptive Behavior*, vol. 9, no. 1, pp. 42–64, Jul. 2001.

[95] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, “Self-taught learning: Transfer learning from unlabeled data,” in *ACM International Conference Proceeding Series*, 2007, pp. 759–766.

[96] J. Schmidhuber, “Deep Learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015.

[97] “AlphaGo | DeepMind,” May 27, 2021. https://deepmind.com/research/case-studies/alphago-the-story-so-far (accessed Jun. 13, 2021).

[98] P.-N. Tan, V. Kumar, and J. Srivastava, “Selecting the Right Interestingness Measure for Association Patterns,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’02*, New York, New York, USA: ACM Press, 2002.

[99] R. Urbanowicz and JH Moore, “Learning classifier systems: a complete introduction, review, and roadmap,” *Journal of Artificial Evolution and*

Applications, 2009, Accessed: Jun. 13, 2021. [Online]. Available: https://www.hindawi.com/archive/2009/736398/abs/

[100] P. Comon, "Independent Component Analysis," *hal.archives-ouvertes.fr*, pp. 29–38, Jun. 1992, Accessed: Jun. 13, 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00346684

[101] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[102] M. D. Ritchie, L. W. Hahn, and J. H. Moore, "Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity," *Genetic Epidemiology*, vol. 24, no. 2, pp. 150–157, Feb. 2003.

[103] J. H. Moore *et al.*, "A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility," *Journal of Theoretical Biology*, vol. 241, no. 2, pp. 252–261, Jul. 2006.

[104] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[105] "What is NumPy? — NumPy v1.24 Manual," *What is NumPy? — NumPy v1.24 Manual*, Jun. 10, 2023. https://numpy.org/doc/stable/user/whatisnumpy.html (accessed Jun. 10, 2023).

[106] "Broadcasting — NumPy v1.24 Manual," Jun. 10, 2023. https://numpy.org/doc/stable/user/basics.broadcasting.html#basics-broadcasting (accessed Jun. 10, 2023).

[107] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, "Cython: The Best of Both Worlds," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 31–39, Mar. 2011.

[108]: : "Anaconda.org," :: *Anaconda.org*, Jun. 10, 2023. https://anaconda.org/ (accessed Jun. 10, 2023).

[109] "Project Jupyter," *Project Jupyter*, Jun. 10, 2023. https://jupyter.org (accessed Jun. 10, 2023).

[110] W. Mckinney, "pandas: a Foundational Python Library for Data Analysis and Statistics," *Python High Performance Science Computer*, Jan. 2011.

[111] "NumPy," *NumPy*, Jun. 10, 2023. https://numpy.org/ (accessed Jun. 10, 2023).

[112] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[113] W. R. Hamilton, “XI. On quaternions; or on a new system of imaginaries in algebra,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 33, no. 219, pp. 58–60, Jul. 1848.

[114] “sklearn.ensemble.RandomForestClassifier,” *scikit-learn*, Jun. 10, 2023. https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed Jun. 13, 2023).

APPENDICES

APPENDIX A. Full Dataset Previews

Figure A.1. PositionalDF Preview

```
PositionalDF
```

	AVx	AVy	AVz	AAx	AAy	AAz	RV	Skill
0	1.343041e+06	1.373690e+06	1.363473e+06	92105100.0	9.420698e+07	93506350.0	2.355810e+06	1
1	-1.653294e+03	0.000000e+00	0.000000e+00	-97005740.0	-9.909747e+07	-98360470.0	1.653294e+03	1
2	-1.059633e+04	-2.119395e+03	0.000000e+00	-500685400.0	-5.088864e+08	-504323600.0	1.080620e+04	1
3	-1.229823e+03	0.000000e+00	0.000000e+00	-144283500.0	-1.474411e+08	-146344500.0	1.229823e+03	1
4	-1.199958e+03	0.000000e+00	0.000000e+00	-140755000.0	-1.438386e+08	-142768900.0	1.199958e+03	1
...	...	...	...	...	...	...	...	...
228525	7.864220e+01	0.000000e+00	0.000000e+00	-1029652.0	-1.797018e+04	-1037557.0	7.864220e+01	0
228526	5.740965e+01	5.739213e+01	-1.148193e+02	-1503527.0	-2.566182e+04	-1516006.0	1.406172e+02	0
228527	0.000000e+00	-5.655774e+01	5.657500e+01	-1482235.0	-2.641382e+04	-1492273.0	7.999693e+01	0
228528	0.000000e+00	5.819283e+01	1.164212e+02	-1525087.0	-2.601170e+04	-1534807.0	1.301549e+02	0
228529	0.000000e+00	1.057793e+02	-6.346436e+02	-2771362.0	-4.638953e+04	-2802890.0	6.433986e+02	0

228530 rows × 8 columns

Figure A.2. FullPV Preview

```
fullPV
```

	Skill	PartWStd	PartWMean	PartWVar	logs_div	logs_subt
0	1	NaN	50.345527	NaN	NaN	NaN
1	1	1.848054	51.652299	3.415303	0.950650	-0.000720
2	1	126.957238	124.947211	16118.140200	0.195034	-0.011158
3	1	106.197663	113.409047	11277.943727	3.446140	0.006613
4	1	93.410319	106.101108	8725.487674	1.025045	0.000233
...	...	...	...	...	...	...
227031	0	18.703862	8.625766	349.834471	0.996915	-0.000310
227032	0	18.694416	8.626675	349.481206	0.988721	-0.001130
227033	0	18.685157	8.629386	349.135110	0.842217	-0.015630
227034	0	18.675737	8.630227	348.783150	1.195134	0.016280
227035	0	18.666379	8.628634	348.433718	1.342393	0.034140

227036 rows × 6 columns

Figure A.3. FullPT Preview

```
fullPT
```

	Skill	SecWstd	SecWmean	SecWvar
0	1	NaN	50.345527	NaN
1	1	1.848054	51.652299	3.415303
2	1	126.957238	124.947211	16118.140200
3	1	106.197663	113.409047	11277.943727
4	1	93.410319	106.101108	8725.487674
...	...	...	...	...
227031	0	0.060052	9.428813	0.003606
227032	0	0.067598	9.448125	0.004569
227033	0	0.762460	9.758418	0.581346
227034	0	0.704910	9.716255	0.496899
227035	0	1.146657	9.382918	1.314822

227036 rows × 4 columns

Figure A.4. Full1SV Preview

```
full1SV
```

	Skill	PartDurWsum	PartDurWstd	PartDurWmean	PartDurWvar	logs_div	logs_subt
0	1	0.014582	NaN	0.014582	NaN	NaN	NaN
1	1	0.028444	0.000509	0.014222	2.589164e-07	0.950650	-0.000720
2	1	0.031147	0.006660	0.010382	4.435273e-05	0.195034	-0.011158
3	1	0.040464	0.005464	0.010116	2.985232e-05	3.446140	0.006613
4	1	0.050014	0.004738	0.010003	2.245327e-05	1.025045	0.000233
...	...	...	...	...	...	...	...
227031	0	98.830810	0.018826	0.100031	3.544031e-04	0.996915	-0.000310
227032	0	98.929870	0.018816	0.100030	3.540454e-04	0.988721	-0.001130
227033	0	99.013300	0.018814	0.100013	3.539657e-04	0.842217	-0.015630
227034	0	99.113010	0.018804	0.100013	3.536083e-04	1.195134	0.016280
227035	0	99.246860	0.018826	0.100047	3.544056e-04	1.342393	0.034140

227036 rows × 7 columns

Figure A.5. Full1ST Preview

```
full1ST
```

	Skill	TimeDursum	TimeDurstd	TimeDurmean	TimeDurvar
0	1	0.014582	NaN	0.014582	NaN
1	1	0.028444	0.000509	0.014222	2.589164e-07
2	1	0.031147	0.006660	0.010382	4.435273e-05
3	1	0.040464	0.005464	0.010116	2.985232e-05
4	1	0.050014	0.004738	0.010003	2.245327e-05
...	...	...	...	...	...
227031	0	0.400310	0.000634	0.100078	4.018250e-07
227032	0	0.499370	0.000713	0.099874	5.084300e-07
227033	0	0.582800	0.006743	0.097133	4.547427e-05
227034	0	0.682510	0.006232	0.097501	3.884368e-05
227035	0	0.816360	0.014087	0.102045	1.984469e-04

227036 rows × 5 columns

Figure A.6. FullTV Preview

```
fullTV
```

	Skill	TaskWStd	TaskWMean	TaskWVar	logs_div	logs_subt
0	1	NaN	50.345527	NaN	NaN	NaN
1	1	1.848054	51.652299	3.415303	0.950650	-0.000720
2	1	126.957238	124.947211	16118.140200	0.195034	-0.011158
3	1	106.197663	113.409047	11277.943727	3.446140	0.006613
4	1	93.410319	106.101108	8725.487674	1.025045	0.000233
...	...	...	...	...	...	...
227031	0	0.992842	9.459419	0.985736	0.996915	-0.000310
227032	0	0.980712	9.460990	0.961797	0.988721	-0.001130
227033	0	1.009155	9.503987	1.018395	0.842217	-0.015630
227034	0	0.997371	9.503062	0.994749	1.195134	0.016280
227035	0	1.051623	9.448540	1.105911	1.342393	0.034140

227036 rows × 6 columns

Figure A.7. FullTT Preview

```
fullTT
```

	Skill	SecTaskWStd	SecTaskWMean	SecTaskWVar
0	1	NaN	50.345527	NaN
1	1	1.848054	51.652299	3.415303
2	1	126.957238	124.947211	16118.140200
3	1	106.197663	113.409047	11277.943727
4	1	93.410319	106.101108	8725.487674
...	...	...	...	...
227031	0	0.386956	9.336935	0.149735
227032	0	0.376041	9.349498	0.141407
227033	0	0.610061	9.472022	0.372174
227034	0	0.590693	9.471508	0.348918
227035	0	0.808871	9.336955	0.654273

227036 rows × 4 columns

Figure A.8. Full2SV Preview

```
full2SV
```

	Skill	TaskDurWSum	TaskDurWstd	TaskDurWmean	TaskDurWvar	logs_div	logs_subt
0	1	0.014582	NaN	0.014582	NaN	NaN	NaN
1	1	0.028444	0.000509	0.014222	2.589164e-07	0.950650	-0.000720
2	1	0.031147	0.006660	0.010382	4.435273e-05	0.195034	-0.011158
3	1	0.040464	0.005464	0.010116	2.985232e-05	3.446140	0.006613
4	1	0.050014	0.004738	0.010003	2.245327e-05	1.025045	0.000233
...	...	...	...	...	...	...	...
227031	0	4.130454	0.010280	0.100743	1.056838e-04	0.996915	-0.000310
227032	0	4.229514	0.010157	0.100703	1.031735e-04	0.988721	-0.001130
227033	0	4.312945	0.010376	0.100301	1.076553e-04	0.842217	-0.015630
227034	0	4.412655	0.010255	0.100288	1.051597e-04	1.195134	0.016280
227035	0	4.546504	0.011305	0.101033	1.278016e-04	1.342393	0.034140

227036 rows × 7 columns

Figure A.9. Full2ST Preview

```
full2ST
```

	Skill	SecTaskWSum	SecTaskWstd	SecTaskWmean	SecTaskWvar
0	1	0.014582	NaN	0.014582	NaN
1	1	0.028444	0.000509	0.014222	2.589164e-07
2	1	0.031147	0.006660	0.010382	4.435273e-05
3	1	0.040464	0.005464	0.010116	2.985232e-05
4	1	0.050014	0.004738	0.010003	2.245327e-05
...	...	...	...	...	...
227031	0	1.417370	0.004721	0.101241	2.228451e-05
227032	0	1.516430	0.004584	0.101095	2.100980e-05
227033	0	1.599860	0.006254	0.099991	3.911315e-05
227034	0	1.699570	0.006056	0.099975	3.667323e-05
227035	0	1.833420	0.009913	0.101857	9.826795e-05

227036 rows × 5 columns

Figure A.10. FullComplete Preview

`fullComplete`

	Skill	TaskWStd	TaskWMean	TaskWVar	TaskDurWSum	TaskDurWstd	TaskDurWmean	TaskDurWvar	SecTaskWvelStd	SecTaskWvelMean	...	PartDurWstd	PartDurWmean	PartDurWvar	TimeDursum	TimeDurstd	TimeDurmean	TimeDurvar	SecWstd	SecWmean	SecWvar
0	1	NaN	50.345527	NaN	0.014582	NaN	0.014582	NaN	NaN	50.345527	...	NaN	0.014582	NaN	0.014582	NaN	0.014582	NaN	NaN	50.345527	NaN
1	1	1.848054	51.652299	3.415303	0.028444	0.000509	0.014222	2.589164e-07	1.848054	51.652299	...	0.000509	0.014222	2.589164e-07	0.028444	0.000509	0.014222	2.589164e-07	1.848054	51.652299	3.415303
2	1	126.957238	124.947211	16118.140200	0.031147	0.006660	0.010382	4.435273e-05	126.957238	124.947211	...	0.006660	0.010382	4.435273e-05	0.031147	0.006660	0.010382	4.435273e-05	126.957238	124.947211	16118.140200
3	1	106.197663	113.409047	11277.943727	0.040464	0.005464	0.010116	2.985232e-05	106.197663	113.409047	...	0.005464	0.010116	2.985232e-05	0.040464	0.005464	0.010116	2.985232e-05	106.197663	113.409047	11277.943727
4	1	93.410319	106.101108	8725.487674	0.050014	0.004738	0.010003	2.245327e-05	93.410319	106.101108	...	0.004738	0.010003	2.245327e-05	0.050014	0.004738	0.010003	2.245327e-05	93.410319	106.101108	8725.487674
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
227031	0	0.992842	9.459419	0.985736	4.130454	0.010280	0.100743	1.056838e-04	0.386956	9.336935	...	0.018826	0.100031	3.544031e-04	0.400310	0.000634	0.100078	4.018250e-07	0.060052	9.428813	0.003606
227032	0	0.980712	9.460990	0.961797	4.229514	0.010157	0.100703	1.031735e-04	0.376041	9.349498	...	0.018816	0.100030	3.540454e-04	0.499370	0.000713	0.099874	5.084300e-07	0.067598	9.448125	0.004569
227033	0	1.009155	9.503987	1.018395	4.312945	0.010376	0.100301	1.076553e-04	0.610061	9.472022	...	0.018814	0.100013	3.539657e-04	0.582800	0.006743	0.097133	4.547427e-05	0.762460	9.758418	0.581346
227034	0	0.997371	9.503062	0.994749	4.412655	0.010255	0.100288	1.051597e-04	0.590693	9.471508	...	0.018804	0.100013	3.536083e-04	0.682510	0.006232	0.097501	3.884368e-05	0.704910	9.716255	0.496899
227035	0	1.051623	9.448540	1.105911	4.546504	0.011305	0.101033	1.278016e-04	0.808871	9.336955	...	0.018826	0.100047	3.544056e-04	0.816360	0.014087	0.102045	1.984469e-04	1.146657	9.382918	1.314822

227036 rows × 31 columns

APPENDIX B. Confusion Matrix Heat Map

Figure B.1. Confusion Matrix - 1

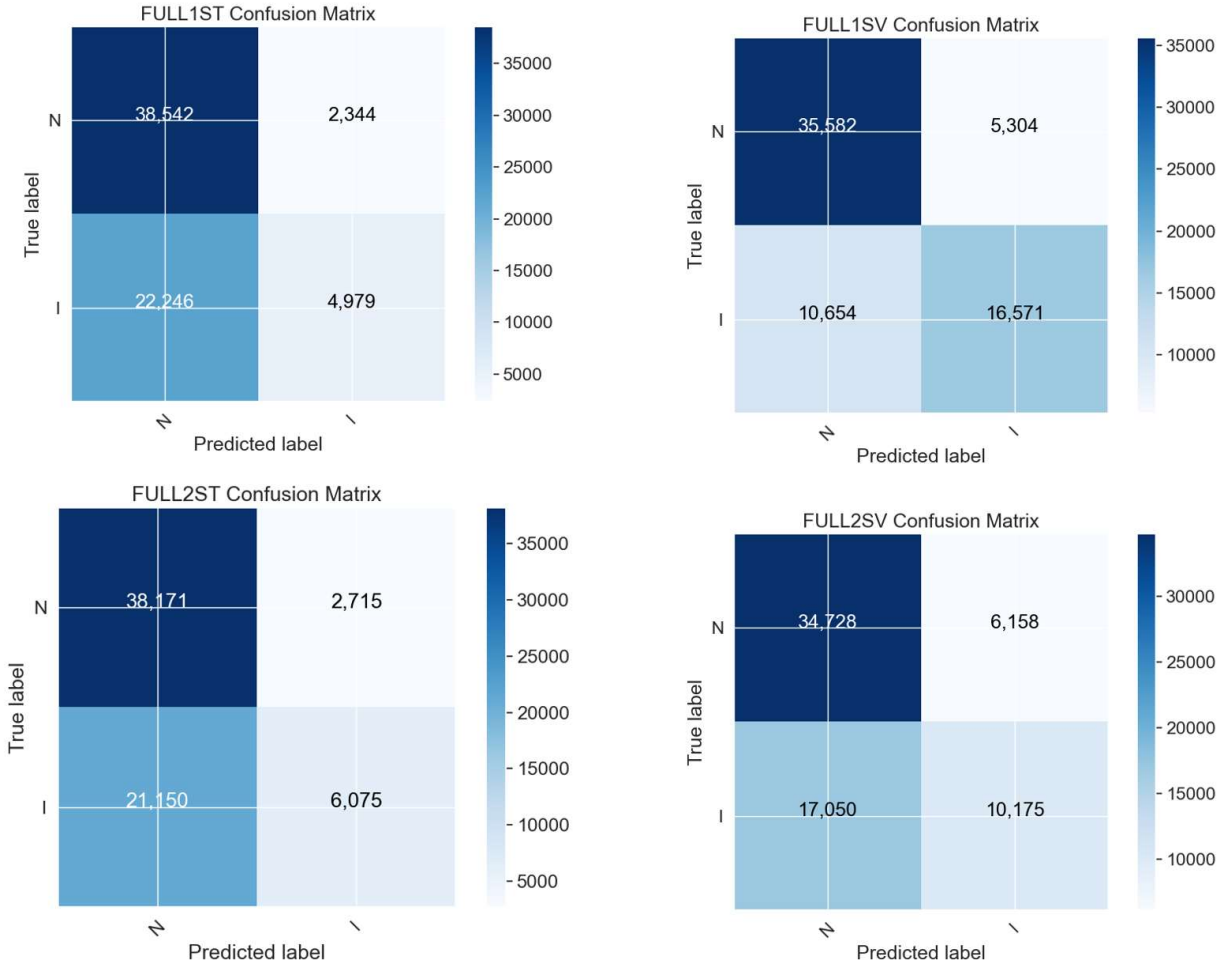


Figure B.2. Confusion Matrix - 2

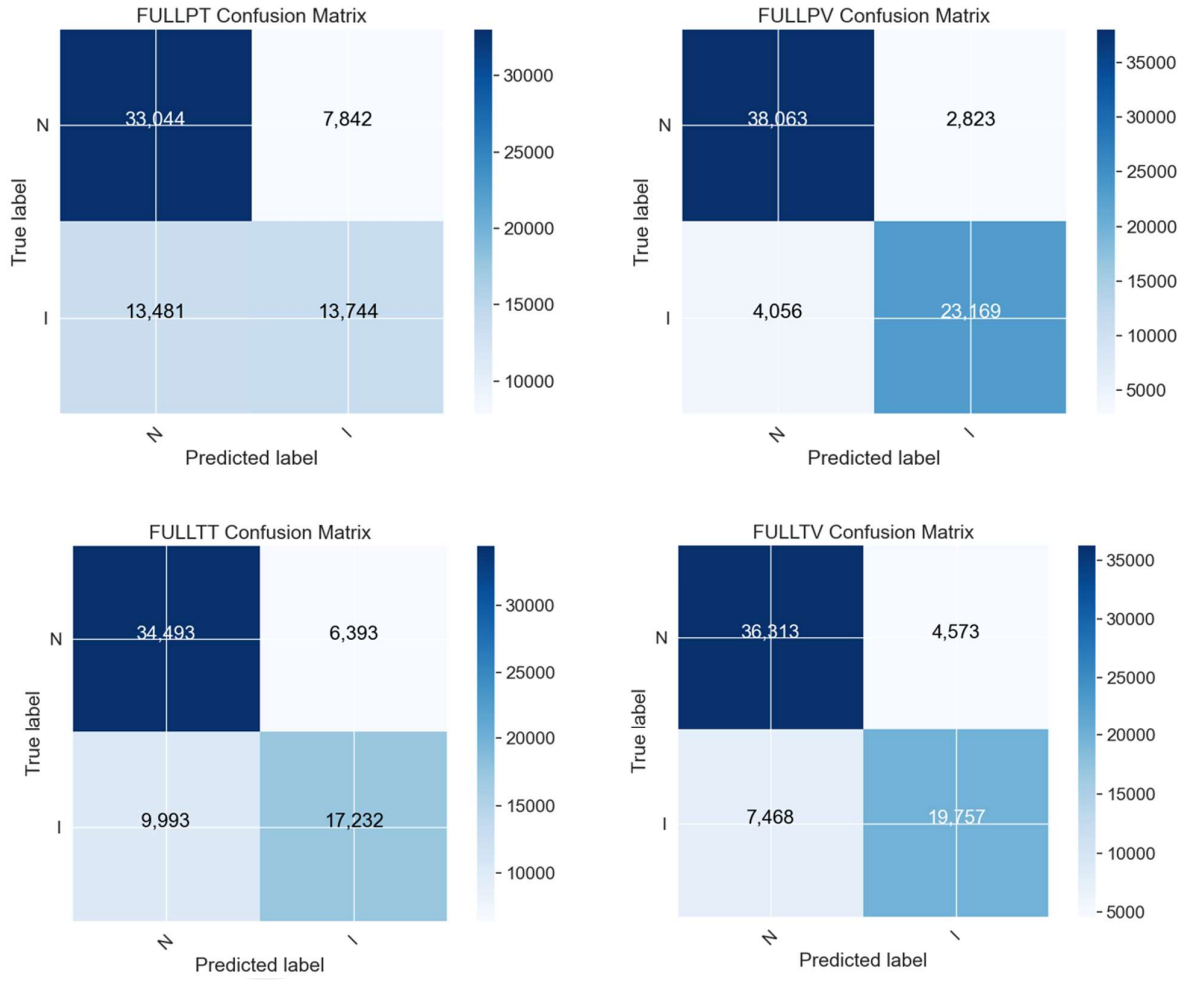
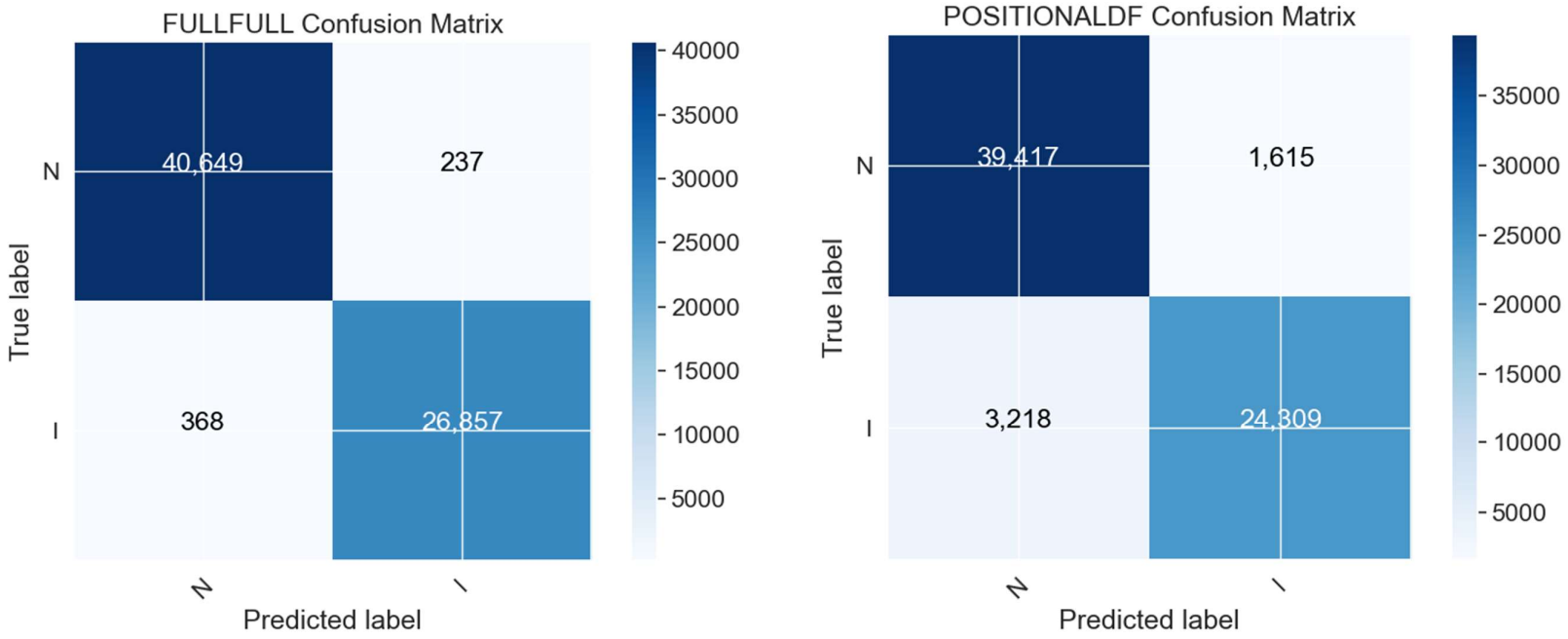


Figure B.2. Confusion Matrix - 3



APPENDIX C. Feature Importance

Figure C.1. Feature Importance - 1

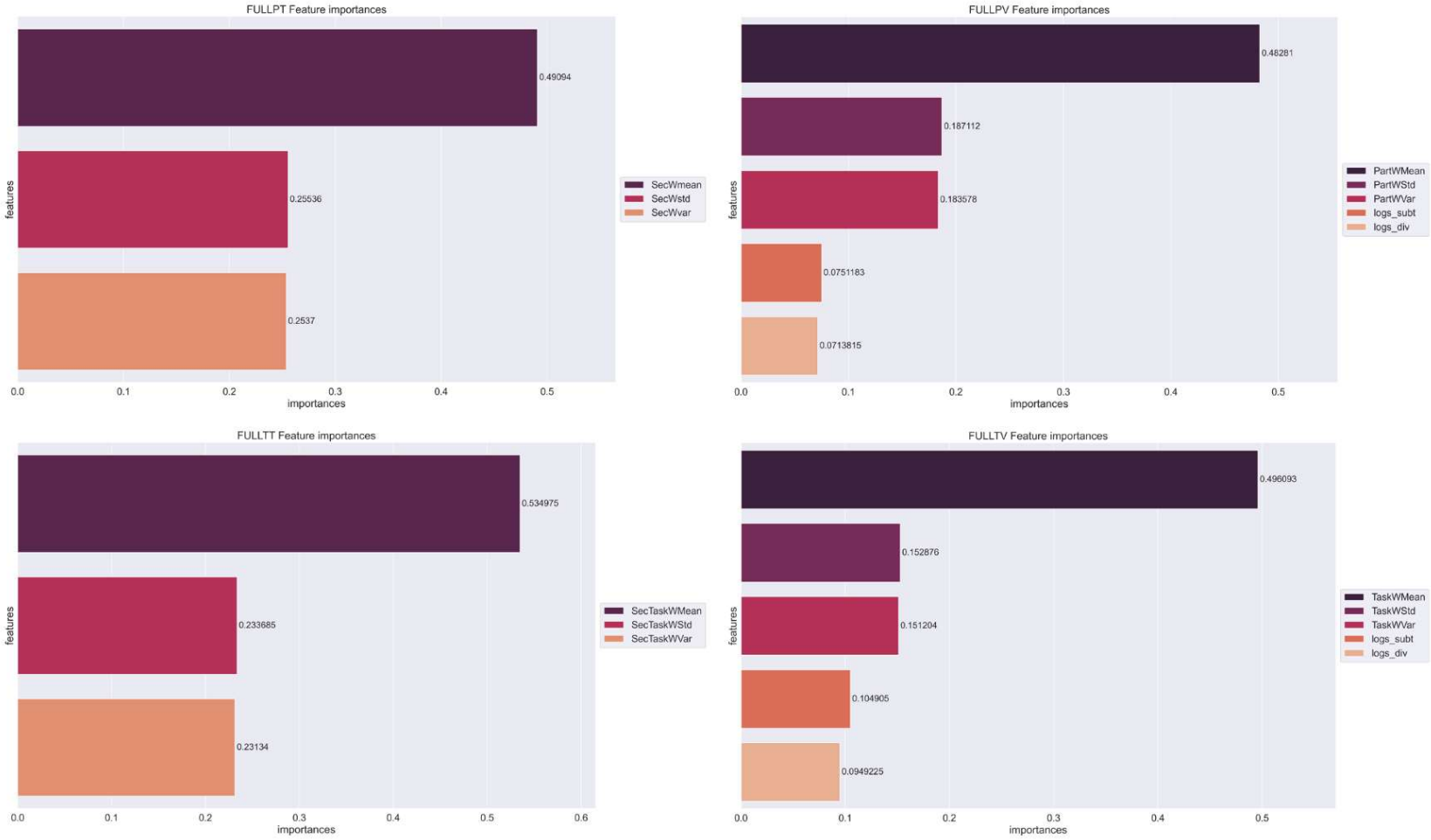


Figure C.2. Feature Importance - 2

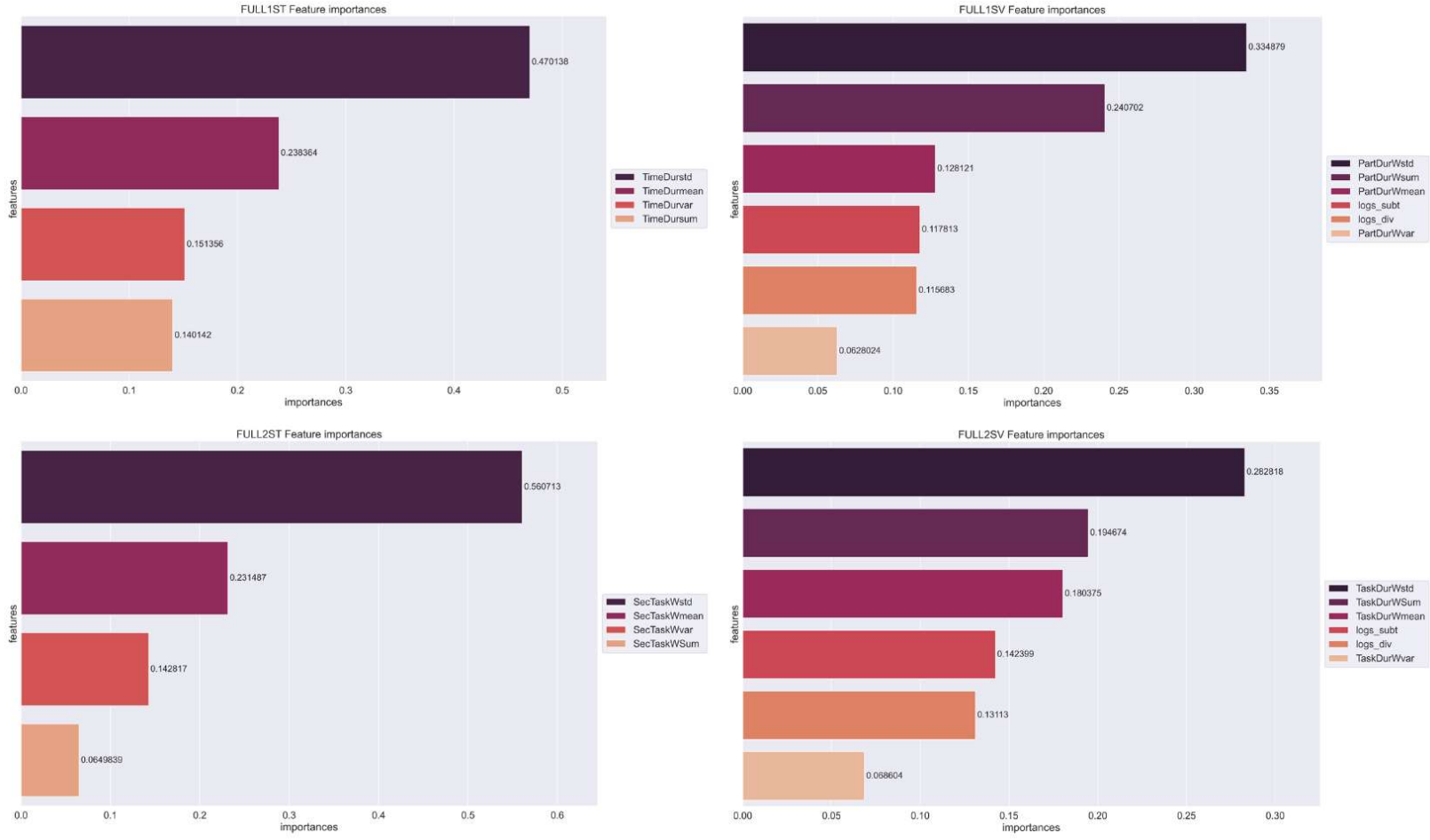


Figure C.3. Feature Importance - 3