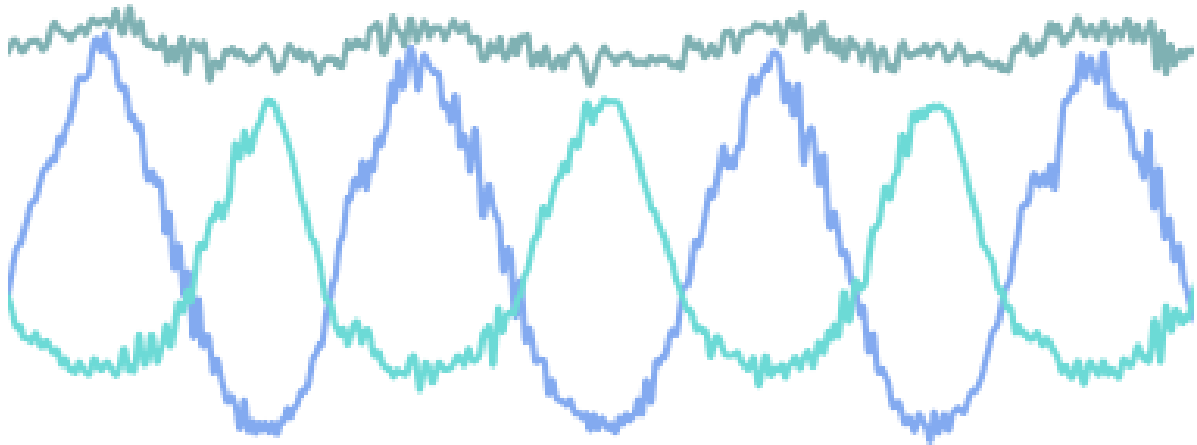




CHALMERS
UNIVERSITY OF TECHNOLOGY



Feature Extraction and Classification of Knee Motions Using a Sensor-Equipped Orthosis

Sensor-Based Motion Analysis of Controlled Knee and Hip Movements

Master's thesis in Biomedical Engineering

ALICIA BENGTSSON
HUGO BURSELL PALM

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

MASTER'S THESIS 2026

Feature Extraction and Classification of Knee Motions Using a Sensor-Equipped Orthosis

Sensor-Based Motion Analysis of Controlled Knee and Hip Movements

ALICIA BENGTSSON
HUGO BURSELL PALM



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical engineering
Division of Systems and Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Feature Extraction and Classification of Knee Motions Using a Sensor-Equipped
Orthosis

Sensor-Based Motion Analysis of Controlled Knee and Hip Movements

ALICIA BENGTSSON

HUGO BURSELL PALM

© ALICIA BENGTSSON, 2026.

© HUGO BURSELL PALM, 2026.

Supervisor: Fabian Just, Department of Biomedical Engineering, Universität Ulm

Examiner: Emmanuel Dean, Department of Electrical Engineering, Chalmers University of Technology

Master's Thesis 2026

Department of Electrical Engineering

Division of Systems and Control

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Cover: Acceleration data from IMU 2 during squats. The image has been included with informed consent from the study participant.

Typeset in L^AT_EX

Printed by Chalmers Reproservice

Gothenburg, Sweden 2026

Feature Extraction and Classification of Knee Motions Using a Sensor-Equipped Orthosis

Sensor-Based Motion Analysis of Controlled Knee and Hip Movements

ALICIA BENGTSSON

HUGO BURSELL PALM

Department of Electrical Engineering

Chalmers University of Technology

Abstract

This thesis investigates the feasibility of using a sensorized knee orthosis for movement classification and sensor-derived feature extraction during controlled lower-limb movements. The orthosis was equipped with two six-axis IMU sensors and an absolute capacitive rotary encoder. A movement data-collection study involving 24 healthy participants was conducted, in which four lower-limb movements and three intentionally simulated execution variations were recorded.

The collected data were used to train and evaluate three neural network architectures commonly applied to time-series classification: CNN-LSTM, LSTM-CNN, and a parallel architecture. The study is limited to controlled motion analysis using a sensorized wearable system and does not evaluate medical, diagnostic, therapeutic, rehabilitation, or clinical outcomes.

The results show that the sensor system provided stable sampling frequencies and data suitable for classification of four movements. The CNN-LSTM model achieved a mean leave-one-subject-out (LOSO) classification accuracy of $99.42 \pm 0.35\%$ for the predefined movement classes and more than 99% pseudo-live classification accuracy after post-processing. In addition, movement-related features such as range of motion, knee angle deviations, movement smoothness, and repetition count were extracted from the recorded sensor data. A modified CNN-LSTM architecture was also evaluated for classification of predefined execution categories, demonstrating the feasibility of distinguishing between intentionally defined movement variations.

The feature-extraction results should be regarded as exploratory because no external kinematic ground truth was available. Similarly, the execution-category classification represents predefined movement categories rather than a validated assessment of movement quality. Overall, the findings demonstrate the potential of wearable sensor systems and machine learning for controlled movement classification and sensor-derived movement analysis.

Keywords: knee motion, hip motion, sensorized orthosis, motion classification, offline classification, pseudo-live classification, feature extraction, jerk, knee angle deviations, range of motion

Acknowledgements

We would like to extend our gratitude to our supervisor Fabian Just, assistant supervisor Domenico Caliandro, and our examiner Emmanuel Dean for their guidance and support throughout all stages of this project. Their knowledge and technical expertise within the field of sensor-integrated systems and movement analysis has been instrumental for the design and execution of the project as a whole.

We would also like to thank Gunilla Kjellby Wendt for the contribution of her expertise within movement assessment when designing the study and the selection of movements and predefined execution variations.

In addition, we would also like to express our gratitude to the lab manager of CASE at Chalmers University of Technology, Rikard Karlsson, for his invaluable support when troubleshooting and upgrading the hardware of the system. We truly appreciate his patience, enthusiasm, and ability to encourage our engineering mindset.

Another person who was of great help was M.Sc student Robert Reinhard Schneider. He was of great help when developing the system software used for data collection and recordings.

Last but not least, we would like to thank everyone who participated in this study. It would not have been possible without you.

Alicia Bengtsson, Gothenburg, May 2026
Hugo Bursell Palm, Gothenburg, May 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

CNN	Convolutional Neural Network
EMG	Electromyography
IMU	Inertial Measurement Unit
LDLJ	Log Dimensionless Jerk
LOSO	Leave-One-Subject-Out
LSTM	Long Short-Term Memory
MCU	Microcontroller Unit
PCB	Printed Circuit Board
ROM	Range Of Motion
SPI	Serial Peripheral Interface
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Contents

List of Acronyms	ix
List of Figures	xv
List of Tables	xix
1 Introduction	1
1.1 Aim	2
1.2 Research Questions	2
1.3 Scope and Limitations	2
2 Theory	3
2.1 Sensorized Orthosis and Sensing Hardware	3
2.1.1 Technical Background	3
2.1.2 Orthosis Hardware Architecture	4
2.2 Time-Series Movement Classification	5
2.2.1 Traditional Approaches for Movement Classification	5
2.2.2 Deep Learning for Time-Series Data Classification	6
2.2.2.1 CNN	6
2.2.2.2 LSTM	6
2.2.2.3 Hybrid CNN-LSTM Architectures	6
2.3 Model Evaluation and LOSO	7
2.4 Sensor-Derived Movement Features	8
2.4.1 Range of Motion	9
2.4.2 Knee Alignment and Movement Deviation	9
2.4.3 Jerk	9
3 Methods	11
3.1 System Setup	11
3.2 Changes to the Hardware	12
3.3 Sampling Frequency Evaluation	13
3.4 Study Design	13
3.4.1 Study Population	14
3.4.2 Recordings	14
3.4.2.1 Movement 1: Heel Slides	15

3.4.2.2	Movement 2: Sitting Leg Raises	16
3.4.2.3	Movement 3: Standing Lateral Leg Raises	17
3.4.2.4	Movement 4: Squats	18
3.4.3	Labelling	19
3.5	Preprocessing and Windowing	20
3.6	Classification Model Architectures	21
3.7	LOSO Cross-Validation Test	23
3.8	Pseudo-Live Classification and Post-Processing	24
3.9	Feature extraction	25
3.9.1	Calculation of Hip Angle	26
3.9.2	Repetition Counter	27
3.9.3	Evaluation of Range of Motion	27
3.9.4	Evaluation of Knee Alignment and Motion Deviation	28
3.9.5	Evaluation of Movement Smoothness and Jerk	28
3.10	Supplementary Analyses	29
3.10.1	Execution-Criterion Classification	29
3.10.2	Exploratory LDLJ Evaluation	29
4	Results	31
4.1	Consistency of the Sampling Frequency	31
4.2	Model Accuracy and F1-score	33
4.2.1	Offline Performance	34
4.2.2	LOSO Performance	34
4.2.3	Pseudo-live Performance	36
4.3	Feature Extraction	37
4.3.1	Repetition Counter	38
4.3.2	Range of Motion	38
4.3.3	Knee Drop	40
4.3.4	Movement Smoothness and Jerk	42
4.4	Supplementary Analyses	48
4.4.1	Execution-Criterion Classification	48
4.4.2	Exploratory LDLJ Evaluation	50
5	Discussion	51
5.1	Limitations	51
5.1.1	Sample Diversity and Generalizability	51
5.1.2	Missing Ground Truth for Feature Extraction	52
5.1.3	Global Coordinate System Assumptions and Drifts	52
5.2	Sampling and Transferring of Data	53
5.3	Classification	54
5.3.1	Model selection	54
5.3.2	LOSO results	55
5.3.3	Pseudo-live results	56
5.3.4	Execution-Criterion Classification	57
5.3.5	Interpretability of Execution-Criterion Classification	58
5.4	Feature extraction	59
5.4.1	Repetition Counter	59

5.4.2	Range of Motion	60
5.4.2.1	ROM for Heel Slides	60
5.4.2.2	ROM for Standing Leg Raises	61
5.4.3	Knee drop	62
5.4.3.1	Knee Drop for Sitting Leg Raises	62
5.4.3.2	Knee Drop for Squats	63
5.4.4	Movement Smoothness and Jerk Analysis	64
5.4.4.1	Jerk During Sitting Leg Raises	64
5.4.4.2	Jerk During Squats	65
5.5	System design	66
5.5.1	Size of the system	66
5.5.2	Integrated battery	66
6	Conclusion	69
6.1	Future Work	69
6.1.1	System and Hardware Development	69
6.1.2	Further Data Collection and Method Development	70
	Bibliography	71
A	Appendix 1	I
B	Appendix 2	V

List of Figures

2.1	Side- and front view of the system used during this project. The placement of the sensors is marked by the arrows.	5
2.2	Three possible ways of combining CNN and LSTM networks. (a) represents the CNN-LSTM architecture, (b) represents the LSTM-CNN architecture, and (c) represents the parallel architecture.	7
3.1	A block diagram of the information flow in the system.	12
3.2	How heel slides were performed in accordance with the different execution variations during the recordings.	16
3.3	How sitting leg raises were performed in accordance with the different execution variations during the recordings.	17
3.4	How standing leg raises were performed in accordance with the different execution variations during the recordings.	18
3.5	How squats were performed in accordance with the different execution variations during the recordings.	19
3.6	Finalized model architecture for the CNN-LSTM model.	22
3.7	Finalized model architecture for the LSTM-CNN model.	22
3.8	Finalized model architecture for the parallel model.	23
4.1	Sampling frequencies of the system when writing the data directly to a computer utilizing UDP for a desired frequency of 50 Hz, 100 Hz, 250 Hz, and 500 Hz. The box in the north-west corner of each plot presents the mean sampling frequency, standard deviation, and proportion of samples outside of the $\pm 5\%$ marks (red dotted line). . .	32
4.2	Sampling frequencies of the system when writing the data over to a locally connected microSD card for a desired frequency of 50 Hz, 100 Hz, 250 Hz, and 500 Hz. The box in the north-west corner of each plot presents the mean sampling frequency, standard deviation, and proportion of samples outside of the $\pm 5\%$ marks (red dotted line). . .	33
4.3	Training and test accuracy for the LOSO cross validation.	35
4.4	Training and test LOSO accuracy for each participant.	35
4.5	Confusion matrices for the CNN-LSTM network during pseudo-live tests, before and after post processing. Normalized along the true label.	37
4.6	The estimated mean range of motion during heel slides and standing leg raises for the three execution criteria.	39

4.7	Mean of maximum and minimum knee bend during heel slides, and hip rotation during standing leg raises, for the three execution criteria.	40
4.8	The maximum knee drop over repetitions observed for all participants during the three execution criteria.	41
4.9	Knee drop and total leg rotation during squats depending on the variation of the performed movement.	42
4.10	Difference in mean of LDLJ for sitting leg raises during the different execution criteria.	43
4.11	Difference in mean of LDLJ for squats during the different execution criteria.	44
4.12	Difference of LDLJ for sitting leg raises during the different execution criteria.	45
4.13	Difference of LDLJ for squats during the different execution criteria. .	47
4.14	Model performance on the test dataset, for the 13 class classification model (C = correct, LO = less optimal, P = poor).	49
4.15	Confusion matrices for the 13 label classification model separated based on exercises and execution criteria.	49
4.16	Mean difference of LDLJ for sitting leg raises when the orthosis was unlocked.	50
A.1	Distribution of age, leg length and BMI within the study population. The y-axis in the histogram shows the number of participants in each of the 10 boxes.	I
A.2	The loss curves for the CNN-LSTM, LSTM-CNN and parallel CNN and LSTM networks.	III
B.1	Sampling frequencies of the system when writing the data directly to a computer utilizing UDP for a desired frequency of 150 Hz, 200 Hz, 300 Hz, 400 Hz, 750 Hz, and 1000 Hz. The box in the north-west corner of each plot presents the mean sampling frequency, standard deviation, and proportion of samples outside of the $\pm 5\%$ marks (red dotted line).	VI
B.2	Sampling frequencies of the system when writing the data over to a locally connected microSD card for a desired frequency of 150 Hz, 200 Hz, 300 Hz, 400 Hz, 750 Hz, and 1000 Hz. The box in the north-west corner of each plot presents the mean sampling frequency, standard deviation, and proportion of samples outside of the $\pm 5\%$ marks (red dotted line).	VIII
B.3	Confusion matrix of the CNN-LSTM network when evaluated on the test dataset. Normalized along the true label.	IX
B.4	Example of the segmentation before and after post-processing. The participant has consented to the publication of the data segmented in this figure.	X
B.5	Difference in mean of LDLJ for heel slides during the three execution criteria.	XIII
B.6	Difference in mean of LDLJ for standing leg raises during the three execution criteria.	XIV

B.7	Difference of LDLJ for heel slides during the three execution criteria.	XV
B.8	Difference of LDLJ for standing leg raises during the three execution criteria.	XVI

List of Tables

3.1	The features that were extracted for each class and what axis and reference point was used for each calculation.	26
4.1	Accuracy and macro F1-score on the training, validation, and test datasets for the three models.	34
4.2	Mean, maximum, and minimum computational time of one window for each model on a MacBook Pro M1.	34
4.3	LOSO accuracy and F1-score per class across iterations for the training and test datasets.	36
4.4	Accuracy and F1-score for the exercises, separated into execution criteria before the post processing of the classification output.	36
4.5	Mean accuracy and F1-score for the movements, separated into execution criteria after the post processing of the classification output.	37
4.6	The accuracy of the repetition counter on the original labelled data and the segmented data from the pseudo-live tests.	38
4.7	Accuracy and macro F1-score for the LSTM-CNN model when trained on all exercises and their execution criteria.	48
A.1	All test-ranges for the hyperparameters during the development of the finalized models.	II
B.1	Accuracy and F1-score for the CNN-LSTM network for each individual class.	IX
B.2	The mean $\pm$ SD and percentiles of the LOSO accuracy for the training and test datasets.	X
B.3	The mean $\pm$ SD and percentiles of the range of motion depending on the execution criteria for heel slides and standing leg raises.	XI
B.4	The mean $\pm$ SD and percentiles of the maximum and minimum joint rotation depending on the execution criteria for heel slides and standing leg raises.	XI
B.5	The mean $\pm$ SD and percentiles for the observed knee drop for sitting leg raises during all three execution criteria, for each participant and repetition.	XII
B.6	The mean $\pm$ SD and percentiles for the observed knee drop and total leg rotation during squats for all three execution criteria, per person and per repetition.	XII

1

Introduction

The ability to objectively measure and analyse human movements has become increasingly applied in a wide range of areas. Advances in wearable sensing technologies, embedded systems, and machine learning have enabled the collection and interpretation of movement data outside traditional laboratory environments [1]. Such technologies have a promising future in areas including sports performance analysis, training optimisation, interactive gaming, and movement research.

Wearable sensor systems offer a practical approach for capturing motion data in a continuous and non-invasive manner. By integrating sensors into wearable systems, it becomes possible to record information related to body movement and joint motion during everyday activities [1, 2]. Combined with data processing techniques and machine learning algorithms, these systems can be used to identify movement patterns, classify different activities, and provide quantitative information about movement execution.

Recent advancements in artificial intelligence and neural networks have further improved the ability to analyse complex temporal sensor data. Architectures such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have demonstrated strong performance in activity recognition and time-series classification tasks [3, 4]. This creates opportunities for developing wearable systems capable of performing automated movement classification and generating sensor-derived information about movement characteristics.

Previous research conducted at Chalmers University of Technology has resulted in the development of a sensorized knee orthosis prototype equipped with multiple sensors for motion capture [5, 6, 7, 8]. Although originally developed as a wearable platform for lower-limb movement monitoring, the device also provides a suitable framework for investigating sensor-based movement analysis and classification. Building upon previous work, the present thesis evaluates the feasibility of using this prototype for movement classification and sensor-derived feature extraction during controlled lower-limb movements performed by healthy participants.

1.1 Aim

The aim of this thesis was to investigate the feasibility of using a sensor-equipped wearable system for sensor-based movement classification and feature extraction. The work focused on how data collected from the wearable system could be used to train neural networks to distinguish between different lower-limb movements, as well as how sensor data could be processed to extract features describing movement patterns and execution characteristics.

1.2 Research Questions

In order to achieve the aim of this thesis, the following research questions were addressed:

1. How accurately can different lower limb movements be classified by a neural network using data collected from the wearable system?
2. What possibilities does the data provide for feature extraction, and how can extracted features be used to quantify intentionally simulated and naturally occurring variations in movement execution, such as range of motion, knee alignment and displacement, and movement smoothness?

1.3 Scope and Limitations

This project was limited to controlled movement analysis using a wearable sensor system and collecting data from healthy participants only. The work formed a technology feasibility study that investigated the ability to use the data recorded with the proposed prototype for movement classification and extraction of sensor-derived movement features.

The project did not evaluate medical, clinical, diagnostic, rehabilitation, therapeutic, or healthcare-related outcomes. The extracted features and classification outcomes should not be interpreted as assessments of health status, treatment effectiveness, rehabilitation progress, or exercise quality in a clinical sense. Any potential future applications within healthcare are outside the scope of this project.

2

Theory

This section presents the relevant theoretical background regarding the development and evaluation of the wearable sensorized system and the associated movement classification and feature extraction carried out in this project. Firstly, previous work within the AURORA initiative and development of the prototype is presented. After this, concepts related to movement classification, deep learning for time-series data, and feature extraction for biomechanical movement data are introduced. This information forms the foundation of the methodological choices and evaluation strategies used throughout this project.

2.1 Sensorized Orthosis and Sensing Hardware

Previous work conducted at Chalmers University of Technology within the larger research initiative AURORA has resulted in the development of a sensorized orthosis platform for lower-limb motion monitoring [9]. Several student projects contributed to the iterative development of the system used in this project.

2.1.1 Technical Background

A first prototype of the system was developed by Zhou [5]. A commercial knee orthosis was outfitted with two IMU sensors (ICM-42688P), one absolute capacitive rotary encoder (AMT223B-V), and an ESP32 Feather S3 microcontroller (MCU). The prototype demonstrated the feasibility of integrating multiple motion sensors into a wearable orthosis platform capable of collecting movement data.

Subsequent work by Shen focused on redesigning the electronics to improve portability, integration, and sensor synchronization [6]. A custom printed circuit board (PCB) was developed using a stacked three-board configuration, allowing the electronics to occupy less space while maintaining compatibility with the existing sensor configuration. This redesign improved the practicality of the system for portable data collection.

The redesigned PCB was later integrated into the orthosis during two engineering design projects [7, 8]. New 3D-printed enclosures were developed to hold the electronics, resulting in a more compact design which allowed for an integrated battery to be positioned below the circuitry as a power supply. The sensor placement re-

mained unchanged from the original prototype, preserving the positioning of both IMU sensors and the angle encoder while incorporating the updated electronics hardware.

In addition to the hardware integration, preliminary investigations into movement classification were carried out using data collected from the system. Three neural networks (one CNN, one LSTM, and one hybrid CNN-LSTM) were trained and evaluated for classification of predefined lower-limb movement classes. The networks achieved a classification accuracy of 73.6%, 75.9% and 78.5% respectively [8]. These results motivated the use of hybrid CNN-LSTM model architectures during the present study.

2.1.2 Orthosis Hardware Architecture

The wearable system used in this thesis consisted of a commercial knee orthosis equipped with two six-axis IMU sensors (ICM-42688P), an absolute capacitive rotary encoder (AMT223B-V), the custom PCB from Shen's thesis, and an ESP32 Feather S3 microcontroller. Together, these components provided measurements of lower-limb motion through a combination of inertial sensing and joint-angle measurements.

One IMU sensor (IMU 1) was positioned on the lower leg frame, while the second IMU sensor (IMU 2) was positioned on the upper leg frame (see figure 2.1). This configuration allowed for motion data to be collected from both segments of the leg during movement. In addition to the IMU sensors, the rotary encoder was mounted coaxially with the joint of the orthosis, enabling direct measurement of the relative joint angle between the upper and lower leg segments.

The microcontroller and the proposed PCB were integrated in the same compartment as the first IMU sensor on the lower leg. The PCB allowed for local storage of data due to an integrated microSD module, while the ESP32 microcontroller allowed for direct transfer of data through the use of Wi-Fi or Bluetooth connection. The two IMU sensors shared one SPI bus while the microSD module and the angle encoder shared a second SPI bus, as proposed by Shen [6].

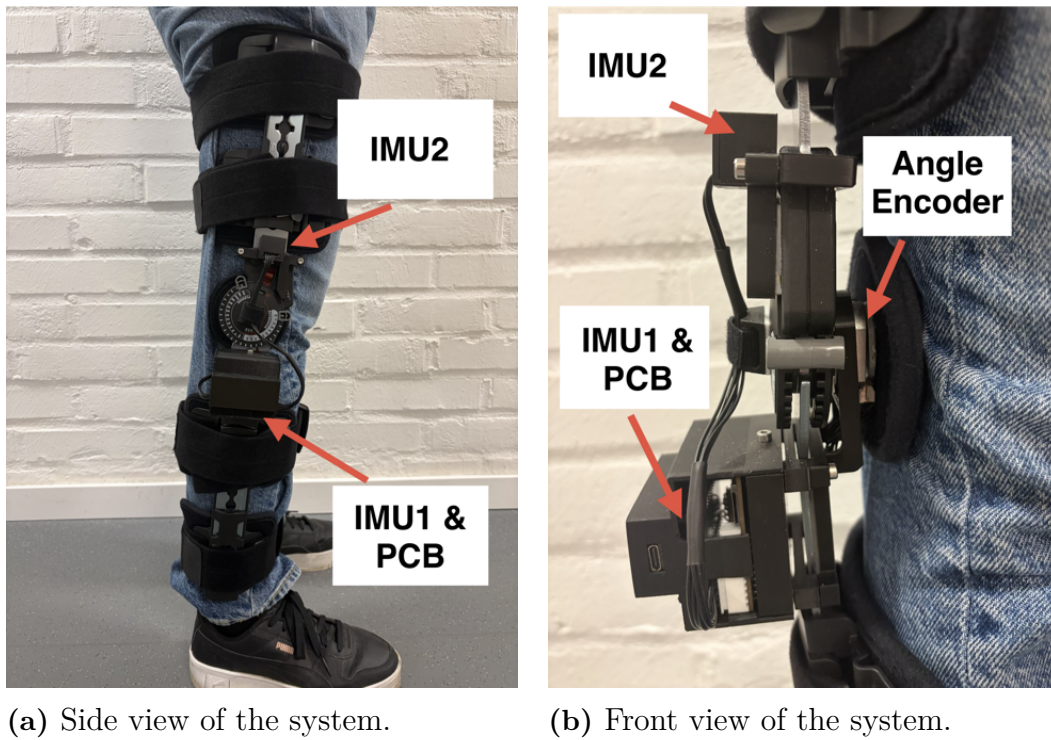


Figure 2.1: Side- and front view of the system used during this project. The placement of the sensors is marked by the arrows.

2.2 Time-Series Movement Classification

Movement classification involves assigning labels to physical activities based on sensor data, often collected from wearable devices. In the context of biomechanical studies, this typically consists of time-series data representing different joint angles, accelerations, or forces recorded during movements [10, 11]. Accurate classification enables objective identification and analysis of different movement patterns, which is crucial for biomechanical movement assessment.[10].

However, time-series data of movements introduce a multitude of challenges when it comes to classifying them; noisy data measurements, variability between individuals, and the human motion being inherently temporal are some examples [10, 11]. All of these factors cause the design of a robust and generalizable classification system to be quite challenging, especially when relying on traditional approaches.

2.2.1 Traditional Approaches for Movement Classification

Traditional approaches for movement classification often rely on manually crafted features used for training a neural network [12]. Some examples of such features for time-series data are mean values, variance, or frequency-domain characteristics and they are often used as inputs to more classical machine learning algorithms such as k-nearest neighbour (KNN) or Random Forest classifiers [12, 13].

While these methods usually are more explainable and work well in controlled settings, they are usually heavily limited by their dependence on handcrafted features, which does not generalize well across different subjects and movement patterns [13]. Furthermore, these types of models usually struggle to capture the temporal dependencies which are inherent in sequential movement data [13].

2.2.2 Deep Learning for Time-Series Data Classification

To address the limitations of traditional approaches for movement classification, deep learning has become increasingly applied for this purpose. These methods enable automatic detection of patterns of interest in the data for the classification task, saving time for preparation of data, and increasing the classification accuracy [13]. Two of the most common machine learning models for this task are Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks [3, 4]. Based on these advantages, deep learning was chosen as the primary approach for movement classification in this project.

2.2.2.1 CNN

Convolutional Neural Networks (CNNs) are designed to extract local features/patterns from structured data. Initially, CNNs were widely used in image analysis and classification tasks involving images, but because of their ability to automatically extract local hierarchical features, they are well suited for extraction of local temporal dependencies present in time-series data [3, 4]. This makes them efficient for capturing local and spatial dependencies in sensor signals. The ability of CNNs to automatically extract local temporal patterns in the data motivated their inclusion in the architectures evaluated throughout this project.

2.2.2.2 LSTM

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Networks (RNNs) which are specifically designed to model and learn from sequential data [3, 4]. LSTMs can capture long-term dependencies and temporal relationships within a sequence by maintaining an internal memory state; thus being well suited for movement classification since the order and duration of specific movements are critical for accurate interpretation. Since biomechanical movement data contains important temporal dependencies, LSTM layers were included in the models evaluated in this project.

2.2.2.3 Hybrid CNN-LSTM Architectures

Given the complementary strengths of CNN and LSTM networks, it has become common practice to combine these networks in different ways for time-series classification tasks [3, 14, 15]. CNNs are primarily utilized to extract local features from input data, whereas LSTMs are deployed to capture temporal dependencies. Within the literature, mainly three strategies are employed to combine CNN and LSTM networks [16].

One way is to use a CNN to extract features from the data and feed them to an LSTM to process the extracted feature sequence. This allows the CNN to find local patterns in the data and for the LSTM to learn temporal dependencies in the features to make a final decision (see **(a)** in figure 2.2 below) [3, 4].

Another way to combine these networks is to do it the other way around, that is to feed the input data to an LSTM which captures the temporal dependencies in the data, and then feed the output from the LSTM to a CNN [14]. In this case, the CNN will be responsible to extract features from the temporal dependencies and make a final decision. This approach has been widely used in time-series analysis and natural language processing (see **(b)** in figure 2.2 below) [14].

The third alternative to combine the two networks is to let them work in parallel. This means that the data is fed as input to both networks simultaneously [15]. The CNN extracts local spatial features from the input data and the LSTM extracts temporal dependencies within the input data. The output from these networks are then fed into a fully connected layer to produce the final decision (see **(c)** in figure 2.2 below) [15].

All three of these hybrid strategies were implemented and experimentally compared during this thesis.

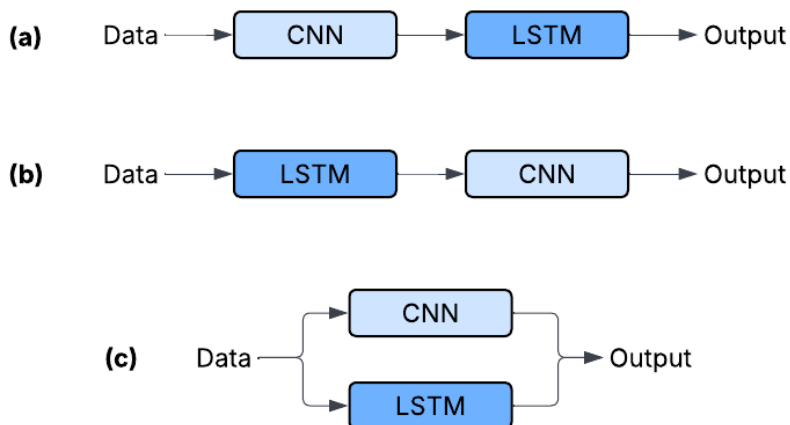


Figure 2.2: Three possible ways of combining CNN and LSTM networks. **(a)** represents the CNN-LSTM architecture, **(b)** represents the LSTM-CNN architecture, and **(c)** represents the parallel architecture.

2.3 Model Evaluation and LOSO

Evaluating machine learning models for movement classification presents challenges that are not always encountered in more conventional classification tasks. Human movement data is highly subject-dependent, meaning that multiple factors such as mobility, movement speed, body proportions, and individual execution strategies

may introduce participant-specific patterns in the data [17]. As a result, movement data collected from the same participant often contains movement characteristics that are not representable for the wider population.

A common approach when evaluating models is to randomly divide the available data into training, validation, and test datasets. While this approach is appropriate for many machine learning tasks, it can lead to misleading measurements regarding performance of networks when trained on time-series movement data [17]. If windows containing similar data from the same participant are randomly split into the three datasets, this will likely result in multiple windows from the same participant being split into both the training and test datasets. This can allow the model to partially learn participant-specific movement characteristics rather than the underlying movement patterns associated with the classification task [17]. As a result, the performance of the network may overestimate the network’s ability to generalize to unseen individuals, even if it is tested on unseen data. This issue becomes particularly important when the objective is to evaluate how well a model can classify movements performed by participants that were not included during training.

To address this challenge, Leave-One-Subject-Out (LOSO) cross-validation is often used in movement classification studies [17]. In LOSO validation, all the data originating from one participant is excluded from the training process and used exclusively for testing. This means that the model is trained using the data from all remaining participants and evaluated on the excluded participant. This is repeated once for every participant in the dataset, resulting in a separate evaluation of the network for each participant. The final performance is often obtained by deriving the mean result from all iterations, providing an estimate of how well the model generalizes to previously unseen participants [17].

For this reason, the results obtained through LOSO cross-validation in this project, should be viewed as the primary results in regards to model generalizability.

2.4 Sensor-Derived Movement Features

Feature extraction refers to the process of transforming raw data (e.g. sensor data) into meaningful information and metrics which can be used for analysis [10]. In the context of movement analysis, feature extraction is often about extracting and analysing parts of sensor data to produce quantitative measurements to capture important characteristics of a signal or motion [10, 11].

While classification models can be used to determine which movement was performed when, they do not inherently provide information about the movement characteristics that contributed to the classification decision. Even when a model can distinguish between predefined execution categories, there may still be an ambiguity about what parts of the recorded signal influenced the network in its decision [18]. Quantitative biomechanical features are therefore important for interpreting movement execution characteristics [10]. For this reason, feature extraction plays an important role in complementing automated classification with interpretable move-

ment metrics.

During movement analysis of lower limb exercises, multiple biomechanical characteristics can be evaluated from sensor data. Three relevant aspects investigated during this study were the smoothness of the performed exercise, the range of motion (ROM), and knee alignment and movement deviations. The smoothness of an exercise can be indicated with a measurement called jerk [19]. During this study, attempts were made to extract these three features from the collected data to evaluate the system and its sensor configuration.

2.4.1 Range of Motion

Range of motion is a fundamental metric in the assessment of movement execution and is measured as the difference between the maximum and minimum joint angles during a motion [20]. ROM is commonly used to quantify movement amplitude during physical activities and controlled movements. By performing movements with different ranges of motion it will be possible to evaluate if the data collected with the system can be used for calculating sensor-derived estimates of the movement amplitude.

2.4.2 Knee Alignment and Movement Deviation

In this study, attempts have been made to identify deviations from predefined reference trajectories. These deviations are represented by a sensor-derived proxy referred to as *knee drop*. The term is used only as a project-specific description of deviations in knee flexion and knee alignment, and does not represent a clinical measurement. Depending on the movement knee drop refers to different deviations; for example a bend of the knee when the leg is supposed to be kept straight (e.g. sitting leg raises), or medial deviations in joint alignment during weight-bearing movements (e.g. squats). By intentionally introducing deviations from a reference movement trajectory, it is possible to evaluate if the data collected with the system is of high enough complexity to estimate these deviations.

2.4.3 Jerk

Jerk is defined as the rate of change of the acceleration over time and is represented by the third derivative of position [19, 21]. It is commonly used as a measurement of movement smoothness, where a lower absolute value of jerk indicates a smoother and more continuous movement [21]. The decision was made to evaluate jerk during this study to determine if the IMU sensors are sensitive enough to enable evaluation of jerk.

In biomechanics, jerk is often used to quantify differences in movement smoothness between repetitions of a motion [19, 21]. Less smooth movements can for example be observed after strenuous activity such as training. Fatigue-induced tremors typically exhibit a narrow frequency span centred around 10 Hz [22, 23]. The ability to detect these variations could therefore prove beneficial for future movement analysis

applications.

Since different exercises and movements tend to naturally have different amounts of jerk, it is important that the jerk of one movement is not compared to that of another [19, 21]. This would otherwise result in a faulty comparison. Furthermore, the execution time of an exercise is also crucial for the calculated smoothness of a movement. The length and weight of a person will also introduce an error since taller people tend to produce more jerk than shorter people. Therefore it is important to use a measurement of jerk which is normalized over time and between people. This measurement is often referred to as log dimensionless jerk (LDLJ) and is defined as:

$$LDLJ = -\ln\left(\frac{t_2 - t_1}{a_{peak}^2} \int_{t_1}^{t_2} \left|\frac{d}{dt}a(t)\right|^2 dt\right) \quad (2.1)$$

where t_1 and t_2 are the start and end instance of a movement, $a(t)$ is the acceleration, and a_{peak} is the maximum acceleration during the time interval t_1 to t_2 [19, 21].

3

Methods

In order to evaluate the system the following was investigated:

1. Correctness of collected data; this included analysis of the frequency at which the data had been collected, as well as testing the limits of the system.
2. How well suited the collected data was for machine learning algorithms to distinguish between lower-limb movements, and if it was possible to extract features represented through intentional execution variations.

The evaluation of the sampling frequency was done during two different settings. The first setting was when the data was written over to a microSD card locally connected to the orthosis. The second setting was when the data instead was transferred over to a computer via a Wi-Fi hotspot by using a User Datagram Protocol (UDP).

To determine how well suited the data was for machine learning algorithms, and what features can be extracted from such data, a controlled movement data-collection study was conducted. The study was carried out with the goal of collecting data for training the machine learning model and feature extraction, and it was approved by the Swedish Ethical Review Authority (registration number: 2025-07967-01).

3.1 System Setup

The orthosis evaluated in this thesis was an Enovis's DonJoy X-ROM Post-OP Knee Brace. It was a dual frame orthosis that stabilizes the knee in the sagittal plane [24]. The orthosis had an adjustable knee joint that could be locked at specific angles to limit its range of motion. This kind of orthosis is standard equipment used in many healthcare facilities.

The orthosis had been outfitted with two ICM-42688 IMUs, one AMT22 rotary encoder and an ESP32-S3 microcontroller. The ICM-42688 IMUs were six-axis IMUs that had the ability of recording both acceleration and angular velocity in three dimensions [25]. The first IMU (IMU 1) was positioned on the shin together with the microcontroller, while the second IMU (IMU 2) was positioned on the thigh. The AMT22 rotary encoder was an absolute capacitive encoder which had the ability of measuring the true rotation based on a reference point [26]. This unit was placed coaxially with the rotational axis of the orthosis' right frame. The ESP32-S3 was a

240 MHz dual-core microcontroller with built in Wi-Fi and Bluetooth transferring alternatives [27]. This sensor configuration was chosen in order to capture both kinematic motion data and joint angle information to promote movement classification and biomechanical feature extraction. All sensors were configured according to their default settings. The two IMUs' full scale ranges were set to $\pm 16g$ and ± 2000 degrees-per-seconds (dps) for the accelerometer and gyroscope respectively, and the encoder had a resolution of 14 bits.

A block diagram visualizing the flow of information in the system can be seen in figure 3.1.

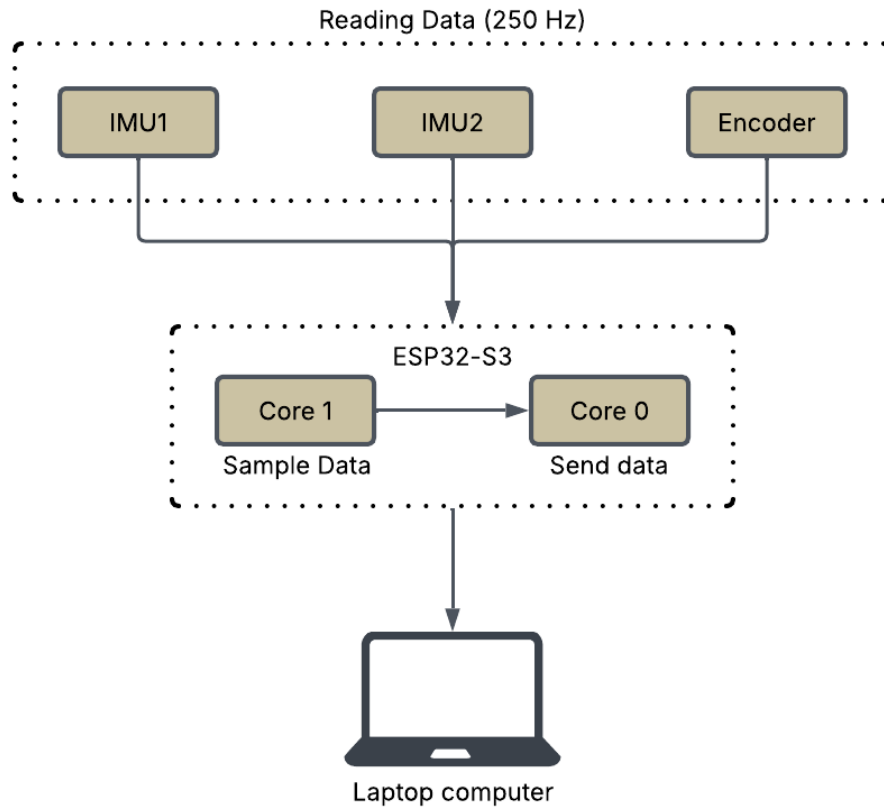


Figure 3.1: A block diagram of the information flow in the system.

3.2 Changes to the Hardware

Throughout this project changes had to be made to the system before the start of the study. The 3D printed casings for the IMU housing and the main electronic box were changed. The screw holes for the bottom plate of the main electronic enclosure had to be moved and enlarged. The original screw holes were slightly misaligned with the circuit board and too small to effectively insert the M2 and M3 brass inserts. The size of the holes were updated in accordance to the recommended hole size for these brass inserts.

The original positioning of the two electronic casings hindered the orthosis's full range of motion. At maximum flexion the two casings would collide. To resolve this issue, a new casing was printed for the second IMU that shifted the IMU enclosure further down the shin brace. To help stabilize the new casing an anchoring arm was implemented into the new design to increase the surface contact between the casing and the orthosis.

To minimize the risk of any external cables getting damaged or pulled out during the recordings all cables were shortened. The Dupont connector that connected the cables for the second IMU to those from the main electronic enclosure was removed and replaced with new cables that directly connected the two electronic enclosures. Furthermore, a hook and loop fastener was added to the centre of rotation of the orthosis and the cables were led through it. This was done to ensure that the cables connecting the two enclosures were subjected to as little movement as possible to minimize noise and the risk of the cables getting pulled, as well as guaranteeing repeatability during all recordings.

3.3 Sampling Frequency Evaluation

To evaluate how well suited the system was for collection of movement data, the stability of the sampling frequency was investigated. A stable sampling frequency is important for reliable time-series movement analysis, since loss of data will introduce ambiguities in the collected data. This may introduce misleading measurements, as well as faulty data for training of a classification algorithm. By analysing the stability of the sampling frequency, it would be possible to determine whether or not the system was well suited for the intended task.

The stability of the sampling frequency was evaluated both when transferring data over Wi-Fi (using UDP), and when writing the data over to the SD card. This was done by recording a short section of data (approximately 10 seconds) at different frequencies while the system was kept completely still. The sampling frequency was then calculated as:

$$f_s = \frac{1}{\Delta t} \quad (3.1)$$

where Δt is the time between two consecutive samples. A scatter plot (sampling frequency against time) was then created where every sample is represented by a dot. The frequencies that were tested were 50 Hz, 100 Hz, 150 Hz, 200 Hz, 250 Hz, 300 Hz, 400 Hz, 500 Hz, 750 Hz and 1000 Hz for both settings.

3.4 Study Design

The biomechanical study carried out during this project had the main goal of collecting movement data for use in the development of machine learning models, and future sensor-based movement analysis research. A total of 24 healthy participants were recruited and invited to a recording session in which they performed different exercises as their movement data was recorded with the sensor-equipped orthosis

worn on the right leg. The participants were asked to perform four controlled lower-limb movements selected to span different knee and hip motion patterns. These movements were heel slides, sitting leg raises, standing lateral leg raises, and two-legged squats.

The movements were chosen to make it possible to evaluate different aspects of the system. Apart from evaluating if the data collected with the orthosis could be used for training a neural network, it was also evaluated if the data from the IMUs and angle encoder were sufficient and of high enough quality to differentiate between:

- Hip and knee extension exercises
- Sitting and standing exercises
- Movements that require different levels of strength and muscle control
- Movements performed in different planes of motion

Each participant performed all four movements according to three predefined execution criteria: “correct”, “less optimal”, and “poor”. The execution categories were designed to simulate predefined deviations in execution amplitude and alignment. This would result in a more extensive dataset which would allow for training of a more robust and generalizable classification model, as well as the ability of developing and evaluating methods for assessing variations of the performed movements.

3.4.1 Study Population

All participants were screened according to the approved inclusion criteria for the study and they were informed and consented to their data being used for this project and future studies within the AURORA initiative. The study population consisted of 15 male subjects, 8 female subjects and 1 subject that wished to not disclose their sex. All subjects were between the ages 19-35 with a mean and median age of 24.6 and 24.5 respectively. Histograms of the age, BMI and leg length distribution in the study population can be seen in Appendix A.1.

3.4.2 Recordings

Every recording consisted of 20 repetitions of each movement for one execution criterion; that is, one recording of all four exercises when performed according to the “correct” execution criterion, one recording of all four movements when performed according to the “less optimal” execution criterion, and one recording of all four movements when performed according to the “poor” execution criterion. The data was recorded using a sampling frequency of 250 Hz, and it was directly transferred to a computer through the MCU’s Wi-Fi hotspot utilizing UDP. The sampling frequency was set to 250 Hz to allow for accurate recordings of movements [28]. The sensor data for one recording was saved as a separate CSV file, resulting in a total of $3 \times 24 = 72$ CSV files containing 13-channel movement data. In order to minimize the effect and bias from fatigue and LSTM learning patterns, the order of which the

movements were executed during the recordings were randomized.

In the beginning of each recording a calibration phase was carried out to remove any baseline deviation and sensor drifts. During this phase the orthosis was locked at 0° and the participants were asked to stand still with their legs put together. During all recordings, the participants were asked to perform the movements in sync with a metronome. The metronome was set to 40 BPM and every beat represented where the participants were supposed to be in the movement. On one beat they were to be in the starting position of the movement, and on the next beat they were supposed to be in the turning point of the movement. The main reason as to why the metronome was used was to guarantee reproducibility.

3.4.2.1 Movement 1: Heel Slides

Heel slides were selected as a controlled knee-flexion/extension movement with adjustable movement amplitude. The movement was performed with the participant sitting on the front edge of a chair while sliding the foot back and forth along the floor. The starting position can be seen in figure 3.2a. During the “correct” execution of the movement, the participant bent their knee to an angle of approximately 90° (see figure 3.2b). To make sure that the angle was reproducible for all 20 repetitions, a piece of tape was put on the floor to mark how far back the participants were to slide their foot. To simulate a variation in the range of motion, the participants performed the movement with approximately 70° of knee flexion during the “less optimal” condition (see figure 3.2c) and approximately 50° during the “poor” condition (see figure 3.2d). These angles were also marked with a piece of tape on the floor. The variations were chosen to represent different predefined movement-amplitude levels.

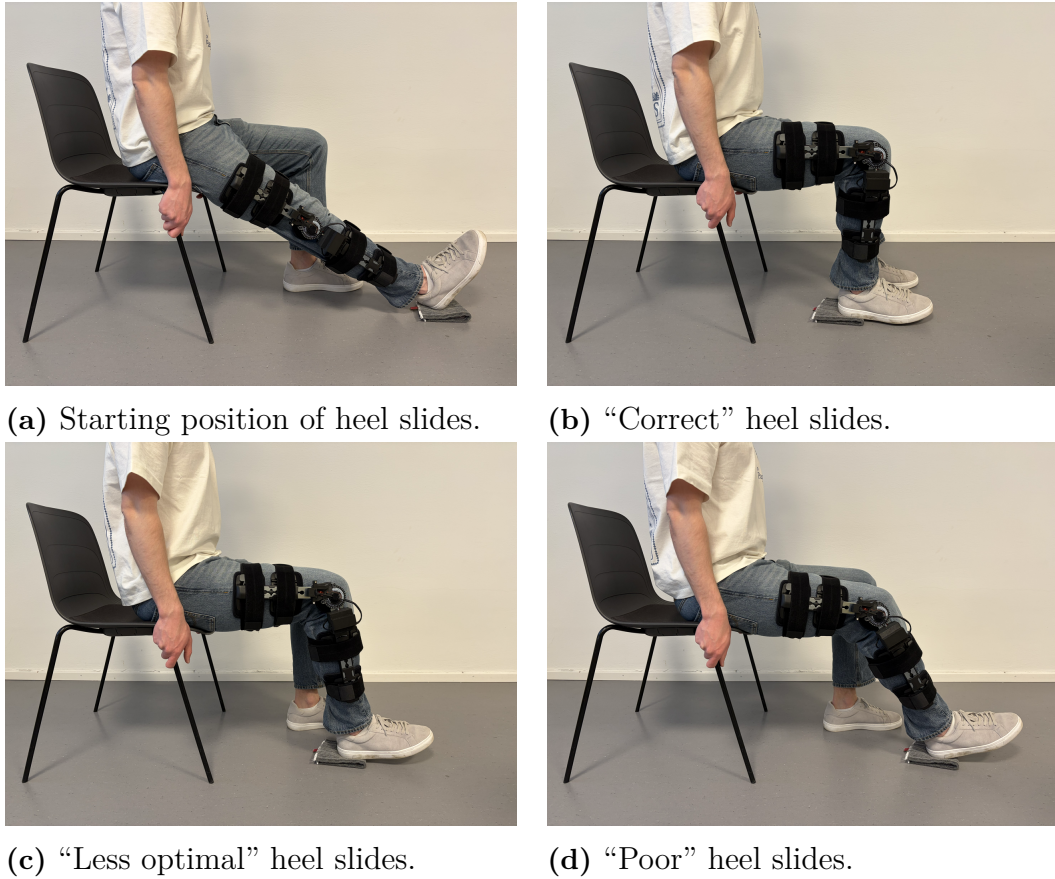


Figure 3.2: How heel slides were performed in accordance with the different execution variations during the recordings.

3.4.2.2 Movement 2: Sitting Leg Raises

Sitting leg raises is a movement commonly used to produce a repeated hip-flexion movement while controlling the knee angle. This movement was carried out with the participants sitting on the front edge of a chair while lifting and lowering their leg with the knee fully extended. The starting position can be seen in figure 3.3a. Maintaining knee extension throughout the movement was considered important for the “correct” execution criterion, and therefore the orthosis was locked at 10° for this execution criterion. This limited the participant from bending their knee more than 10° throughout the whole movement (see figure 3.3b). During the “less optimal” and “poor” execution criteria, the participants were instructed to progressively flex their knee during the leg elevation. This was done to simulate predefined knee-flexion variation during leg elevation (referred to as knee drop in this thesis). To ensure reproducibility, the orthosis was mechanically restricted to 20° and 30° of knee flexion during the “less optimal” and “poor” execution criteria respectively. The participants were asked to bend their knee until reaching the mechanical stop during each repetition (see figure 3.3c and 3.3d).

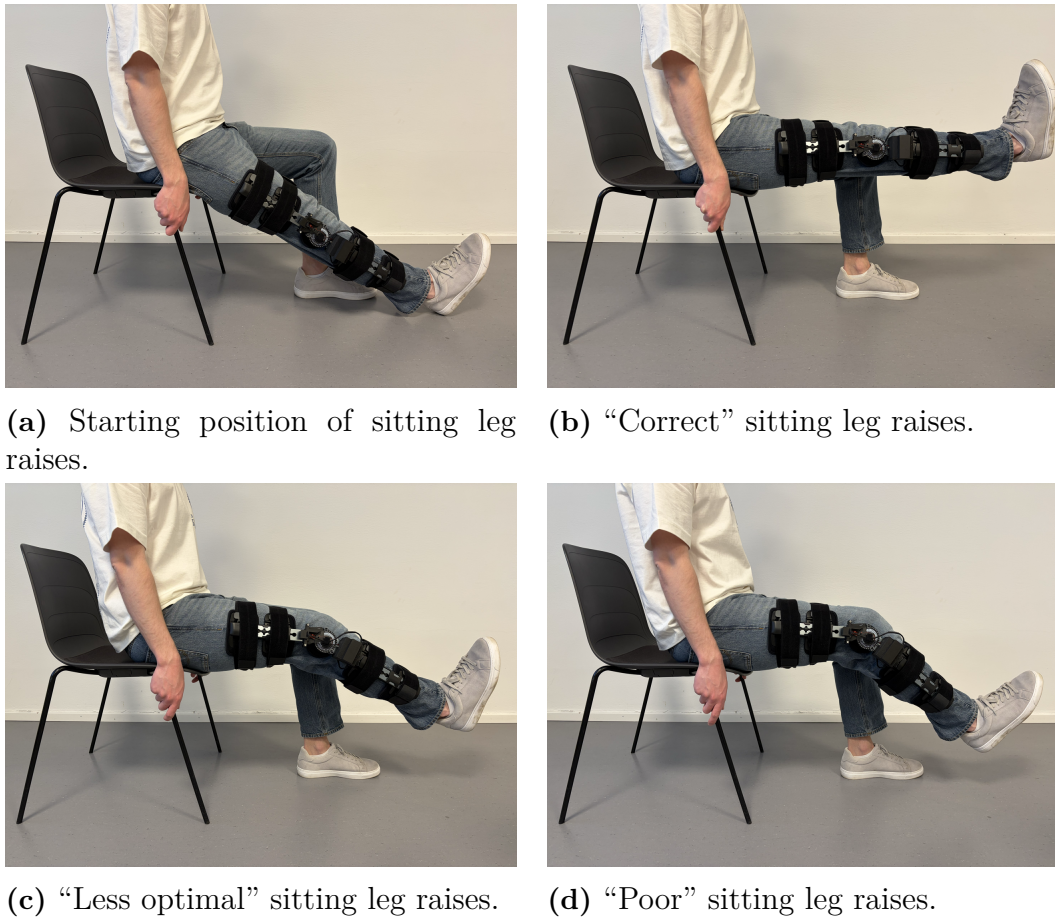


Figure 3.3: How sitting leg raises were performed in accordance with the different execution variations during the recordings.

3.4.2.3 Movement 3: Standing Lateral Leg Raises

During standing lateral leg raises, the participant performed leg abductions while keeping the leg completely straight. To produce different movement amplitudes the participants were instructed to lift their leg at different degrees. The starting position can be seen in figure 3.4a. For the “correct” execution criterion the participants performed a 45° hip abduction (see figure 3.4b). To ensure reproducible results the participants were instructed to perform the movement by lifting their foot until they touched a soft wall or curtain while standing at a piece of tape to ensure that roughly the same angle was achieved during all 20 repetitions. The position of the tape was determined based on the participants leg length and the degree of hip abduction in accordance with trigonometric functions. During the “less optimal” and “poor” variations the participants were instructed to lift their leg to 30° and 15° respectively (see figure 3.4c and 3.4d).

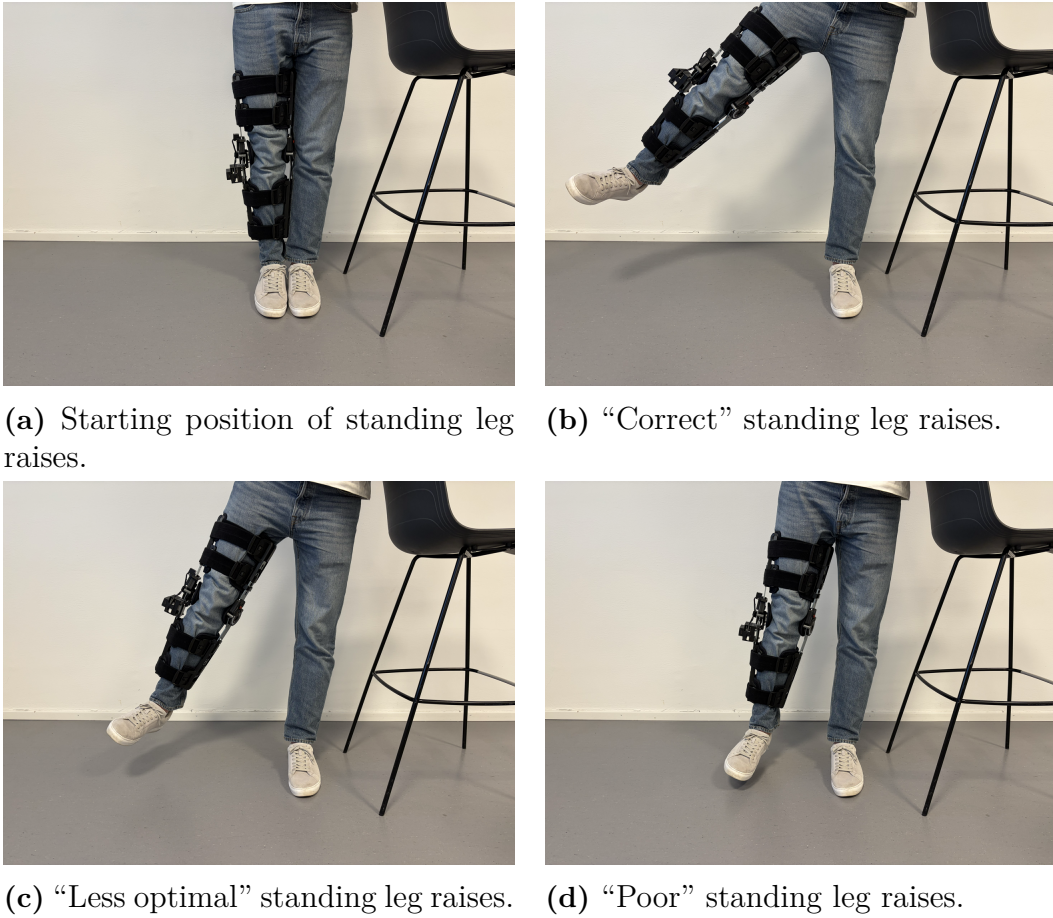


Figure 3.4: How standing leg raises were performed in accordance with the different execution variations during the recordings.

3.4.2.4 Movement 4: Squats

During the “correct” variations of the squats the participants were instructed to stand firmly on both feet at shoulder width, toes pointing slightly outwards (see figure 3.5a for starting position), and bending their knees circa 90° while maintaining proper knee alignment over the feet (see figure 3.5b). The “less optimal” and “poor” categories intentionally introduced increasing medial knee displacement during the upward phase of the movement (referred to as knee drop in this thesis). This was simulated at two different levels. During the “less optimal” condition the participants were instructed to displace their knees partially inwards (see figure 3.5c), and during the “poor” variation they were instructed to displace their knees inwards until contact occurred between the knees (see figure 3.5d).

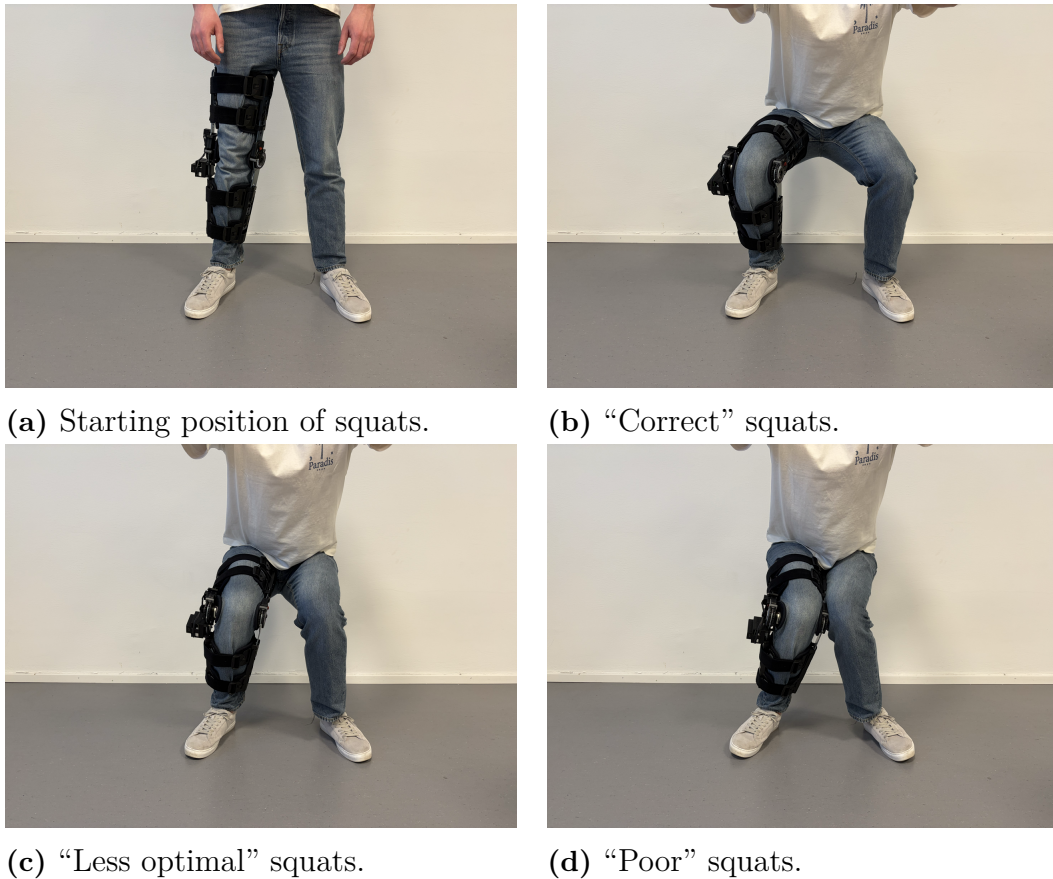


Figure 3.5: How squats were performed in accordance with the different execution variations during the recordings.

3.4.3 Labelling

In order to train a neural network on classifying movements based on time-series movement data, a ground truth needed to be established in regards to when a specific movement was performed. This was done by labelling the data live during the recordings using the Wi-Fi connection. In the beginning of each recording the orthosis would be locked at 0° and the participant was asked to stand still. This part of the recording was labelled as 9 and later used to calibrate the data. The labels used for each movement were:

- Label 1. Heel slides
- Label 2. Sitting leg raises
- Label 3. Standing lateral leg raises
- Label 4. Squats

As the participants performed a movement the corresponding key was pressed allowing the system to label the data as it was being collected. To minimize human error the start of the movement was signalled to both the participant and the la-

beller using a metronome and a countdown. Whenever a predefined movement was not performed the data was labelled as zero which represents non-movement data or “other” movements during the recordings.

3.5 Preprocessing and Windowing

Before the data was used to train a neural network for classification and segmentation of movements, it was preprocessed. The preprocessing steps involved calibration, filtering, and splitting the data into training, validation, and test datasets.

During the calibration phase of each recording (the section labelled as 9), any sensor offset was removed from the signal by calculating the mean output during the calibration phase and then subtracting this value from the rest of the signal. This was done to adjust for any sensor offsets or drifts in the channels from the IMU sensors and the angle encoder. The acceleration channel of the IMU sensors representing the direction pointing downwards when standing up (y-axis) was calibrated by centering the signal baseline around -1 instead of 0. This was done to account for the gravitational force that the sensors were being subjected to, assuming that all participants were standing with their legs perpendicular with the floor during the calibration.

In order to remove random noise from the data, filtering of the signal was applied. This was done with a Savitzky-Golay filter with a window length of 25 samples and a polynomial order of 4. With a sampling frequency of 250 Hz, the window length of 25 samples corresponded to (~ 0.1 seconds). This window size was selected to preserve as much of the original signal as possible, while at the same time reducing the effect of the high-frequency noise. Since Savitzky-Golay filters are smoothing filters, a window size too large would increase the risk of manipulating or attenuating the true movement peaks in the signal, while a too small of a window size would keep more noise present in the data after filtering. The window size can be further motivated since tremors related to muscle fatigue have a narrow frequency span centred around 10 Hz [22, 23]. Furthermore, the polynomial order of the filter was set to 4 since the complexity of the filtered data had to be high enough for the subsequent jerk measurements to be applicable. Since jerk is the third derivative of position, a minimum of a fourth-order polynomial is mathematically required to avoid constraining the calculated jerk to a constant value within each filter window.

The data from the 24 participants was then split into different datasets which allowed for evaluation of the networks. To guarantee that no data leakage occurred during the later evaluations of the network, three participants were randomly selected and excluded from the dataset used for the offline training and evaluation.

The remaining 21 people were split into train/val/test sets of size 80/10/10%. For each recording a sliding window was passed through the entire signal. The window size was set to 1125 samples (~ 4.5 seconds) and the stride of the sliding window was set to 250 samples (~ 1 seconds), resulting in an overlap of $\sim 78\%$. Each window was then randomly assigned to the train, validation or test set, resulting in each

participant contributing to all the datasets. The true label for each of these windows was assigned as the label of the middle pixel in each window. Since all participants contributed to all datasets there is a risk of data leakage. These datasets were therefore only used as a means of evaluating which of the three CNN and LSTM configurations were the most optimal for this project. To evaluate the final model’s generalizability and accuracy on unseen data the model was further evaluated with LOSO cross validation and further tests on the three unseen participants who were excluded from the data split earlier.

The window configuration was established to ensure that every window contained data from roughly 1.5 repetitions; meaning that every window was represented by at least one complete movement cycle along with its initiation and termination phases. This allowed the LSTM network to learn temporal dependencies within the data without suffering from vanishing gradients, while also allowing the CNN to learn from informative spatial-temporal feature maps. Furthermore, the $\sim 78\%$ overlap of the windows acted as a data augmentation during training, which significantly expanded the dataset and prevented overfitting.

3.6 Classification Model Architectures

During the course of this project multiple classification models were trained and tested on the collected data. All models that were tested were designed based on the three more common hybrid CNN and LSTM architectures, e.g. CNN-LSTM, LSTM-CNN, and CNN and LSTM in parallel. The finalized architectures for these networks can be seen in figure 3.6, 3.7, and 3.8 below.

All of the networks had the same architecture for the CNN and LSTM parts separately, but the overall network architecture differs between them according to figure 2.2. This was done to enable evaluation of the optimal way of combining these types of networks for the classification task at hand, and therefore the models needed to be comparable. The hyperparameters for the finalized model architectures were defined heuristically based on the accuracy of the training and validation datasets only. Table A.1 in the appendix contains the ranges for some hyperparameters that were tested to finalize the model architecture.

All models were trained using the Adaptive Moment Estimation (Adam) optimizer with a learning rate of 0.0001, and Cross Entropy Loss was used as the loss function. The batch size was set to 256. The models were trained for a different number of epochs to avoid any of the models overfitting to the training data. The CNN-LSTM, LSTM-CNN and the parallel networks were trained for 10, 5, and 6 epochs respectively, and their loss curves can be seen in appendix A.2. The training of the networks were performed on a MacBook Pro M1 using a GPU that was accessed through MPS. To evaluate the models, the accuracy and macro F1-score for the training, validation, and test datasets were derived for all three models. To save time during training, the training accuracy was evaluated on each batch during the last epoch and the mean accuracy from the last epoch was calculated. The validation and test accuracies were evaluated after the networks had been trained. Note the

train/val/test accuracies might have ended up inflated from possible data leakage as explained in section 3.5.

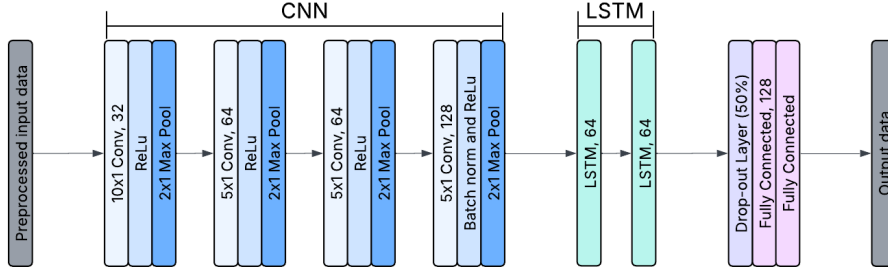


Figure 3.6: Finalized model architecture for the CNN-LSTM model.

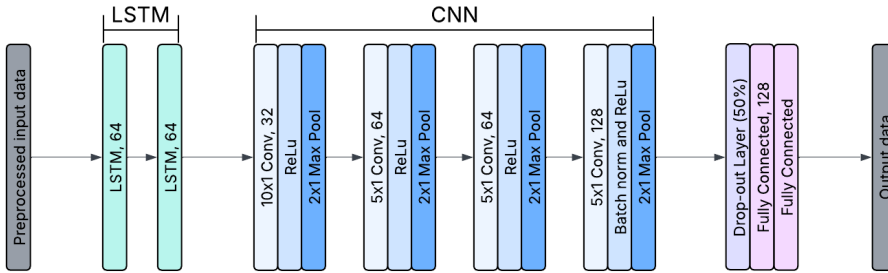


Figure 3.7: Finalized model architecture for the LSTM-CNN model.

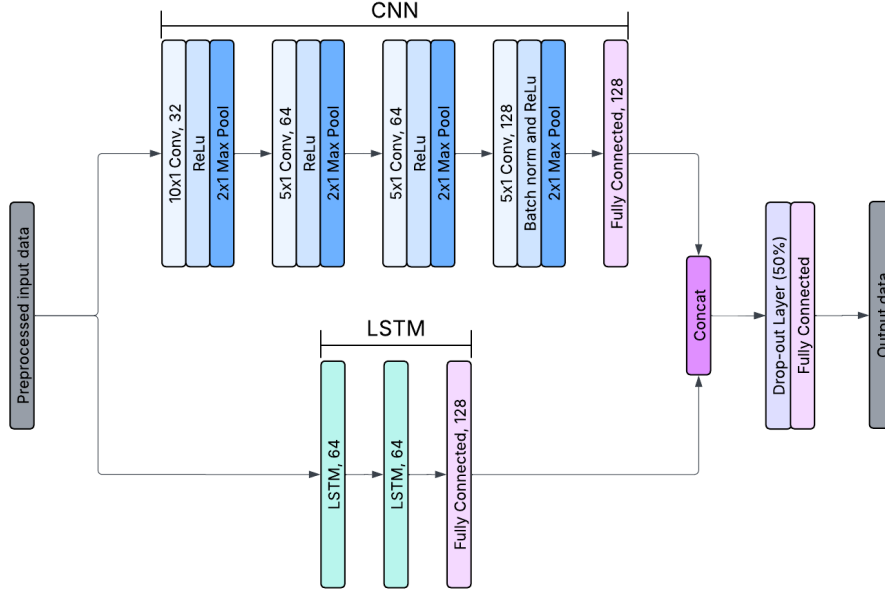


Figure 3.8: Finalized model architecture for the parallel model.

Depending on the future usage areas of the data and classification algorithms, different requirements regarding time consumption need to be fulfilled. Therefore the models were not only evaluated by their accuracy and F1-score, but also based on how long it took them to classify one window on average. This was done by letting each network classify one recording (the same recording for all networks) and computing the average, maximum, and minimum time it took to classify one window while utilizing CPU.

When the accuracy, F1-score, and time consumption of the networks were available, a trade-off between these parameters was conducted in order to determine what network would be more suitable for real-life usage areas. When the trade-off had been made, one network was chosen to be further evaluated and the others were discarded for the remainder of this project.

3.7 LOSO Cross-Validation Test

When one network had been chosen to be represented throughout the remainder of this project, the generalizability of the model was evaluated by implementing the LOSO method, covered in section 2.3. While the original train/validation/test split allowed for windows from the same participant to appear in multiple datasets, the data for one of the 24 participants were completely excluded during the training of the network during the LOSO cross-validation. This data was instead exclusively used as the test data to avoid data leakage. This was repeated for all participants so that everyone served as the unseen test data once. The final LOSO performance was derived through the mean training and test accuracy across all iterations. Fur-

thermore, a bar plot of the training and test accuracies for each participant was created.

3.8 Pseudo-Live Classification and Post-Processing

To evaluate the model’s potential for classifying live fed data a pseudo-live test was performed. A pre-recorded signal was loaded into the memory and processed window-by-window, to simulate live conditions by classifying the data within a sliding window. The sliding window had a size of 1125 samples and a stride of 25 samples (~ 0.1 seconds) to simulate a new classification occurring every 0.1 seconds. For each window the middle pixel was classified and compared to the true label to evaluate the model accuracy. To make it possible to extract movement specific features from the data it is necessary to fully segment the signal. This was done by post-processing the classified signal by extrapolating the classified pixels so that every pixel was assigned an estimated label.

To summarize the post-processing code, it worked by utilizing two running counts, `running_zeros`, and `running_exercises`. `Running_zeros` represented the number of windows classified as zeroes in a row, and `running_exercises` was the number of windows classified as an exercise between start and end. As the sliding window moved through the signal, each window was classified, and if the window was a zero the `running_zeros` count increased with one. If the classified window was an exercise then `running_exercises` was increased with one and `running zeros` was set to zero and the exercise label was saved to a list. If this was the first window to be classified as an exercise and `running_zeros` is 50 or higher its index would be saved as the start and the function for narrowing down the actual start was called. Any subsequent windows that were classified as an exercise would increase `running_exercises` and be added to the list. If a start existed and the `running_zeros` was equal to 50 then the index of the last window to be classified as an exercise would be regarded as the end and the function for narrowing down the actual end was called. If the `running_exercises` count was 30 or higher, all indices between the start and end would then be classified as the mode of the list containing all the window labels between start and end. If the `running_exercises` count was lower than 30 the entire section between start and end would be labelled as a non-exercise. Note that the 50 sample threshold means that if two exercises were closer than 5 seconds to each other the code would treat them as one single movement.

To better narrow down the start and end of a movement the model performed extra classifications around the start and end index. The interval between the index representing the start and the prior zero window was repeatedly divided in half to find an index closer to the true start. The middle index of this interval was classified, and depending on its label it was either regarded as the new start or the new prior zero window. This method was then repeated for the new interval between the zero window and start until the size of the interval was ~ 8 times smaller (~ 3 cuts). This limit was set to reduce time consumption of this code and to minimize latency.

As mentioned earlier all participants performed the movements in sync with a

metronome set to 40 BPM, resulting in the repetition time being approximately 3 seconds for all exercises. The 30 window limit represents approximately 3 seconds of classified data which guarantees that the model has classified at least one repetition of a movement before it is recognized as an exercise. The 50 windows limit for consecutive zeroes was set to ensure that the model would need to misclassify more than one and a half repetition before an movement is incorrectly split into smaller sections.

It is important to note that these tests can not be regarded as live. Since the model classified the middle pixel of each window a latency would be introduced to the classification. This latency would be half a window or ~ 2.25 seconds. To decrease this latency the classified pixel would need to be moved closer to the end of the window or the window size would need to be decreased. This was not done to optimize the segmentation accuracy. Additionally, the post-processing kept a running count of zeroes and exercises live but the extrapolation occurred offline since the latency of the post-processing was an entire exercise plus ~ 5 seconds.

To evaluate the results from the pseudo-live test the model was tested on all three recordings from the three participants, mentioned in section 3.5, that were randomly selected to represent test data for later evaluations. The results were saved for all three participants and a combined accuracy and F1-score was calculated for the “correct”, “less optimal” and “poor” execution criteria both before and after the post processing. A combined confusion matrix for all three recordings was also derived to evaluate which classes were being confused.

3.9 Feature extraction

There are many different metrics that can be used to evaluate deviations from a movement’s reference execution. In this study attempts were made to assess how suitable the recorded movement data was for evaluating range of motion, knee drop, and movement smoothness for different exercises. Apart from extracting different features from the data, a repetition counter was implemented for all exercises. All feature extraction functions were evaluated on the original data from all participants given the true labels, instead of on the segmented data from the pseudo-live tests. This decision was made to ensure that all the test results were products of the feature extraction and not the segmentation to allow for proper evaluation of the feature extraction pipeline.

A summary of all features and what axis and reference point was used during calculations can be seen in table 3.1. All movement signals have been calibrated and filtered according to section 3.5.

	Class	Axis	Unit (axis)	Unit (output)	Reference Point
Repetition counter	Heel slides	Angle encoder	Degrees	Repetitions	Straight leg
	Sitting leg raises	IMU2 gyroscope x-axis	dps	Repetitions	Beginning of movement
	Standing leg raises	IMU2 gyroscope z-axis	dps	Repetitions	Beginning of movement
	Squats	Angle encoder	Degrees	Repetitions	Straight leg
ROM	Heel slides	Angle encoder	Degrees	Degrees	Straight leg
	Standing leg raises	IMU2 gyroscope z-axis	dps	Degrees	Beginning of movement
Knee drop	Sitting leg raises	Angle encoder	Degrees	Degrees	Straight leg
	Squats	IMU1 gyroscope y-axis	dps	Degrees	Beginning of repetition
LDLJ	Heel slides	IMU1 & IMU2	m/s^2	Unitless	Beginning of repetition
	Sitting leg raises	IMU1 & IMU2	m/s^2	Unitless	Beginning of repetition
	Standing leg raises	IMU1 & IMU2	m/s^2	Unitless	Beginning of repetition
	Squats	IMU1 & IMU2	m/s^2	Unitless	Beginning of repetition

Table 3.1: The features that were extracted for each class and what axis and reference point was used for each calculation.

3.9.1 Calculation of Hip Angle

Both the sitting and standing leg raises are mainly characterized by rotation in the hip, meaning that in order to evaluate for instance range of motion the hip rotation needs to be derived. The hip angle was derived from the gyroscopic data according to equation 3.2 below, where t_1 and t_2 are the start and end instances of the movement signal respectively. The gyroscopic data was recorded with the unit degrees-per-seconds.

$$\int_{t_1}^{t_2} g(t) dt \quad (3.2)$$

The choice was made to derive the hip rotation from the data recorded with the second IMU positioned on the thigh since it is subjected to a more cyclic motion compared to the IMU on the shin during both exercises. The second IMU is also positioned closer to the hip and can only be shifted through movement around the hip joint as opposed to the IMU on the shin that can also be moved around the knee joint. Depending on the exercise, the angle was calculated around different axes; the z-axis for the standing leg raises and the x-axis for the sitting leg raises. This was done because during standing leg raises the hip's axis of rotation is aligned with the IMU's z-axis, while for sitting leg raises it is aligned with the IMU's x-axis.

3.9.2 Repetition Counter

In order to count the number of repetitions of an exercise, a signal with as little noise as possible is desirable. Based on this the angle from the absolute encoder was the optimal choice for the knee exercises (heel slides and squats), while for the hip exercises (sitting and standing leg raises) the hip angle calculated from the second IMU was utilized according to the axes mentioned in section 3.9.1.

The total number of repetitions was calculated through threshold-crossings. This works by counting the number of times the signal passes an established threshold in one direction. While this works great in theory, so is not always the case in practice, for example when the signal contains noise or if the sensor has collected some other quickly alternating movement, for example shakiness. This could cause the signal to pass the threshold multiple times during one repetition resulting in an incorrect number of registered repetitions. To address this issue a second threshold was established for the minimum allowed distance between crossings. The threshold used for the threshold-crossing was the mean signal value for the segmented exercise and the minimum allowed distance between crossings was set to 200 samples (0.8 seconds). The 200 samples should not result in any missed repetitions as long as the participants did not perform a repetition faster than 0.8 seconds. If sampling frequency is changed this value should be updated to maintain a time of 0.8 seconds. The repetition counter was tested on both the original data using the true labels, as well as the data after the pseudo-live classification to get an idea of how well both the repetition counter and segmentation performed.

3.9.3 Evaluation of Range of Motion

Similarly to the repetition counter the angle encoder and the hip angle was used to evaluate the participants range of motion for the heel slides and the standing leg raises respectively. For the standing leg raises, the position of the leg during the beginning of the movement was used as the 0° reference. To calculate the maximum and minimum rotation of the joint the peaks and valleys of the signal were found using SciPy's *find_peaks* command and a mean value was calculated for the mean maximum rotation and mean minimum rotation. The mean range of motion was derived by taking the difference between these two values. To visualize the range of motion for the different exercises, depending on the execution criterion, a box plot for all participants' mean range was created. Similarly, box plots were also used to

visualize the mean maximum, and mean minimum extension of the knee and hip for all participants.

3.9.4 Evaluation of Knee Alignment and Motion Deviation

As mentioned in section 2.4.2, knee drop was used to introduce two different simulated deviations from the desired movement trajectory. Therefore, two different methods for calculating the knee drop were derived; one for the sitting leg raises and one for the squats. The knee drop for the sitting leg raises was derived from the angle encoder, where the minimum and maximum recorded angle was calculated for each repetition separately. The mean of the minimum and maximum knee drop for each participant were then visualized in a box plot based on the exercise criterion.

For squats, knee drop was not measurable with the sensors alone, but needed to be derived in accordance with equation 3.2 for each repetition separately. As mentioned in section 2.4.2, knee drop during squats was represented by medial deviations of the knee, which could also be characterised by internal rotation around the body's longitudinal axis. The y-axis of the first IMU, positioned at the shin, was aligned with the longitudinal axis of the body during rest and only deviated slightly from it throughout the movement. The internal rotation around this axis was therefore used as an approximation of the true internal rotation. This assumed that the participants were standing with their knees pointing along the optimal trajectory in the beginning of each repetition. The starting position before each repetition would be regarded as having a medial knee displacement of 0, or a rotational angle of 0° . A positive change of angle would represent an external rotation of the knee while internal rotation would result in a negative change of the angle. The mean knee drop for each participant depending on the execution criterion (defined in the experimental design) was visualized in a box plot. The total knee rotation (the difference between internal and external rotation) was visualized in the same manner.

3.9.5 Evaluation of Movement Smoothness and Jerk

To evaluate the system's ability to capture changes in movement smoothness across repeated executions, the dimensionless logarithmic jerk measurement (LDLJ) was used (see equation 2.1). Before applying the equation to the data, it was filtered through a low-pass filter with a cut-off frequency of 10 Hz. This was done to assure that high-frequency noise was minimized, preventing its amplification during successive time differentiations when calculating jerk. LDLJ was calculated for every repetition across all exercises, and the mean of each repetition for each exercise and execution criterion was derived for both IMU 1 and IMU 2. To account for all dimensions, the jerk signal for each IMU was computed as the Euclidean norm (vector magnitude) of the time-derivatives of the acceleration across all three axes (x, y, z). The time boundaries for each repetition, defining t_1 and t_2 in equation 2.1, were obtained by identifying the peaks of the movement data utilizing the *find_peaks* algorithm from the SciPy library, similarly as during the evaluation of range of motion.

To closer evaluate the temporal changes in movement smoothness, the mean LDLJ value of all participants' first repetition was used as a reference value for the other 19 repetitions. The difference between the mean of the first repetition and all subsequent repetitions was calculated and visualized using a scatter plot. This enabled analysis of how movement smoothness and jerk developed over time and across repetitions. Focus was placed on the exercises which were considered to be more physically demanding, i.e. squats and sitting leg raises, and therefore would be more likely to reveal changes in movement smoothness during repeated executions.

In addition to the scatter plots, box plots of the difference of LDLJ between the first and consecutive repetitions were generated to further analyse trends and distributions of the smoothness during the exercises.

3.10 Supplementary Analyses

In addition to the primary methods and analyses presented previously, supplementary analyses were conducted to further investigate the possibilities for classifying the execution categories. Furthermore, unexpected findings related to the development of LDLJ for the “correct” sitting leg raises were further investigated. These analyses were exploratory and the aim was to provide additional methodological insights about the possibilities for classification of exercise variations and calculation of movement smoothness. The following sections present the methods for these supplementary analyses.

3.10.1 Execution-Criterion Classification

In order to evaluate the choice of using feature extraction as a means of assessing the variations in the performed exercises, an exploratory study was performed where a model was trained to do this automatically by classifying both the exercises and their execution criteria. This was done for the CNN-LSTM model architecture presented in figure 3.6. The training of this model was done using the same loss function, optimizer and learning rate as for the other models. The model was trained for 20 epochs and the batch size was halved from the previous training of the networks (batch size of 128). During the training of this model all 24 participants were included in the train/val/test sets according to the 80/10/10% split as earlier. The results from this model were presented with an accuracy and F1-score for the training, validation and test datasets as well as confusion matrices for the test dataset.

3.10.2 Exploratory LDLJ Evaluation

During the primary feature extraction analysis, an unexpected relationship was observed between repetition count and LDLJ for “correct” sitting leg raises. Based on both literature and the results from the “less optimal” and “poor” sitting leg raises, the movement smoothness was expected to decrease over repetitions. However, the primary analysis for “correct” sitting leg raises shows the opposite trend.

To further investigate this observation, an exploratory pilot-study was performed. This study was carried out on the two authors of this report, where 30 sitting leg raises were performed in one recording. In contrast to the recordings during the main study, the orthosis was not locked. This was done to evaluate whether or not locking the orthosis could have contributed to the unexpected relation between jerk and executed repetitions. The hypothesis was that some participants unknowingly used the mechanical lock as a support since they were unable to keep their leg completely straight due to fatigue. This could have resulted in the exercise becoming easier which in turn could have made their movement smoother, resulting in a decrease of jerk.

4

Results

This section presents the results obtained throughout this project. First, the results from the sampling frequency analysis are presented, followed by the results regarding movement classification and model performance. After this, the results from the feature extraction are presented, and lastly, the results from the supplementary analyses are introduced.

4.1 Consistency of the Sampling Frequency

The two figures below show the results of the tests carried out to investigate the system's consistency with regards to sampling frequency. Figure 4.1 shows the sampling frequency when utilizing UDP and sending the data directly to a computer through a Wi-Fi hotspot, while figure 4.2 shows the sampling frequency when sending the data to a microSD card locally connected to the orthosis. Each dot in the plots represents the sampling frequency of one sample and was calculated according to equation 3.1. The plots show the results for when using a sampling frequency of 50 Hz, 100 Hz, 250 Hz, and 500 Hz. To see the results of the other frequencies that were tested, please see figure B.1 and B.2 in the Appendix.

The consistency of the sampling frequency when utilizing UDP decreased as the sampling frequency was increased. This can be seen as an increase in standard deviation when the desired sampling frequency was increased. The recordings at 250 Hz had a mean and standard deviation of 250.02 ± 1.76 Hz and the proportion of samples outside the $\pm 5\%$ threshold was 0%.

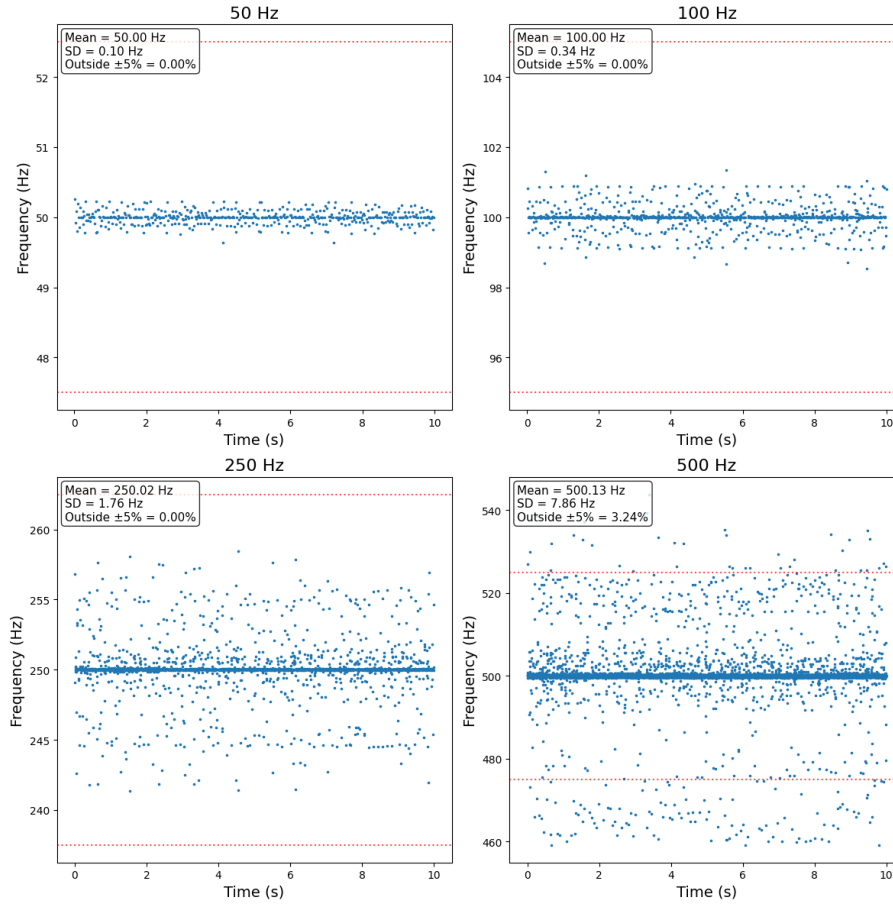


Figure 4.1: Sampling frequencies of the system when writing the data directly to a computer utilizing UDP for a desired frequency of 50 Hz, 100 Hz, 250 Hz, and 500 Hz. The box in the north-west corner of each plot presents the mean sampling frequency, standard deviation, and proportion of samples outside of the $\pm 5\%$ marks (red dotted line).

The consistency of the sampling frequency when writing the data directly over to the microSD card is presented in figure 4.2 below. The results indicate that the amount of outliers increased as the desired sampling frequency did, and the standard deviation increased in a similar way. The recordings at 250 Hz had a mean and standard deviation of 250.72 ± 20.04 Hz and the proportion of samples outside the $\pm 5\%$ threshold was 5.08%.

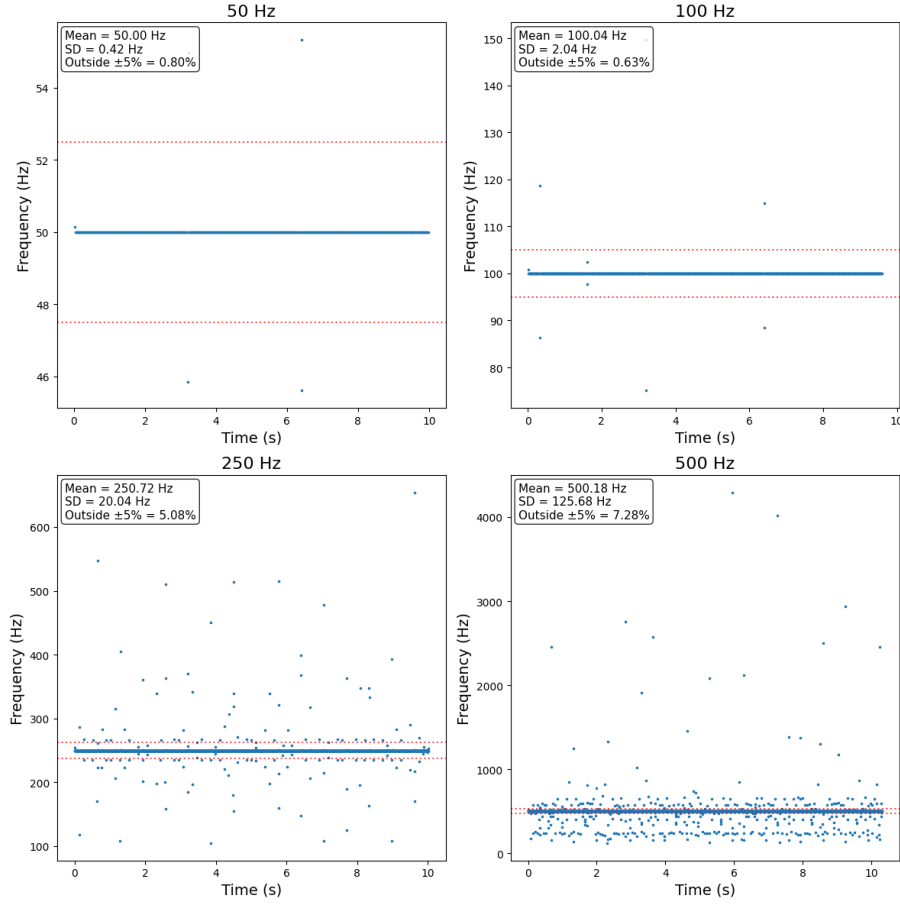


Figure 4.2: Sampling frequencies of the system when writing the data over to a locally connected microSD card for a desired frequency of 50 Hz, 100 Hz, 250 Hz, and 500 Hz. The box in the north-west corner of each plot presents the mean sampling frequency, standard deviation, and proportion of samples outside of the $\pm 5\%$ marks (red dotted line).

4.2 Model Accuracy and F1-score

The performance of the models were evaluated offline using accuracy and F1-score metrics, as well as the average time consumption for classifying one window. The model that performed the best in regards to these conditions was then evaluated during pseudo-live conditions. These tests were conducted on pre-recorded data, but the pseudo-live tests were designed to better simulate real-time usage conditions as described in section 3.8 and to test the model on completely unseen data. This model was also evaluated through LOSO cross validation to test its generalizability. The results from these three approaches are presented separately in the three following sections.

4.2.1 Offline Performance

In table 4.1 below, the overall accuracy and macro F1-score of the three trained networks are presented. All networks achieved an accuracy above 99% and macro F1-score above 0.99 on the training, validation, and test datasets.

	Training data		Validation data		Test data	
	Accuracy (%)	macro F1	Accuracy (%)	macro F1	Accuracy (%)	macro F1
CNN-LSTM	99.56	0.9959	99.65	0.9966	99.48	0.9952
LSTM-CNN	99.72	0.9955	99.65	0.9970	99.51	0.9974
Parallel	99.59	0.9961	99.61	0.9962	99.40	0.9945

Table 4.1: Accuracy and macro F1-score on the training, validation, and test datasets for the three models.

Table 4.2 below presents the time consumption for the three different models when classifying one window with a size of 1125 samples on a MacBook Pro M1. The CNN-LSTM network achieved the lowest mean classification time of 2.29 ms. The longest time it took to classify one window was 7.38 ms and the shortest time was 2.13 ms. Considering the reduced computational costs and that the accuracy and macro F1-score for this network was in range with the other two, the decision was made to further evaluate this network, and discard the other two. The class-specific accuracy and F1-score for the CNN-LSTM model on the datasets can be found in appendix B.1 and a confusion matrix of its performance on the test dataset can be found in appendix B.3.

	Mean time $\pm$ SD (ms)	Max time (ms)	Min time (ms)
CNN-LSTM	2.29 $\pm$ 0.36	7.38	2.13
LSTM-CNN	28.61 $\pm$ 2.08	54.58	27.77
Parallel	28.44 $\pm$ 2.69	60.03	27.53

Table 4.2: Mean, maximum, and minimum computational time of one window for each model on a MacBook Pro M1.

4.2.2 LOSO Performance

The performance of the CNN-LSTM model during LOSO cross validation can be seen in the box plot in figure 4.3. The mean $\pm$ SD and percentiles of the model accuracy can be found in table B.2 in the appendix. According to the box plot, the model acquired a test accuracy of above 98.6% for all participants.

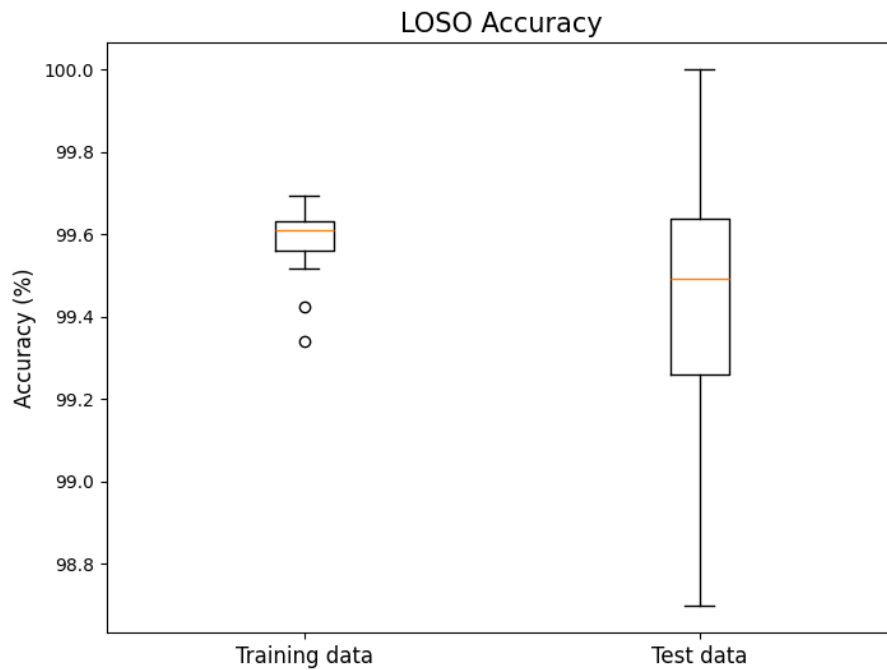


Figure 4.3: Training and test accuracy for the LOSO cross validation.

In figure 4.4 below the bar plot containing the train and test accuracies for each test participant from the LOSO cross validation is presented.

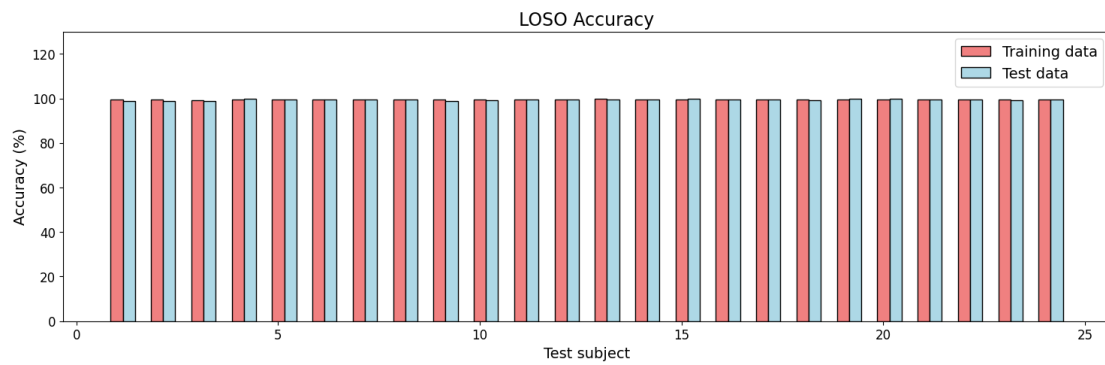


Figure 4.4: Training and test LOSO accuracy for each participant.

In table 4.3 the mean accuracy and F1-score of each iteration can be observed for each class. The model achieved a mean test accuracy over 99% for all classes across all participants and an F1-score above 0.99.

	Training data		Test data	
	Accuracy (%)	F1-score	Accuracy (%)	F1-score
Other	99.52 ± 0.11	0.9952 ± 0.0008	99.38 ± 0.49	0.9937 ± 0.0034
Heel-slide	99.56 ± 0.14	0.9953 ± 0.0019	99.35 ± 0.83	0.9929 ± 0.0100
Sitting Leg Raises	99.59 ± 0.08	0.9962 ± 0.0008	99.49 ± 0.77	0.9952 ± 0.0048
Lateral Leg Raises	99.69 ± 0.07	0.9968 ± 0.0007	99.51 ± 0.72	0.9948 ± 0.0043
Squat	99.71 ± 0.09	0.9971 ± 0.0007	99.36 ± 1.55	0.9941 ± 0.0091

Table 4.3: LOSO accuracy and F1-score per class across iterations for the training and test datasets.

4.2.3 Pseudo-live Performance

In table 4.4 below, the accuracy and F1-score of the CNN-LSTM network during pseudo-live conditions are presented. During this test, no post processing was applied and the test was conducted on the data collected from the three participants which were excluded from the training dataset population. Note also that the metrics have been calculated based on the execution criteria of the exercises.

The majority of the accuracies and F1-scores were above 99% and 0.99 respectively, with the exception of the “correct” and “less optimal” other class, “poor” heel-slides, and “correct” standing leg raises and squats. The lowest accuracy and F1-score was observed for the non-exercise label (“other”) during the recording of “correct” movement executions where the model got an accuracy of 98.24% and an F1-score of 0.9877.

	Correct		Less optimal		Poor	
	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score
Other	98.24	0.9877	98.87	0.9922	99.09	0.9927
Heel-slide	99.08	0.9940	99.29	0.9964	98.84	0.9925
Sitting Leg Raises	99.78	0.9955	99.89	0.9918	99.72	0.9947
Lateral Leg Raises	98.81	0.9859	99.78	0.9933	99.61	0.9936
Squat	100.0	0.9829	99.67	0.9934	99.94	0.9939

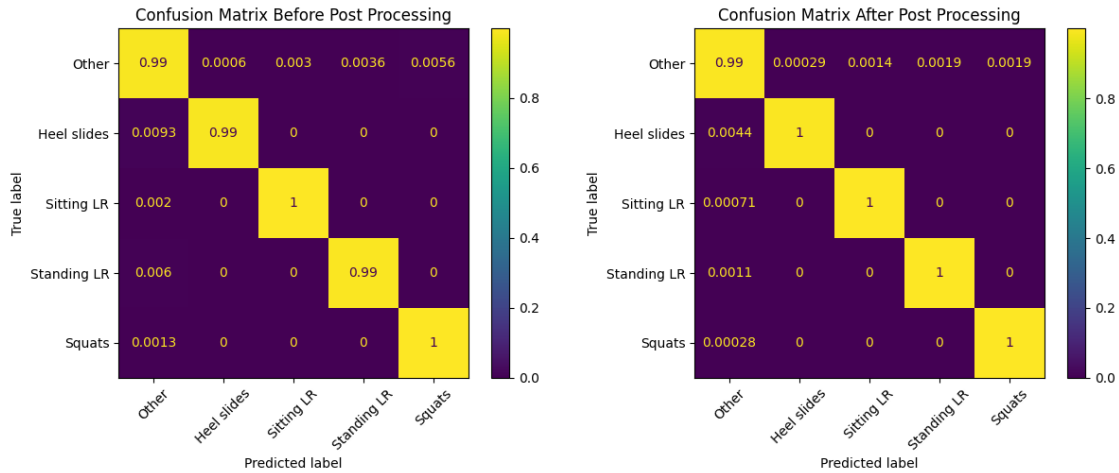
Table 4.4: Accuracy and F1-score for the exercises, separated into execution criteria before the post processing of the classification output.

In table 4.5 below the mean accuracy and F1-score for the CNN-LSTM network during pseudo-live conditions are presented. During this test the post processing was applied. All accuracies and F1-scores were above 99% and 0.99 across all classes and execution criteria.

	Correct		Less optimal		Poor	
	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score
Other	99.45	0.9964	99.42	0.9963	99.49	0.9962
Heel-slide	99.61	0.9974	99.67	0.9984	99.39	0.9961
Sitting Leg Raises	99.96	0.9983	99.96	0.9967	99.86	0.9970
Lateral Leg Raises	99.87	0.9953	99.91	0.9963	99.89	0.9972
Squat	100.0	0.9968	99.94	0.9966	99.98	0.9964

Table 4.5: Mean accuracy and F1-score for the movements, separated into execution criteria after the post processing of the classification output.

In figure 4.5 below the combined confusion matrices of the CNN-LSTM network is presented when evaluated during pseudo-live conditions. Figure 4.5a contains the confusion matrix before the post processing and figure 4.5b contains the confusion matrix after the post processing for all three recordings and test recordings. The confusion matrices have been normalized along the true label axis. It is possible to see that the percentage of misclassifications decreased after the post processing. An example of a segmented signal before and after the post-processing can be seen in figure B.4 in the Appendix.



(a) Confusion matrix before post processing **(b)** Confusion matrix after post processing

Figure 4.5: Confusion matrices for the CNN-LSTM network during pseudo-live tests, before and after post processing. Normalized along the true label.

4.3 Feature Extraction

This section presents the results obtained from the feature extraction analysis for all exercises. The estimated features that were evaluated are range of motion, knee drop, and movement smoothness through jerk-based measurements. The results for

the repetition counting algorithm are also presented. The sensor-derived estimates of the features have been visualized primarily using box plots to facilitate comparison between exercises and execution criteria.

4.3.1 Repetition Counter

The results for the repetition counter can be seen in table 4.6 where the repetition counter achieved an accuracy of 100% on both the original labelled data and the segmented data from the pseudo-live tests.

	Accuracy (%)
Original data	100.0
Segmented data	100.0

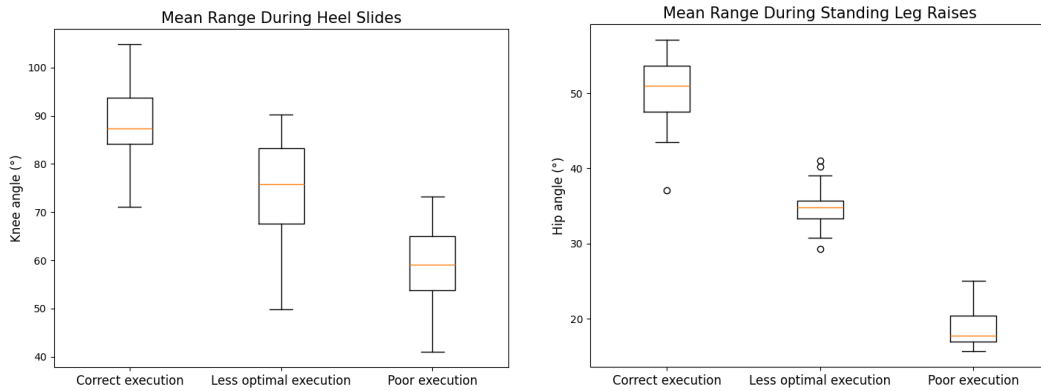
Table 4.6: The accuracy of the repetition counter on the original labelled data and the segmented data from the pseudo-live tests.

4.3.2 Range of Motion

In figure 4.6, the estimated mean range of motion for heel slides and standing leg raises are presented as box plots over the three execution criteria. The mean, standard deviation and percentiles for these boxes can be found in table B.3 in the appendix.

During heel slides (see figure 4.6a) the estimated median ROMs were around $\sim 87^\circ$, $\sim 76^\circ$, and $\sim 59^\circ$ for the “correct”, “less optimal”, and “poor” execution criteria respectively while the target angles were 90° , 70° and 50° .

During standing leg raises (see figure 4.6b) the estimated median ROMs were around $\sim 51^\circ$, $\sim 35^\circ$, and $\sim 18^\circ$ for the “correct”, “less optimal”, and “poor” execution criteria respectively while the target values were 45° , 30° , and 15° .



(a) The mean range of motion during heel slides for the three execution criteria.

(b) The mean range of motion during standing leg raises for the three execution criteria.

Figure 4.6: The estimated mean range of motion during heel slides and standing leg raises for the three execution criteria.

The estimated mean maximum and minimum rotation of the knee and hip joint during heel slides and standing leg raises can be seen in figure 4.7. The mean, standard deviation and percentiles for these box plots can be found in table B.4 in the appendix.

The estimated mean maximum knee angle for heel slides (figure 4.7a) are centred around $\sim 93^\circ$, $\sim 77^\circ$, and $\sim 59^\circ$ for the “correct”, “less optimal” and “poor” variations respectively. The target maximum knee angles were $\sim 90^\circ$, $\sim 70^\circ$, and $\sim 50^\circ$.

In figure 4.7b, the estimated mean maximum hip angle during standing leg raises are presented. The measurements are centred around $\sim 53^\circ$, $\sim 36^\circ$, and $\sim 19^\circ$ for the “correct”, “less optimal”, and “poor” variations respectively, while the expected target outcome were $\sim 45^\circ$, $\sim 30^\circ$, and $\sim 15^\circ$.

Figure 4.7c shows the estimated mean of the minimum knee angle for heel slides. In this figure the medians are centred around $\sim 6^\circ$, $\sim 4^\circ$, and $\sim 0^\circ$ for the “correct”, “less optimal” and “poor” variations respectively. The expected target minimum knee angle was $\sim 0^\circ$ for all three execution criteria.

The estimated mean of the minimum hip rotation for the standing leg raises can be seen in figure 4.7d. The measurements are centred around $\sim 1^\circ$ for the “correct” execution criterion and $\sim 0^\circ$ for the “less optimal” and “poor” execution criteria and the expected target outcome for all three were $\sim 0^\circ$.

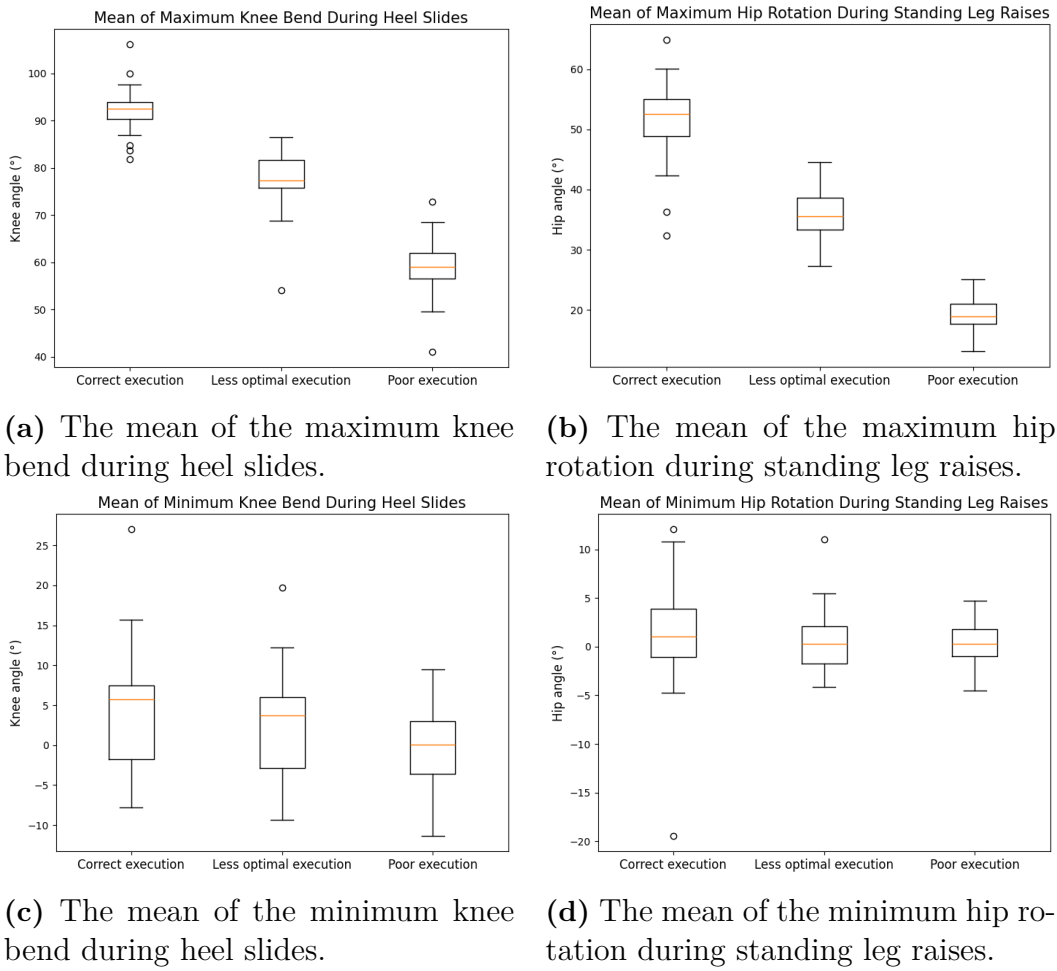
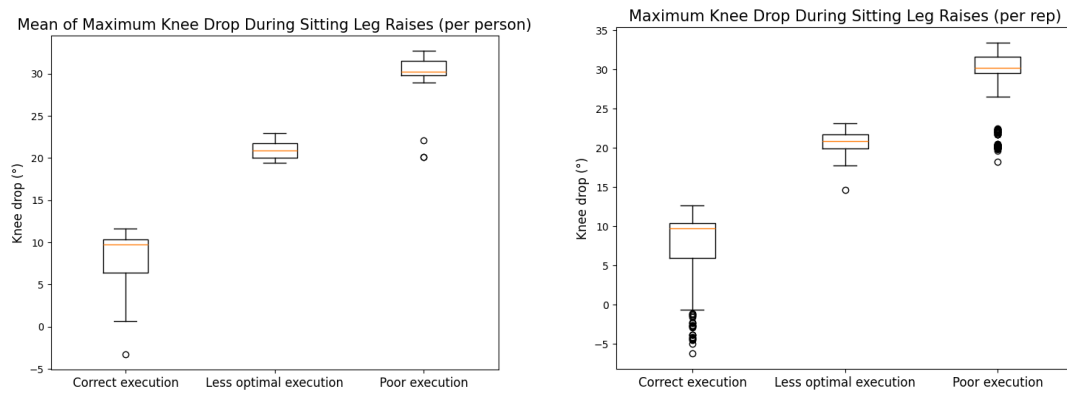


Figure 4.7: Mean of maximum and minimum knee bend during heel slides, and hip rotation during standing leg raises, for the three execution criteria.

4.3.3 Knee Drop

In figure 4.8 below, two box plots for the estimated knee drop during sitting leg raises are presented. Figure 4.8a shows the mean of the maximum knee drop over all repetitions for the execution criteria separately, while figure 4.8b shows the estimated maximum knee drop for all repetitions. The mean, standard deviation and percentiles of these figures can be found in table B.5 in the appendix. In both figures the medians are centred around the angle of which the orthosis was locked during the different execution criteria, i.e. $\sim 10^\circ$ for the “correct” execution criterion, $\sim 20^\circ$ for the “less optimal” execution criterion, and $\sim 30^\circ$ for the “poor” execution criterion.



(a) The mean of the maximum observed knee drop from each participant. (b) The maximum knee drop observed from all repetitions.

Figure 4.8: The maximum knee drop over repetitions observed for all participants during the three execution criteria.

The estimated knee drop during squats and the total knee rotation proxy in the transverse plane can be seen in figure 4.9. The mean and standard deviation and percentiles of these figures can be found in table B.6 in the appendix.

The estimated mean of the maximum knee drop for each repetition during squats can be seen in figure 4.9a for each participant depending on the execution criterion. In this figure the median for each execution criterion can be seen to decrease as the participants were instructed to increase how much they dropped their knees. The standard deviations for the boxes increased for the “less optimal” and “poor” variations. The maximum knee drop for all repetitions can be seen in figure 4.9b.

The estimated total rotation of the knee during the squats can be seen in figure 4.9c and 4.9d where both the external and internal rotation have been taken into account. The mean was calculated for the total rotation during each repetition and can be seen in figure 4.9c for each participant. The total rotation for all repetitions can instead be seen in figure 4.9d. In both figures all percentiles increased as the intentionally simulated deviation increased as described by the execution categories.

4. Results

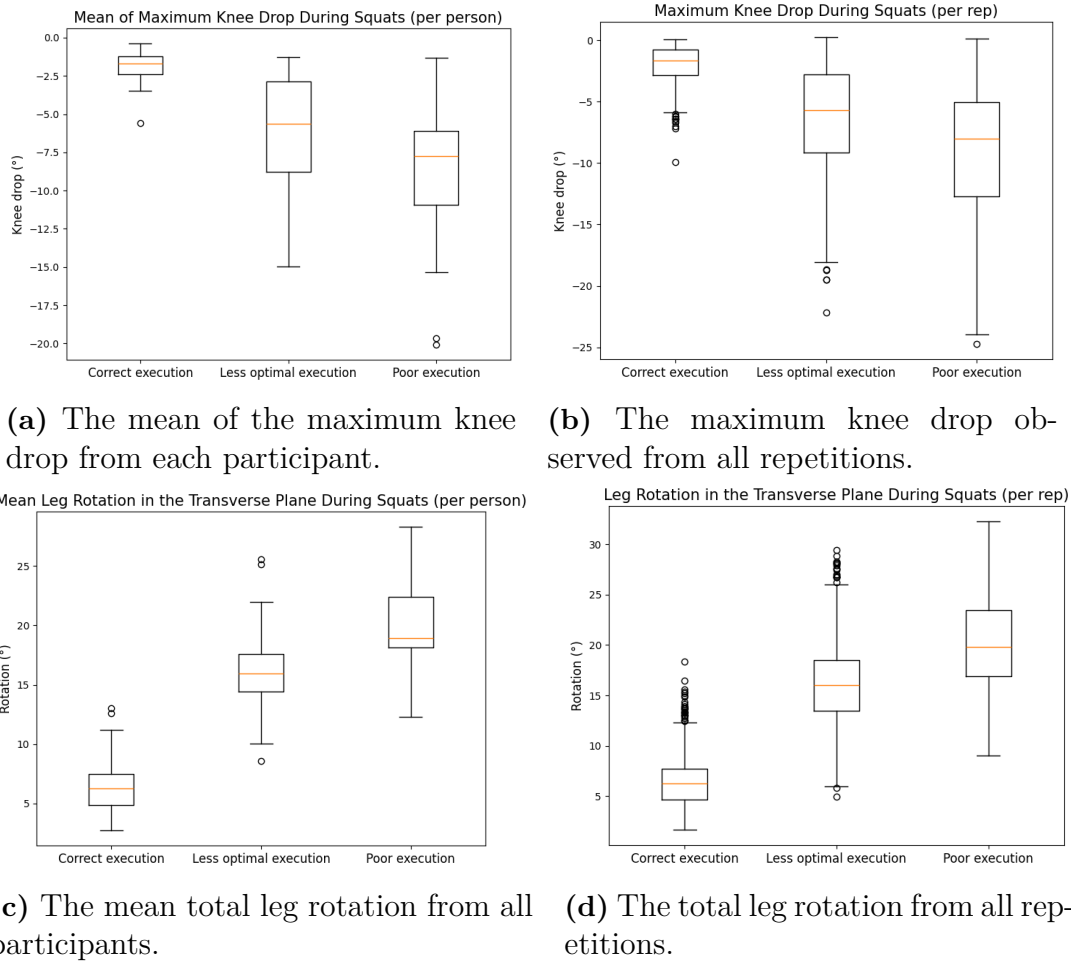
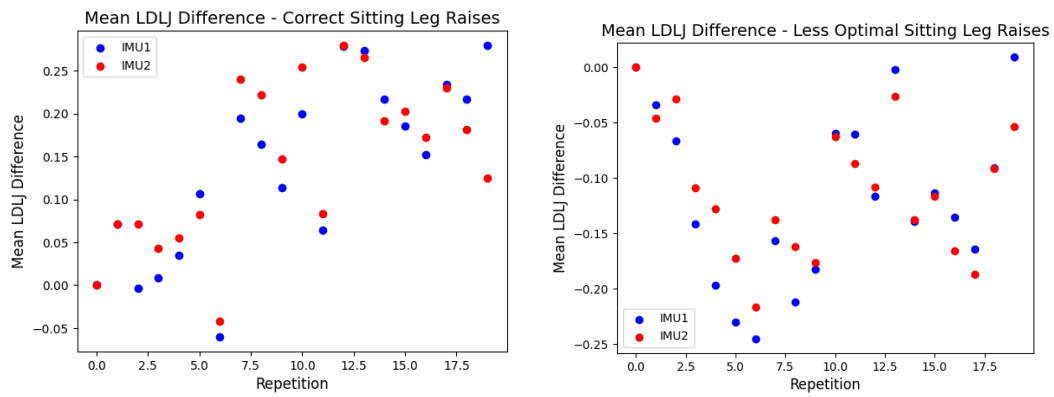


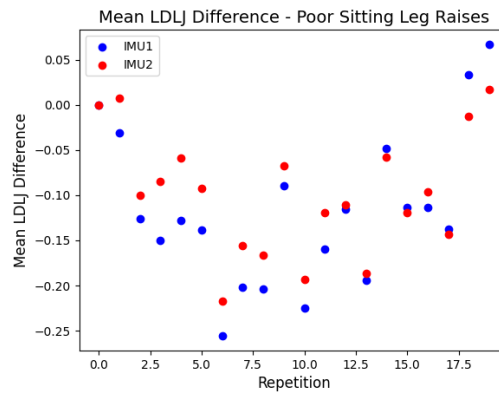
Figure 4.9: Knee drop and total leg rotation during squats depending on the variation of the performed movement.

4.3.4 Movement Smoothness and Jerk

In figure 4.10 below, the difference in mean of LDLJ for sitting leg raises is presented as scatter plots for the different execution criteria. For the “less optimal” and “poor” executions (figure 4.10b and 4.10c) the mean difference of LDLJ decreased over the first repetitions, and during the final repetitions it increased again. For the “correct” execution criterion (figure 4.10a) the mean difference in LDLJ instead increased over repetitions.



(a) Difference in mean of LDLJ for the “correct” sitting leg raises. (b) Difference in mean of LDLJ for the “less optimal” sitting leg raises.



(c) Difference in mean of LDLJ for “poor” sitting leg raises.

Figure 4.10: Difference in mean of LDLJ for sitting leg raises during the different execution criteria.

In figure 4.11 below, the difference in mean of LDLJ for squats is presented as scatter plots for the different execution criteria. For all execution criteria the movement smoothness decreased as the number of repetitions increased.

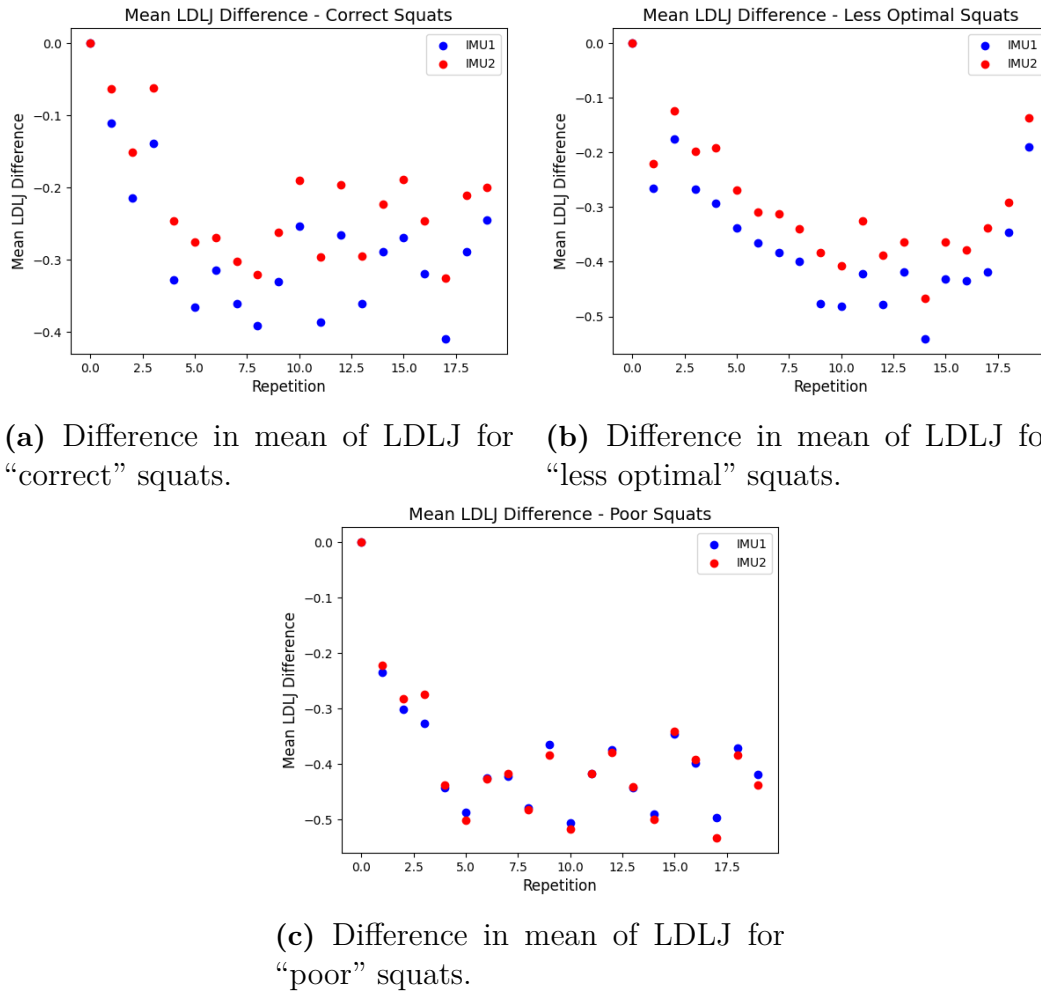
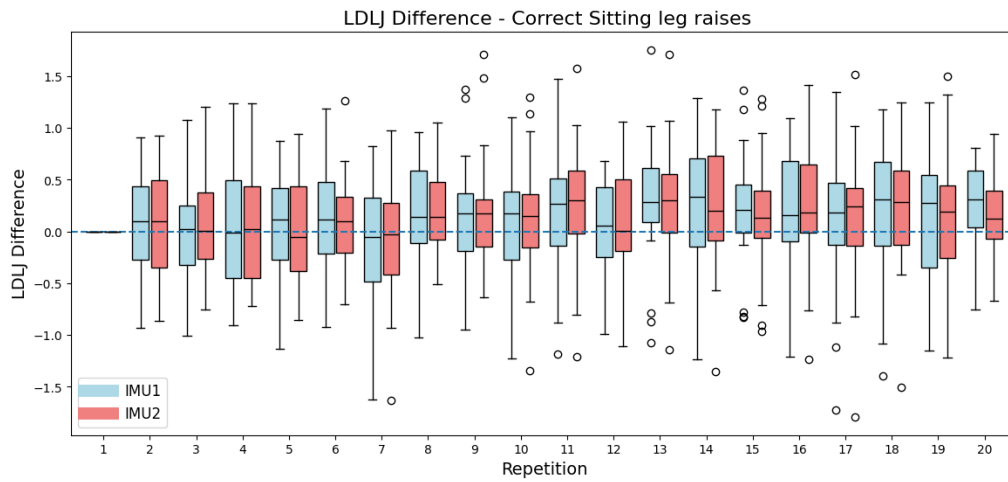
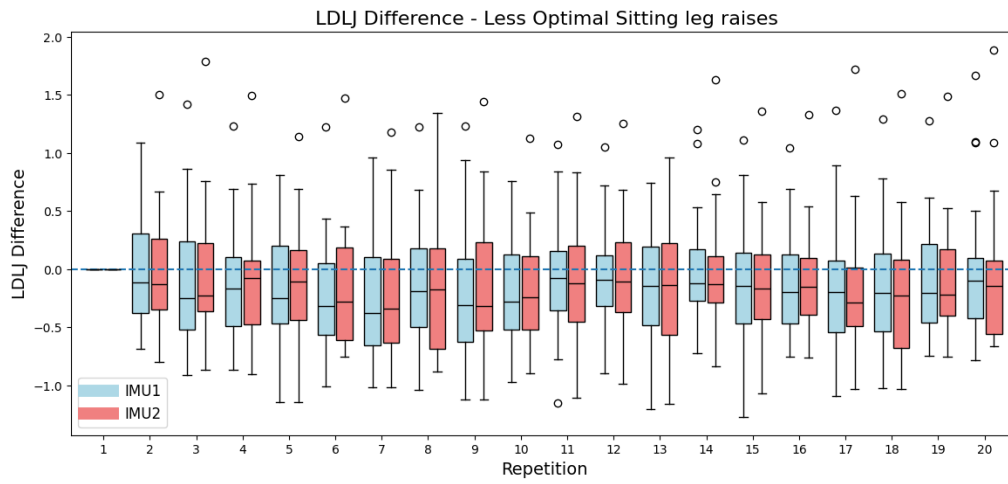


Figure 4.11: Difference in mean of LDLJ for squats during the different execution criteria.

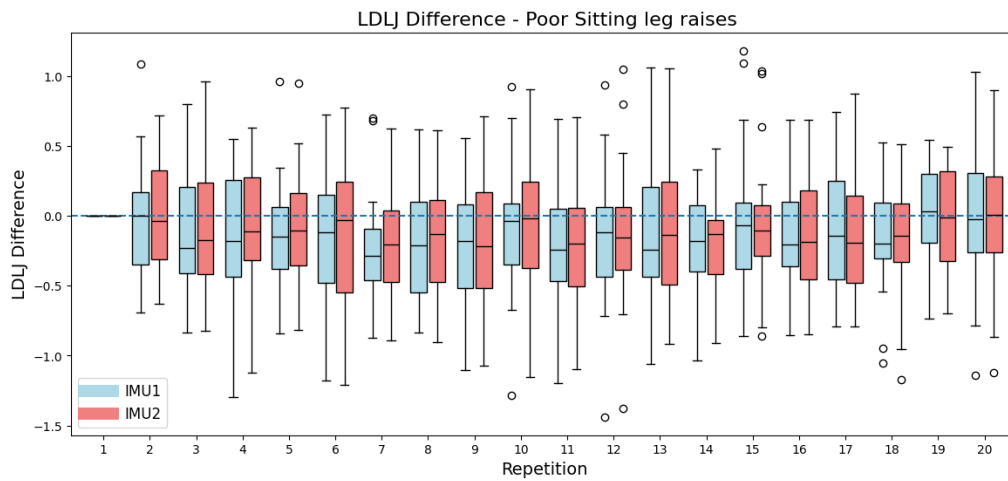
In figure 4.12 below, the box plots of the difference of LDLJ are presented for the three execution criteria for sitting leg raises. The smoothness of the exercise decreased with the amount of executed repetitions for the "less optimal" and "poor" execution criteria (figure 4.12b and 4.12c). This is especially noticeable for quartile 2 and 3 (P25-P75) while quartile 1 and 4 were more noisy. When looking at the box plot for the "correct" execution criteria for sitting leg raises (figure 4.12a) the smoothness increased with the amount of repetitions. This is especially noticeable in quartile 2 and 3 while being more noisy in quartile 1 and 4. To see the results of the other movements, please see figure B.7 and B.8 in the Appendix.



(a) Difference of LDLJ for “correct” sitting leg raises.



(b) Difference of LDLJ for “less optimal” sitting leg raises.

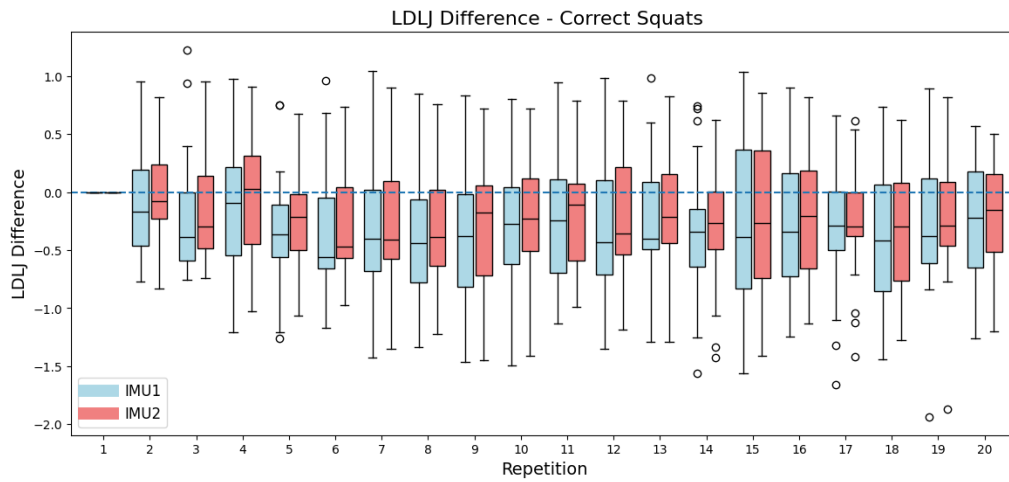


(c) Difference of LDLJ for “poor” sitting leg raises.

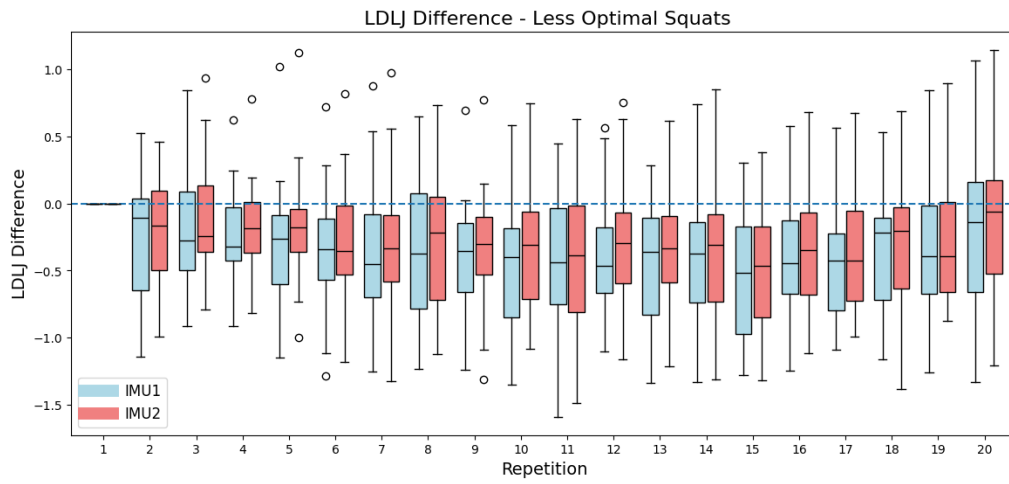
Figure 4.12: Difference of LDLJ for sitting leg raises during the different execution criteria.

4. Results

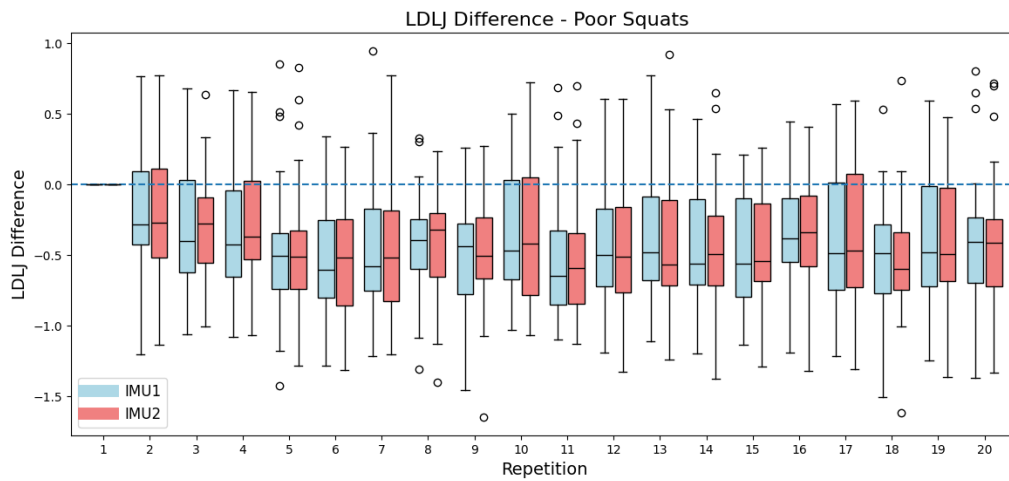
In figure 4.13 below, the box plots of the difference of LDLJ are presented for the three execution criteria for squats. The smoothness of the exercise decreased with the amount of executed repetitions for all the execution criteria. This is more visible for quartiles 1, 2, and 3 (P0-P75). It is also possible to see from the box plots that the smoothness seems to have decreased between the three execution criteria.



(a) Difference of LDLJ for “correct” squats.



(b) Difference of LDLJ for “less optimal” squats.



(c) Difference of LDLJ for “poor” squats.

Figure 4.13: Difference of LDLJ for squats during the different execution criteria.

4.4 Supplementary Analyses

This section presents the results from the supplementary analyses. The performance of the exploratory extended execution-criterion classification model is presented through accuracy, macro F1-score, and confusion matrices, while the results from the exploratory LDLJ evaluation are presented in a scatter plot, similarly to the ones found in figure 4.10 and 4.11.

4.4.1 Execution-Criterion Classification

Table 4.7 below shows the accuracy and macro F1-score for the LSTM-CNN model when trained on all movements and their corresponding execution criteria (13 classes in total).

	Training data		Validation data		Test data	
	Accuracy	macro F1	Accuracy	macro F1	Accuracy	macro F1
CNN-LSTM	95.78	0.9346	95.82	0.9343	96.33	0.9406

Table 4.7: Accuracy and macro F1-score for the LSTM-CNN model when trained on all exercises and their execution criteria.

The confusion matrices below (figure 4.14 and 4.15) show the results from testing the LSTM-CNN model on classifying the movements and their execution criteria. The matrices have been normalized along the true label axis. The abbreviations C, LO, and P stand for “correct”, “less optimal”, and “poor” respectively. The model managed well on classifying most labels correctly. The class with the lowest accuracy was the “less optimal” variations of squats.

It is noticeable from figure 4.15a that the model did not confuse any exercises with each other. From figure 4.15b it is also possible to see that the model managed to classify the execution-criteria well. It did however confuse some execution criteria, mainly neighbouring criteria; that is “correct” executions with “less optimal” executions, and “less optimal” executions with “poor” executions, and vice versa.

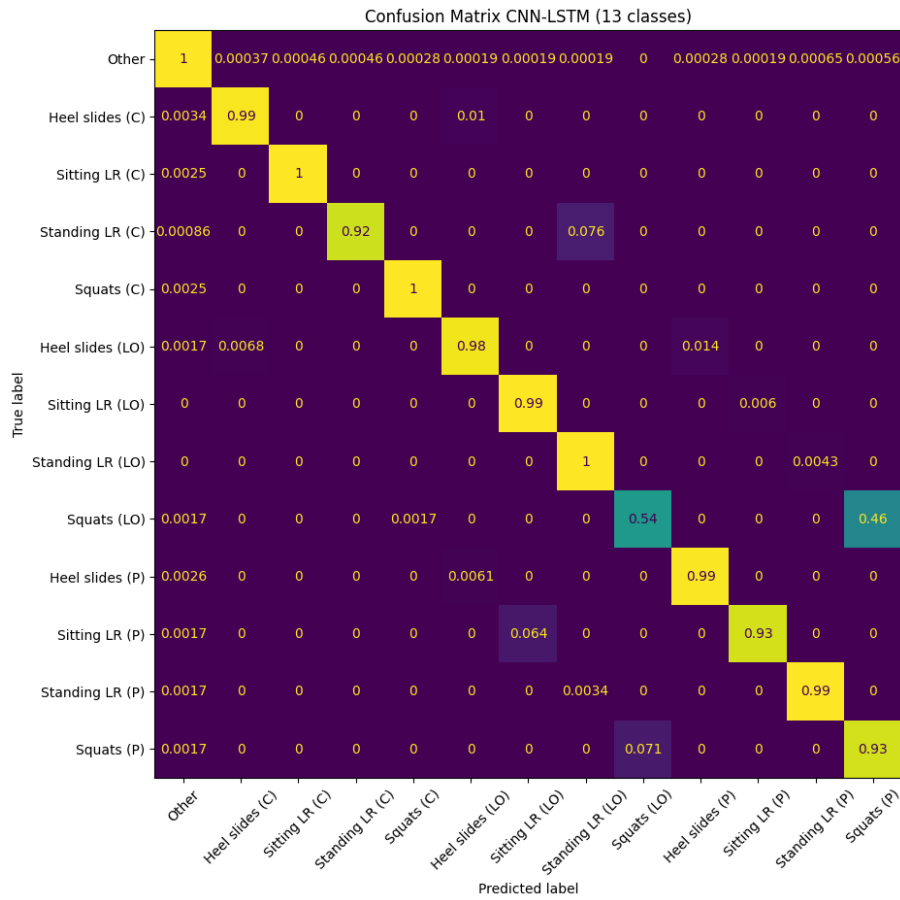
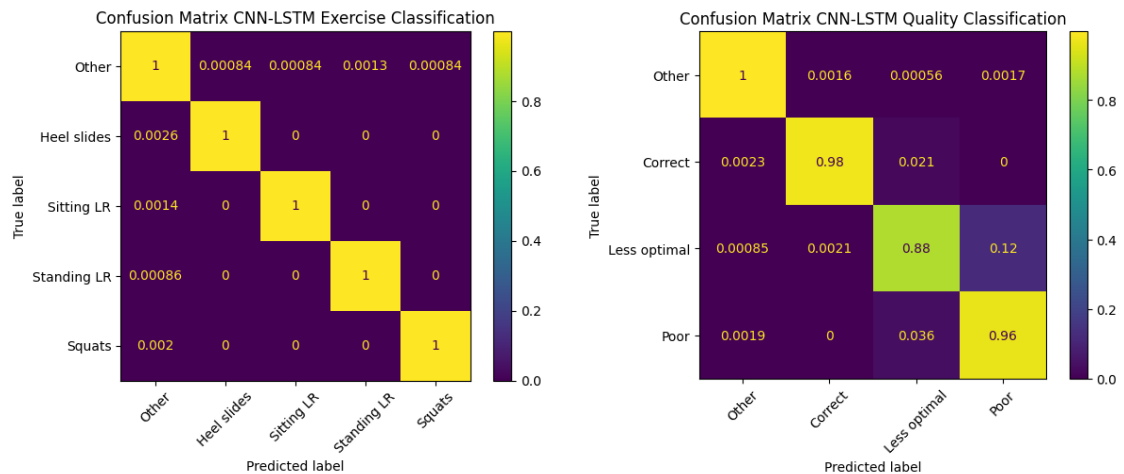


Figure 4.14: Model performance on the test dataset, for the 13 class classification model (C = correct, LO = less optimal, P = poor).



(a) Confusion matrix for the classification of exercises.

(b) Confusion matrix for the classification of execution criteria.

Figure 4.15: Confusion matrices for the 13 label classification model separated based on exercises and execution criteria.

4.4.2 Exploratory LDLJ Evaluation

Figure 4.16 below shows the scatter plot of the mean difference in LDLJ for the “correct” execution variation of sitting leg raises during the additional pilot study.

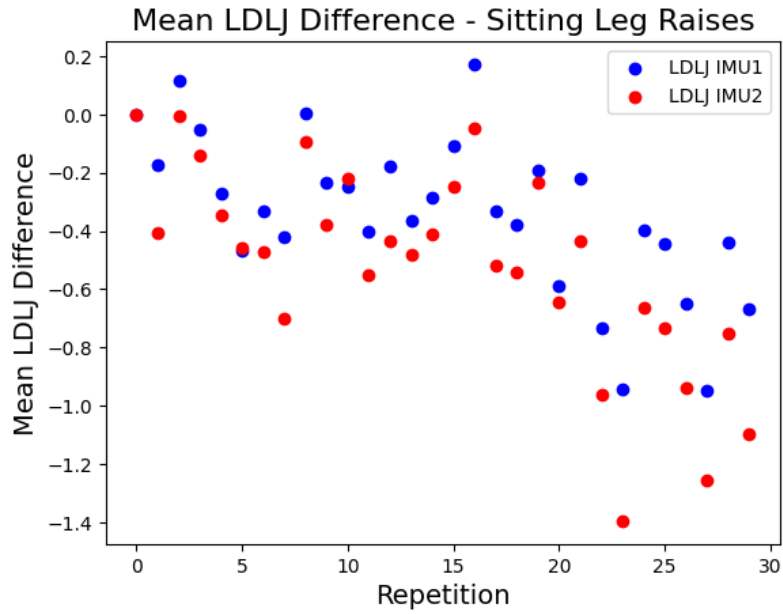


Figure 4.16: Mean difference of LDLJ for sitting leg raises when the orthosis was unlocked.

5

Discussion

One of the most important aspects that need to be taken into account when evaluating this system is how it may be used in the future. It could be used in further movement analysis studies, or it could be used for gathering information about performed movements for offline review of the movement recordings. Another potential usage area is for the system to be used in combination with an app to provide live training-feedback. Lastly it could also be used in VR or AR or even as a tool in serious gaming where the user can control the game with the orthosis to for example help them train certain movements in a controlled environment. All these usage areas will entail different system and model requirements. It is from the perspective of offline review of the movement recordings that the model and feature extraction has been designed, but the other usage areas will also be discussed in this chapter.

5.1 Limitations

While the results from this study are promising, several important limitations need to be considered when interpreting the findings and evaluating the feasibility of the proposed system. These limitations affect both the generalizability of the results and the interpretation of extracted biomechanical features. The following sections summarizes the most important limitations of this project.

5.1.1 Sample Diversity and Generalizability

One major limitation of this study is that the dataset only consisted of movement data from healthy young adults who performed simulated exercises, under controlled conditions. Although movement variations were deliberately introduced during the recordings, these movements do not necessarily represent typical movement characteristics for individuals who naturally perform these variations. Therefore the results can not be generalized beyond the performance of healthy individuals during controlled motion-analysis.

The results presented in this report should instead be viewed as a biomechanical proof-of-concept which demonstrates the possibilities and limitations for capturing of interesting features and classifying movements under controlled experimental settings.

5.1.2 Missing Ground Truth for Feature Extraction

Another important limitation is the lack of an external ground truth for the evaluation of extracted biomechanical features, with the repetition counter being the sole exception. In the context of this report, a ground truth is a directly measured, or verified value that represents the absolute true state of a given metric. For instance, the exact joint angle of the hip defining the range of motion during standing leg raises could ideally be quantified using state-of-the-art systems, such as motion capture.

The lack of a ground truth introduces challenges in directly verifying the accuracy of the extracted features. As a consequence, the analytical focus must shift from absolute quantification to trend analysis. By evaluating relative changes, consistencies and patterns visible in the data, it is still possible to draw meaningful conclusions about the challenges and possibilities for extraction of features from the data collected with the system.

5.1.3 Global Coordinate System Assumptions and Drifts

Throughout this project certain assumptions had to be made regarding how the participants move in unmeasured coordinate systems. The range of motion for the standing leg raises were calculated using the second IMU since no IMU was positioned at the hip. This assumption works only if the participant does not rotate their leg so that their right foot is pointing along the motion trajectory during the movement execution. While for the calculations of knee drop during squats the rotation around the global coordinate system was estimated as the rotation around the first IMUs y-axis. However, during squats it is common to lean forwards to maintain balance which moves the y-axis out of alignment with the axis of the global coordinate frame.

With larger deviations from the true axis of rotation greater errors will be introduced to the angle calculations. It was observed throughout this project that when the misalignment increased the accumulated integral errors would also increase which introduced a drift to the calculated angle. This drift was stronger and more common for the calculations of knee drop during squats than for the calculations of range of motion during standing leg raises. When also taking the natural gyroscopic drift into account the decision had to be made to calculate the knee drop for each repetition separately to minimize the effect of the drifts. In contrast, the effect of the accumulated integral error and the gyroscopic drift was relatively small for the range of motion estimations for the standing leg raises. The decision was therefore made to calculate the range of motion for the entire movement in the same way as for the heel slides. This made it possible to estimate and evaluate the maximum and minimum abduction of the hip joint during standing leg raises.

To minimize the accumulated integral errors it would be necessary to use an axis that is more representative of the true rotational axis. For both the range of motion and for the knee drop a possible solution would be to use the coordinate system in the beginning of the repetition as a reference and calculate the rotation around

the z-axis or y-axis of this reference system. However, this reference could still be incorrect if the participant is not in the proper starting position at the beginning of the repetition. A much more robust method for the knee drop would be to exchange the IMU sensors for 9-axis IMU sensors that include a magnetometer which would allow for calculations of the rotation around the true global axis.

5.2 Sampling and Transferring of Data

In the earlier projects conducted by both Zhou and Shen the decision was made to utilize two different SPI buses for the transferring of data between the sensors and the microcontroller. This decision was made to minimize the risk of data transmission conflicts. As a result the two IMU sensors share one SPI bus while the angle encoder and the microSD module share a second SPI bus. However, as can be seen by the results in figure 4.2 this decision resulted in the sampling frequency not being completely stable when sending the data to the microSD card locally connected to the device. The sampling frequency over time contains outliers that are both a lot higher and lower than the desired frequency. This phenomenon coincides with the writing to the microSD card. Since the angle encoder and the microSD module utilizes the same SPI bus it is impossible to collect data from the angle encoder while writing information to the microSD. This results in sampling delays which can be seen in the results as decreases and increases in sampling frequency.

One of the benefits of the ESP32-S3 Feather is its double core, which allows it to perform two tasks at the same time. One common area where this is utilized is for the sake of data collection, where one core would be responsible for reading the sensors and the other for packaging and storing the data. The current design of the system fails to utilize this functionality since the shared SPI bus makes it impossible to separate the writing to the microSD from reading the angle encoder. The drops in frequency that can be observed in figure 4.2 were deemed detrimental for this project since the frequency drops to approximately 100 Hz and lower would result in sections of 10 ms or longer with missing data. As a result the decision was made for this project to send the data directly to the computer using the ESP32-S3 Wi-Fi hotspot instead of saving it to the microSD which allowed for the utilization of the dual core.

In the future one possible goal for the system is for it to continuously record and send data live for later review of movement characteristics. Therefore, one possible area of use for the microSD card could be as a backup if the orthosis ever loses signal and fails to send it to the appropriate server, as was discussed in both Zhou's and Shen's reports [5, 6]. Based on this it would be of interest to investigate in a future project if it is more important for the angle encoder and microSD module to share the same SPI bus to minimize the risk of transmission conflicts, or to separate the two to make it possible to utilize the microcontroller's dual core while saving data to the microSD.

When collecting data using the Wi-Fi connection the sampling frequency becomes more stable. This is apparent when comparing the standard deviation and propor-

tion of outliers presented in figure 4.1 and 4.2, which strengthens the decision of utilizing UDP during this study. The deviations that are still present while utilizing UDP could be caused by normal data acquisition latencies of the sensors.

While the UDP protocol enables fast streaming of data it also comes with the risk of data loss since it does not check if the data has been received. During all recordings for this thesis no packages were dropped but the usage of UDP needs to be evaluated for future projects. Another possibility is to use a Wi-Fi connection utilizing TCP which does not risk data being lost or corrupted. However, this method is slower than UDP which could result in delays depending on how fast the system is collecting and transmitting data, and how fast the receiver can process incoming data. In a worst case scenario this could result in sender and receiver buffers being filled and blocking any new data from being transmitted. Therefore, further tests will need to be conducted to determine the feasibility of data transmission protocols depending on the future usage areas of the system.

5.3 Classification

During this project three classification models were designed and evaluated on the data collected through the study. All models performed well in regards to accuracy and F1-score (see table 4.1), with the LSTM-CNN network performing slightly better than the other models on all three datasets. When comparing the computational time for the models (see table 4.2), the CNN-LSTM network instead outperformed the other networks. These results indicate that a trade-off has to be made in regards to accuracy and time consumption.

5.3.1 Model selection

All networks performed well with regards to classification accuracy. However, it is important to highlight that the accuracies presented in table 4.1 could be inflated by data leakage. These accuracies were therefore only used for evaluation of which of the three models would be further analysed on completely unseen data. To determine the best model, different usage areas needed to be taken into account. Depending on the future usage area of the product it could become important to determine if the slightly higher accuracies of the LSTM-CNN network is worth the extra computational time. If the model is to be used for offline review of the movement recordings the higher computational times could be acceptable since no live transmission of data is needed. However, in live-feedback and serious gaming applications, the demands on computational efficiency increases. During gaming it can be necessary for a classification to occur every 10-100th ms, depending on the game requirement. Thus, the LSTM-CNN and the parallel network could have a too high computational time for them to be applicable for live-feedback and gaming.

Two other aspects that need to be taken into account when discussing future usage areas are what sample in the sliding window the model needs to classify to minimize delays, and on what hardware all computations will occur. All the models were

trained on classifying the middle pixel of each window, which will subsequently introduce a latency of half a window ($\sim 2.25s$) in the classification. This is the most optimal pixel to classify for segmentation since the model will have access to both preceding and succeeding samples. For the application of offline review of the recorded movements this delay was deemed acceptable since the main requirement of this usage area is high quality segmentation and feature extraction rather than live-feedback.

However, if the model is to be used for live-feedback or serious gaming the classified sample would likely need to be moved closer to the end of the window to minimize delays. This would probably result in the model accuracies and F1-scores decreasing since the model would have less access to future data when making its predictions, which could negatively affect the performance of the LSTM, since it learns temporal dependence within the data. Consequently, this could also negatively affect the segmentation results by causing delayed detection of the start and end of a movement.

Additionally, if the classified sample is shifted to the very end of the window the model would become a prediction model rather than a classification model, which would not be optimal for accurate movement segmentation. In this case further tests would need to be conducted to determine the optimal position for the pixel to maximize the classification accuracy while minimizing the delay.

As mentioned earlier it is also important to consider what hardware all computations will realistically occur on depending on the future usage areas. All presented computational times in table 4.2 were obtained on a laptop, however, depending on the usage area it could be more realistic for the computations to occur on a different hardware. If the computations were made locally on a phone or remotely on a server the computational times would likely differ from the ones computed in this project. For offline review of the recorded movements, the computations would likely occur on a remote server, while it would be more feasible for the data to be processed on a phone or computer during the live-feedback and gaming applications. Based on this the decision was made to continue with the CNN-LSTM network since fast computational times are more important than a slightly higher accuracy for two out of three potential future usage areas.

5.3.2 LOSO results

In addition to the train/validation/test split evaluation, the model was also evaluated using the LOSO cross-validation method. The results from this evaluation (seen in figure 4.3 and 4.4, and table B.2) shows that the accuracy on data from unseen participants is still high and in range with the accuracy on the training data. This indicates that the model was not overfitted to the training dataset, and that the model was able to generalize to participants not included in the training. Furthermore, this suggests that the CNN-LSTM model did not only learn participant-specific movement patterns, but also managed to identify more general movement characteristics.

When evaluating the accuracy and F1-score for all the different movements during LOSO cross-validation, it is possible to see that the model achieves an accuracy of above 99% for all classes. However, squats had the largest standard deviation of the test accuracy while the heel slides had the largest standard deviation of the F1-score for the test set. There could be multiple reasons for these misclassifications. Since the participants were allowed to move freely when transitioning between movement tasks, some transition periods may have contained motion patterns that resembled the predefined movement classes. For example, sitting down in preparation for heel slides or sitting leg raises, as well as standing up after completing these movements, may have produced sensor signals similar to those observed during heel slides or squats. As a result, some transition segments may have been assigned to neighbouring movement classes rather than being classified as non-exercise movements.

When comparing these results to those obtained in the prior study reviewed in section 2.1.1, the network implemented in this study demonstrated a substantially higher classification accuracy. Several factors likely contribute to this performance disparity; most notably, the limited dataset size in previous research, the increased complexity of the network architecture utilized in this work, and the differences in the specific types of movements classified across the respective studies.

5.3.3 Pseudo-live results

As mentioned previously, a future goal of the system is for the classification of movements to be performed on live fed data. To evaluate the possibilities of using the system for such applications, the CNN-LSTM model was also tested during pseudo-live conditions. The results presented in table 4.4 suggest that the model might be suitable for this application since it achieved a pseudo-live classification accuracy close to 99% or more for all classes. Based on these results it is also apparent that the data collected from the orthosis is informative enough to allow for accurate classification and segmentation of the four movements. The confusion matrix in figure 4.5a shows that the only misclassifications that occurred were between exercise classes and the non-exercise (“other”) class, while no exercises were confused with one another. This indicates that the model managed to find distinctions between the different movement patterns within the exercises.

The misclassifications involving the “other” class were expected since participants were allowed to move between exercises in a natural manner to simulate normal transitions and non-exercise behaviour. As discussed earlier, this might have caused normal movements to be confused with the predefined classes (see example of this in figure B.4 in the Appendix). Another likely source of misclassification is the transition periods between exercises and non-exercise movements, which occurred at the start and end of each exercise. Since the data was labelled manually, small mismatches between label assignment and true movement initiation most likely occurred, resulting in false positives and false negatives at exercise boundaries.

The classification accuracy and F1-score after post-processing is presented in table 4.5. After post-processing the accuracy and F1-score increased across all classes

which can also be observed in the confusion matrix in figure 4.5b. These results suggest that the proposed post-processing was effective in increasing the classification accuracy during pseudo-live conditions. Furthermore, the results indicate that the collected sensor data was robust enough, not only for accurate offline classification, but also for continuous monitoring of exercises during pseudo-live conditions.

Even though the post-processing steps increased the overall accuracy and F1-score, it is important to have in mind that the usefulness of post-processing is strongly related to the intended usage area of the system. For offline review of the recorded movements, accurate segmentation is crucial since the subsequent feature extraction relies on correctly identified exercise boundaries. Therefore, the additional processing delays introduced by the post-processing would be acceptable. However, if the system would be used for serious gaming, where responsiveness and low latency is prioritized, the post-processing would not be applicable since it requires fully segmented exercise sections which introduces a latency of an entire exercise and five seconds. For 20 repetitions with a metronome at 40 BPM as in this study, it would mean that the total latency would be approximately 60 seconds plus five seconds and additional computational times. Therefore, this method can not be applied for live classification in serious gaming.

For serious gaming applications, the most relevant performance metrics would instead be the results from the pseudo-live tests without applying post-processing shown in table 4.4. It is however, important to keep in mind that the model was classifying the middle pixel of each window as was discussed earlier. For gaming applications it would be of higher interest to move this pixel closer to the end of the window to minimize latency since the current latency of 2.25 seconds is too long for gaming. As mentioned, this would decrease the accuracy of the model and possibly introduce delays in the detection of the start and end of an exercise. This could increase the misclassifications of the “other” class further.

What is left to be evaluated is whether or not the remaining misclassifications of the “other” class in table 4.4 will have too big of an impact on the gaming experience or not. Depending on where the misclassifications occur it will affect the user experience to different degrees. If the majority of the errors are concentrated close to exercise boundaries due to uncertainties during labelling, the impact on gameplay may be limited. Furthermore, to make a serious game more resilient to misclassification it is possible to implement inertia in the game, meaning that a label needs to be classified multiple times in a row before it has its full effect on the game. By doing so it will be possible to reduce the impact of occasional misclassifications that otherwise could ruin the immersion and user experience. This implies that the model and system would probably be applicable in serious gaming, but further tests would need to be performed in future studies.

5.3.4 Execution-Criterion Classification

Since the models performed as well as they did on classifying the movements, it raises the question of whether or not the collected sensor data contains enough information

for a model to classify the execution criteria of the performed exercises as well. This type of functionality could potentially be valuable, since machine learning models may be capable of finding subtle patterns and relationships within the movement data which are hard for humans to detect manually. A model which is capable of providing live feedback in regards to movement execution and training progress may support future non-medical training-feedback applications after further validation.

To investigate this possibility, the decision was made to train an additional CNN-LSTM on classifying both the exercises type and the execution criterion simultaneously. It is however important to note that these tests are only exploratory and that the overlapping windows between train/val/test could inflate the accuracies. These tests were solely performed to evaluate if it would even be possible to train an execution-criterion classification model for future studies.

The results in figure 4.14 and 4.15 shows that the model was still capable of distinguishing the exercises from each other, while simultaneously classifying most of the execution criteria correctly. The model managed to correctly identify more than 88% of the true exercise criterion of each class which shows promise in regards to the data containing meaningful information related to the variations in the exercise execution. The confusion matrices also reveal that most misclassifications occurred between neighbouring execution variations, i.e. between “correct” and “less optimal”, and “less optimal” and “poor”. This behaviour was expected since the border between different execution criteria is not discrete, but rather gradual. Consequently, movements categorized as “less optimal” may share characteristics with both the “correct” and the “poor” variations.

Of all the exercises, the model seemed to struggle the most with classifying “less optimal” squats where 46% of windows are being misclassified as “poor”. One possible reason for this could be that no markings or physical restraints were used to indicate how much the participants were supposed to drop their knees during each execution criterion. This could have resulted in overlaps between the “less optimal” and “poor” variations. This can partially be seen in figure 4.9.

The results from this test are promising and indicate that it could be possible to create a model that can classify different predefined execution criteria of an exercise. However, the train, validation and test accuracies compared to when classifying only the exercises implies that classification of execution variations is more complex. To improve the accuracy further, more data would need to be collected. By increasing the size of the dataset, the possibilities to train an even more generalizable model would likely increase. Another way to achieve a better accuracy would be to increase the depth of the model so that it can recognize more complex features. However, without expanding the dataset, a deeper and more complex network would most likely be more prone to overfit.

5.3.5 Interpretability of Execution-Criterion Classification

While the results from the execution-criterion classification show promise, they also highlight an important challenge when it comes to the interpretability of deep

learning models in real-life applications. In contrast to movement classification, execution-criterion classification not only tries to classify each exercise, but also to evaluate what variations are present in the movement, which increases the risks of introducing substantial classification ambiguities.

For a classification model for execution-variations to be interpretable, it is important to understand which part of the movement data influenced the model in its predictions. For example, if the model classifies a squat as “poor”, it is important to determine whether the decision was based on actual biomechanical features related to the predefined execution criteria of the performed squat, or whether the model has instead learned from unrelated patterns or confounding variables within the dataset.

One limitation of the current dataset is that the variations of exercises are categorized into three distinct classes, “correct”, “less optimal”, and “poor”. While this approach increased the chances for reproducibility of the deviations from the reference execution performed in this project, it most likely oversimplifies the complexity of human movement variations, since it is more of a spectrum rather than three distinct categories. This increases the risk of a model learning participant-specific movement characteristics instead of generalizing to biomechanical features related to the exercise execution. One possible future step to address individual variability could be to use transfer learning as a way of personalizing a model to one user. This approach would allow for a model to be partially adapted to the patterns of one participant’s movement variations.

5.4 Feature extraction

During this study an attempt to extract four different sensor-derived features from the data was made. The features that have been estimated and evaluated are the amount of repetitions made for the different exercises, the range of motion in the hip and knee joints, the knee drop during squats and sitting leg raises, and the smoothness of more physically demanding exercises. The end goal for the feature extraction is to evaluate whether or not the system is capable of recording data with high enough complexity that would allow for estimation of biomechanical characteristic such as movement amplitude and deviations from a reference execution.

5.4.1 Repetition Counter

The results in table 4.6 present that the repetition counter achieved an accuracy of 100% for both the pre-labelled data and the segmented data from the pseudo-live tests, which indicates that not only the repetition counter works for this dataset, but also that the data is of high enough quality to assess the repetition count of the specific exercises evaluated in this study. It is however important to note that the repetition counter only works as long as the pseudo-live segmentation does. If the classification model does not recognise parts of or the whole segment of an exercise, the repetition counter will fail to register all reps. However, the accuracy of the

repetition counter on the segmented data indicates that the segmentation works as intended in this case and does not miss any significant parts of the exercises.

Another aspect to keep in mind is that all exercises were performed in sync with a metronome which can cause the rep counter to succeed better than it would during real scenarios. In real-life settings the repetition time would most likely differ between users. The repetition counter developed during this thesis has been optimized for the data collected during this project, and therefore it will likely struggle if deployed in real scenarios.

5.4.2 Range of Motion

Many movements can be performed with different movement amplitudes. Therefore, range of motion detection could be important for future usage areas such as non-medical training-feedback but also gaming where different movement amplitudes could influence the game in different ways.

The calculated range of motion during the heel slides and the standing leg raises can be seen in figure 4.6a and 4.6b respectively. As can be seen in these figures they follow a clear trend as the mean range of motion decreases between the “correct” execution criterion and the “poor” execution criterion. This indicates that it could be possible to evaluate the range of motion using the data collected from the orthosis. However, deviations from the expected values can be seen in the aforementioned box plots. These deviations are most likely results of the exercise performance rather than errors in the data collected from the sensors.

5.4.2.1 ROM for Heel Slides

The target ROM during the heel slides recordings were 90°, 70° and 50° for the “correct”, “less optimal” and “poor” execution criteria respectively. However, in figure 4.6a and table B.3 it can be observed that there were some deviations from the expected values. It is also possible to see that the interpersonal variations were relatively big based on the standard deviation and the difference between the P25 and P75 percentiles. Since the angle encoder is an absolute capacitive encoder, these variations were most likely caused by the executions rather than incorrect readings from the sensor.

The two most probable causes for the deviation around the median ROM is the tape markings used to indicate the inner position of the heel slide, and the sitting location on the chair. The tape markings were used to simulate different ROMs while still maintaining a natural movement. However, since the participants were healthy and not physically limited by a mechanical lock, it is to be expected that some participants would slide their foot too far back or not far enough compared to the predefined execution criteria. During the exercise execution the participants were also instructed to sit at the front edge of the chair. However, there is no guarantee that they sat down at the exact same spot during every recording which would naturally result in deviations from the target angle even if they followed the tape markings.

When looking at the mean of the maximum recorded angle in figure 4.7a it is possible to observe that the deviations in this figure are smaller. This suggests that the deviation of the maximum value is only part of the reason for the bigger range in figure 4.6a. The other part is if the participants were not fully extending their leg when returning to the starting position. In figure 4.7c it is clear that not every participant extended their leg entirely when returning to the neutral position and that $\sim 50\%$ of the participants were still bending their knee more than 5° during the “correct” and “less optimal” execution criteria. It was only for the “poor” execution criterion that the median was centred at $\sim 0^\circ$. Since they were instructed to perform the exercises in sync with a metronome it would be reasonable to assume that some participants had trouble extending their knee properly since their focus was on the metronome and remembering to bend their knee to the correct marking.

For both the maximum flexion of the knee and for the mean range of motion during the heel slides it was also apparent that the participants more often than not bent their knee 60° instead of 50° during the recording for the “poor” execution criterion. This is to be expected since 50° is a relatively small and unnatural knee flexion which would result in the limit being harder to reach consistently without overshooting.

5.4.2.2 ROM for Standing Leg Raises

Figure 4.6b shows that the spread of the estimated hip angle is narrower than the angle of the knee joint during heel slides (figure 4.6a). This is likely related to the study design, where the participants managed to return to their starting position more consistently during standing leg raises. This can also be observed in figure 4.7d where the minimum angle remains relatively consistent between repetitions and participants. This is to be expected since the participants had a clear marking on the floor which indicated where they were supposed to stand during the different execution criteria.

Although the participants were instructed to perform the exercise with predefined target angles, the estimated median ROM deviates with approximately 5° from the expected angles. This is either caused by the derivation of the hip angle or related to natural variations in how the exercise was executed. The different exercise execution criteria were designed under the assumption that the hip joint would remain vertically aligned with the marking on the floor throughout the movement and that the participants would stop the movement when reaching the physical barrier. In practice, small variations in posture and movement strategy likely affected the estimated hip angle.

The largest interpersonal differences in hip angle can be observed for the standing leg raises when performed according to the “correct” execution criterion. One possible explanation for this is that the 45° angle was more challenging for some participants to reach, causing them to stop the movement before the desired angle was reached. This would explain the lower achieved angles of the hip joint in figure 4.6b. In contrast, some participants compensated by leaning their trunk sideways during the movement to come in contact with the physical barrier. Since the markings

on the floor were derived from the assumption of a fixed axis of rotation, these compensatory movements could shift the position of the hip joint, leading to larger estimated angles. This would explain the higher achieved angles of the hip joint in figure 4.6b.

The larger upper whisker for the standing leg raises when performed according to the “poor” execution criterion instead indicates that the smaller target angle of 15° was more difficult to reproduce. This might have caused some participants to overshoot the intended angle. Furthermore, the physical barrier used during the recording was intentionally soft and movable to avoid abrupt impacts that could potentially influence the calculation of jerk. As a result, some participants likely moved slightly beyond the intended stopping position.

Another factor influencing the ROM is the method used for deriving the hip angle from gyroscopic measurements. The results indicate that it might be possible to derive the hip angle from the gyroscopic data based on the observed differences between the different execution criteria. However, as mentioned earlier there is no guarantee that the IMU’s z-axis will align perfectly with the hip’s axis of rotation. Based on the results in figure 4.7d it would not appear as if this sensor drift has caused any major errors. The -20° angle recorded for one participant during the good executions is the only measurement which is likely influenced by accumulated integral errors and gyroscopic drift but all other values could be caused by errors in the execution.

5.4.3 Knee drop

The evaluation of knee drop allowed for further assessment of the possibility of utilizing the orthosis for estimating knee-joint deviations from a reference execution. These deviations were estimated through a project specific proxy, both for knee flexion and medial knee displacement.

5.4.3.1 Knee Drop for Sitting Leg Raises

In figure 4.8 the recorded knee drop for the sitting leg raises can be seen for each participant and each repetition. This is the only sensor-derived feature that does have a ground truth since the orthosis was locked according to physical limits. The lock was not utilized for the “correct” execution criterion but based on the results from the “less optimal” and “poor” execution criteria it is likely that the encoder is capable of registering the angle drop. There are some slight deviations around the expected angle during “less optimal” and “poor” variations, which is likely caused by the participants not allowing their knee to fully drop during the exercise. Furthermore, these deviations could also have originated from variations in the calibration of the zero location of the encoder. During the calibration the orthosis was locked at 0° but some slight movement is still possible even if the orthosis is locked. This would result in the zero position deviating a few degrees between recordings. There are also some outliers for the “poor” execution criterion which are a result of human error and not any calculation or sensor issues. The orthosis was accidentally locked

at 20° instead of 30°.

There is also some interesting information to gather from the “correct” execution criterion where the inter-participant spread is larger. During this recording the participants were instructed to keep their leg as straight as possible, but if they got tired and started dropping their knee the lock would hinder them from bending their knee more than 10°. This was introduced to assure that no executions in the “correct” execution criterion recording were carried out in accordance with the “less optimal” execution criterion and thereby increasing the chances of reproducibility. In figure 4.8a it is possible to see that the mean knee drop for each participant differs greatly based on the difference between the P0-P50 range and the P50-P100 range. Based on the median value it is also clear that more than 50% of participants rested on the lock during most of the recording. This is also visible while looking at figure 4.8b that contains the knee drop for every recorded rep. Based on this it is possible to speculate that for healthy people it is quite difficult to maintain a straight leg for 20 reps of sitting leg raises and based on the outliers in figure 4.8b it is quite probable that many people started with a straight leg but started naturally dropping their knee as they got tired.

5.4.3.2 Knee Drop for Squats

The knee drop during squats was evaluated by calculating the rotation around the first IMU’s y-axis since this was assumed to be perpendicular to the ground. The results in figure 4.9 indicate that it is possible to capture the existence of the knee drop this way, since the calculated knee drop increases on average between the “correct” execution criterion and “poor” execution criterion for the performed repetitions. It is however important to note, that no ground truth is available, and therefore it is impossible to determine whether or not the measurements are correct. Additionally these calculations are only estimated as the rotation around the first IMU’s y-axis which is not the same as rotation around the world coordinate system but rather an estimate of this rotation. Therefore these values are not actual medial displacements but rather a project specific derived proxy.

As can be seen in figures 4.9a and 4.9b the estimated internal rotation increases as the participants purposely drops their knees further inwards. At the same time, these results also suggest that there was no knee drop during some repetitions indicated by the whiskers at 0°. This can be seen for all execution criteria which theoretically should not be the case. All repetitions for the “less optimal” and “poor” variations should contain some knee drop since this was deliberately introduced for these execution criteria. There are multiple possible reasons for this both in regards to the execution itself, but also the calculation.

Since the knee drop is calculated through an integral of angular velocity, the calculations lack an absolute reference orientation and therefore treats the initial position of the knee as the zero position. For this to be correct and result in accurate measurements, the initial position must coincide with the knee’s motion trajectory during the squat. This would ensure that any calculated internal or external rotation would

represent a deviation from the perfect execution. However, it is possible that some participants positioned their knees pointing more forward or inward at the beginning of each rep, resulting in an outward rotation of the knees during the initiation of the repetition. In such cases, the initial outward rotation would be interpreted as external rotation relative to the starting position. Consequently, when the participant performed the actual knee drop later during the repetition, the movement could instead be interpreted as a return to the initial orientation rather than as internal rotation. This is a possible explanation for why some repetitions of “less optimal” and “poor” squats appear to contain very small amounts of knee drop, despite the movement being deliberately introduced.

To further evaluate this, the total knee displacement was evaluated by analysing both internal and external rotation throughout the repetitions. The results shown in figure 4.9c and 4.9d indicate a clearer positive relation between total knee displacement and exercise variation compared to the isolated knee drop measurements. This suggests that the total knee displacement might be able to better capture this movement variation more consistently than solely relying on calculated internal rotation relative to the starting position of the squats.

5.4.4 Movement Smoothness and Jerk Analysis

In biomechanics, jerk is often used to quantify differences in movement smoothness throughout a movement. Jerk was therefore evaluated during this study to determine if the IMU sensors are sensitive enough to extract information about movement smoothness. The movement smoothness was evaluated by calculating the log dimensionless jerk for the two most physically demanding exercises in this project; sitting leg raises and squats.

5.4.4.1 Jerk During Sitting Leg Raises

The results from the main study regarding sitting leg raises revealed one of the most unexpected results of this project. For the “less optimal” and “poor” execution criteria of this exercise the movement generally became less smooth. However, the “correct” criterion displayed the opposite behaviour, namely that the estimated jerk decreased over time indicating an increase in movement smoothness.

One possible reason for this could be explained by looking at the study design. In order to standardize the execution criterion and to ensure proper knee extension throughout the whole movement, the orthosis was locked at 10° during the “correct” execution criterion of sitting leg raises. While this helped with the standardisation of the execution criterion, it may also have altered the movement of the participants as explained in section 3.10.2.

To test this hypothesis, an additional experiment was carried out as described in section 3.10.2, where two extra recordings of sitting leg raises performed according to the “correct” execution criterion were recorded without locking the orthosis at 10° . For these recordings, the expected trend became more apparent as the movements gradually became less smooth over repetitions as shown in figure 4.16. Even though

the additional recordings were limited and only performed by the authors ($n=2$), the results suggest that the locking of the orthosis might have caused the unexpected trend for the “correct” sitting leg raises during the main study. However, due to the limited number of participants and the risk of confirmation bias, this is something that has to be further evaluated in future studies.

The smoothness of a movement is not solely influenced by the characteristics of the movement itself, but also by external constraints and experimental conditions. A mechanically constrained movement might therefore appear smoother over repetitions, while a more natural movement may reveal the opposite trend, even when the overall movement task remains unchanged. This could explain the pattern observed when comparing the original test to the results from the experimental hypothesis-generating observation.

The results also indicate the need for carefully designed study protocols for evaluation of movement smoothness. For the sake of reproducibility, standardization is an important factor, but mechanical constraints might introduce unintentional altering of the movement patterns that are wished to be evaluated. For future studies it would be of interest to investigate more natural execution conditions and to combine the measurement of jerk with electromyography (EMG).

When further analysing the scatter plots of the mean difference of LDLJ from the main study (figure 4.10), it is also possible to note that the smoothness of the exercise seems to have increased during the last two reps. The most probable reason for this is that the participants were told when they had two more reps to go, which could have affected the way they executed the last repetitions. This phenomena can also be seen in the scatter plots for the squats (figure 4.11).

5.4.4.2 Jerk During Squats

Unlike sitting leg raises, the jerk measurements for squats showed that the movement smoothness generally decreased across repetitions, particularly during the “less optimal” and “poor” execution criteria.

A possible explanation for this could be that squats are a more complex movement. Compared to sitting leg raises, squats involve larger movement amplitudes and coordinated movements across different joints. As the repetitions progressed, greater movement variability may therefore have become apparent in the recorded signals, resulting in increased jerk values. Furthermore, unlike the sitting leg raises, the squats were also performed without a mechanical constraint of the orthosis. Since there was no mechanical stopping point to stabilize the movement, the risk of artificially affecting the amount of jerk was reduced and could therefore be one explanation to why the observed trends were more clearly visible for squats.

However, similar to sitting leg raises, it is still important to be cautious when interpreting jerk as anything other than movement smoothness. An increase of jerk may be caused by several factors. Some of which are variations in movement speed, balance adjustments, individual movement strategies, or other sources of movement

variability. Additionally, some participants may naturally have more dynamic movement characteristics than others.

Despite these limitations, the trends observed in the LDLJ evaluation for sitting leg raises and squats, indicates that the proposed sensor system might be capable of capturing meaningful variations in movement smoothness from the recorded data. However, further studies will need to be conducted together with a ground truth to further evaluate this statement.

5.5 System design

As have been discussed, the orthosis with its integrated sensors appear to be able to capture data of high enough quality to allow for accurate movement classification. It also appears to be of good enough quality to be used to estimate features from predefined movement characteristics. However, since the system is still a prototype there are some things that can still be improved upon.

5.5.1 Size of the system

The PCB designed by Shen allowed the IMU casing and the electronic box to be smaller than for the first prototype created by Zhou. However, the design can still be improved. The current PCB design has the microcontroller as a separate unit stacked on top of the rest of the circuitry. In a future iteration of the PCB it would be of interest to combine the microcontroller with the circuit board to minimize the height of the attached casings. At this moment the current design has made the boxes smaller mainly by minimizing their depth and length while the height has increased. This increases the risk of a user hitting the box when walking past a chair for example, which occurred a few times during the recordings for the main study. This only introduced noise to the recorded data which could be removed by a filter, however from the perspective of wanting the orthosis to last over time it would be of utmost importance to make the casings even smaller to minimize the damage it could sustain.

5.5.2 Integrated battery

As mentioned in section 2.1.1 the orthosis is outfitted with an integrated battery positioned below the circuitry. However, despite this the decision was made to remove the battery and use a power bank throughout all recordings for this project. The reason for this decision was both to minimize the risk for the participants and the risk of damaging the circuitry. The integrated battery is a lithium ion battery and since the orthosis is strapped to the participants leg throughout the recordings the decision was made to remove the battery because of the possible fire risk that could be disastrous for the participant. Since the orthosis is still a prototype the protective casing is 3D printed and consequently there are some situations that could result in the battery getting damaged. One example is if the participants falls and damages the outer casing it could also damage the battery. Another example

is if the cables connecting the battery to the circuitry are damaged since they are accessible without removing the protective casing. The probability of this happening is relatively low but the consequences were deemed too high to ignore.

Another reason for choosing to use the power bank instead of the integrated battery is because of the position of the on-off switch. Since the orthosis is still a prototype some upgrades are needed to make it more user friendly, one of these changes is the position of the on-off switch. This switch is currently positioned almost in the middle of one of the circuitry boards. To turn it on, one would have to use something long and thin to reach inside the electronic enclosure to flip the switch unless the plastic cover is removed. Based on the study design three recordings were to be recorded from 24 people meaning that the switch would have needed to be flipped 48 times if the orthosis was not turned off between recordings or 144 times if it was turned off. Based on this the meticulous task of turning on the switch at least 48 times while trying to avoid damaging the circuitry was deemed too many.

In later iterations of this orthosis it would be of interest to update the material of the casing containing the battery to something more durable that would be able to better protect the battery. Another functionality that would be beneficial is to add some way to easily access the battery so that it can be removed without taking apart the entire electronic enclosure that also contains the microcontroller and one of the IMUs to make it safer. It would also be of interest to update the enclosure and possibly the PCB to make the switch more easily accessible from the outside.

6

Conclusion

This project investigated the potentials of a sensor-equipped orthosis system for classification of controlled lower-limb movements and extraction of sensor-derived movement features. The results demonstrate that the data collected with the system is of high enough quality to classify four knee and hip exercises with high accuracy using deep learning models. Among the evaluated models, the CNN-LSTM model performed best in regards to the trade-off between classification accuracy and computational efficiency. The model achieved LOSO cross-validation accuracies over 99% for all movement classes across all predefined execution criteria.

The possibility of extracting sensor-derived movement features from the data, such as range of motion, movement deviations, and motion smoothness was evaluated. The trends that are visible in the results indicate that there is a possibility of using the system for this task, but due to the lack of a ground truth further studies need to be conducted in order to draw quantitative conclusions about how accurately these features can be extracted.

An important limitation of this study is that the dataset consisted solely of data collected from relatively young and healthy participants, meaning that the results should not be generalized beyond the tested participant group and controlled movement protocol. Therefore, a natural next step for future work would be to expand the dataset to include a more diverse cohort.

6.1 Future Work

For the developed orthosis system to be recognised as a prototype platform for controlled movement analysis research, several aspects need to be addressed in future studies.

6.1.1 System and Hardware Development

An important next step is to continue developing the usability and robustness of the orthosis system. Future studies should be carried out with the goal of making the circuitry and electronic enclosures smaller, while also improving accessibility and protection of components such as the battery and on-off switch.

Additionally, the system should be outfitted with nine-axis IMUs, including a magnetometer, instead of six-axis IMUs. This could improve the estimation of a global coordinate system for the assessment of knee alignment during exercises involving movement in multiple planes, as well as estimations related to the hip range of motion.

6.1.2 Further Data Collection and Method Development

Future work should also focus on gathering additional data to further evaluate the possibilities and limitations of the classification models and feature extraction pipeline. This includes collecting data from more individuals that perform variations of additional controlled lower-limb movements relevant to movement analysis. By expanding the dataset in this way, it would contribute to a broader and more generalizable database for both classification and extraction of sensor-derived movement features.

The feature extraction pipeline should also be developed to further investigate the possibilities for extraction of more information regarding exercise variations. The current study only evaluated four exercises where one movement variation was simulated for each exercise. Since training executions can vary in a multitude of ways, and since there are more features which could be of interest for biomechanical evaluation, this should be investigated in future studies. Considering the broader scope of AURORA, this should ideally be done in collaboration with movement-analysis experts to ensure that extracted features are relevant for movement analysis.

It would also be of interest to include ground truth measurements when further developing the feature extraction pipeline. Motion capture systems could be used to accurately evaluate the extracted biomechanical features, especially the range of motion in the knee and hip joints, and the medial knee displacement during squats. For the evaluation of movement smoothness and jerk, electromyography (EMG) could potentially be used as a ground truth.

Bibliography

- [1] R. Madrigal-Cerezo, N. Domínguez-Sanz, and A. Martín-Rodríguez, “Wearable biosensing and machine learning for data-driven training and coaching support,” *Biosensors*, vol. 16, p. 97, Feb 2026.
- [2] D. Navakauskas and M. Dumpis, “Wearable sensor-based human activity recognition: Performance and interpretability of dynamic neural networks,” *Sensors*, vol. 25, p. 4420, Jul 2025.
- [3] G. Sharma, T. Yadav, V. Yadav, and S. Mahajan, “Hybrid cnn-lstm model for accurate human activity recognition using wearable sensor data,” *International Journal for Multidisciplinary Research*, vol. 7, Dec. 2025. Accessed: May 15, 2026.
- [4] F. Nazari, D. Nahavandi, N. Mohajer, and A. Khosravi, “Comparison of deep learning techniques on human activity recognition using ankle inertial signals,” *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, p. 2251–2256, Oct 2022.
- [5] Y. Zhou, “Design and development of a novel sensorized orthosis for orthopedic rehabilitation,” Master’s thesis, Chalmers University of Technology, 2025. Master’s thesis, Systems, Control and Mechatronics (MPSYS), MSc, Available: <https://odr.chalmers.se/items/71182742-82e3-43d0-aef3-52630db29a5f>, Accessed: 2026-02-17.
- [6] P. Shen, “Design and development of an adaptive robotic rehabilitation exoskeleton for lower limb,” Master’s thesis, Chalmers University of Technology, 2026. Master’s thesis at the department of electrical engineering, Available: <https://odr.chalmers.se/items/8e9d1702-4b65-4c79-8100-15a8323d2349>, Accessed: 2026-04-02.
- [7] E. Fredlund, N. Timpe, and W. Peng, “Designing exoskeleton for physiotherapy and rehabilitation,” project report, Chalmers University of Technology, Gothenburg, Sweden, 2025. Course project report in SSY226.
- [8] L. Barberán, B. Houshmand, and L. Persson, “Designing next-gen adaptive exoskeleton for physiotherapy and rehabilitation,” project report, Chalmers University of Technology, Gothenburg, Sweden, 2025. Course project report

in SSY226.

- [9] S. Tavakoli, “Intelligent systems for rehabilitation and physiotherapy,” Nov 2023. Available: <https://www.chalmers.se/en/current/news/e2-intelligent-systems-for-rehabilitation-and-physiotherapy/>, Accessed: 2026-05-14.
- [10] X. Li, J. Liu, Y. Huang, D. Wang, and Y. Miao, “Human motion pattern recognition and feature extraction: An approach using multi-information fusion,” *Micromachines*, vol. 13, p. 1205, Jul 2022.
- [11] Y. Benachour, M. Rehman, and F. Flitti, “Towards improved human arm movement analysis: Advanced feature engineering and model optimization,” *Engineering Applications of Artificial Intelligence*, vol. 156, p. 111194, Sep 2025.
- [12] S. M. Sid’El Moctar, I. Rida, and S. Boudaoud, “Comprehensive review of feature extraction techniques for semg signal classification: From handcrafted features to deep learning approaches,” *IRBM*, vol. 45, p. 100866, Dec 2024.
- [13] M. A. Yadav and D. S. Patil, ““optimizing semg gesture recognition with stacked autoencoder neural network for bionic hand”,” *MethodsX*, vol. 14, p. 103207, Feb 2025.
- [14] K. Xia, J. Huang, and H. Wang, “Lstm-cnn architecture for human activity recognition,” *IEEE Access*, vol. 8, p. 56855–56866, Mar 2020.
- [15] X. Shi, J. Hou, J. Wang, H. Lu, S. Li, X. Meng, and K. Li, “Research on movement intention recognition based on cnn-lstm,” *Electronics*, vol. 15, p. 797, Feb 2026.
- [16] M. Rahman, “Different ways to combine cnn and lstm networks for time series classification tasks | by mijanur rahman | medium,” Dec 2022.
- [17] D. Gholamiangonabadi, N. Kiselov, and K. Grolinger, “Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection,” *IEEE Access*, vol. 8, p. 133982–133994, Jul 2020.
- [18] L. Xiang, Z. Gao, P. Yu, J. Fernandez, Y. Gu, R. Wang, and E. M. Gutierrez-Farewik, “Explainable artificial intelligence for gait analysis: Advances, pitfalls, and challenges - a systematic review,” *Frontiers in Bioengineering and Biotechnology*, vol. 13, Oct 2025.
- [19] P. Brasiliano, G. Pavei, and E. Bergamini, “Smooth moves: Comparing log dimensionless jerk metrics from body center of mass trajectory and wearable sensor acceleration during walking,” *Sensors*, vol. 25, p. 1233, Feb 2025.
- [20] D. Bandara and J. Arata, “Active range of motion measurement system using an optical sensor to evaluate hand functions,” *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*,

p. 1–4, Jul 2023.

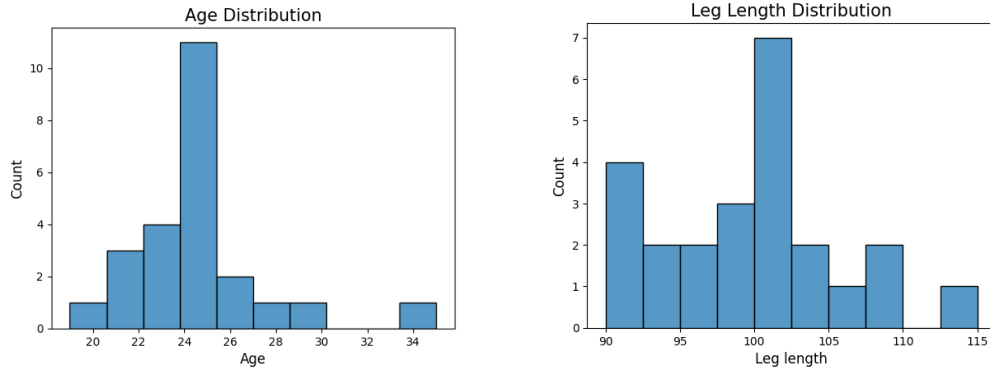
- [21] A. Roren, A. Mazarguil, D. Vaquero-Ramos, J.-B. Deloose, P.-P. Vidal, C. Nguyen, F. Rannou, D. Wang, L. Oudre, and M.-M. Lefèvre-Colau, “Assessing smoothness of arm movements with jerk: A comparison of laterality, contraction mode and plane of elevation. a pilot study,” *Frontiers in Bioengineering and Biotechnology*, vol. 9, Jan 2022.
- [22] S. Kuliś, P. Pietraszewski, and B. Callegari, “Characteristics of post-exercise lower limb muscle tremor among speed skaters,” *Sensors*, vol. 25, p. 4301, Jul 2025.
- [23] P. Palani, S. Panigrahi, G. Renganathan, Y. Kurita, and A. Thondiyath, “Empirical classification of fatigue-induced physiological tremor in robot-assisted manipulation tasks using bilstm-gru network,” *Frontiers in Rehabilitation Sciences*, vol. 6, Jun 2025.
- [24] Enovis, *DonJoy® X-ROM™ Post-Op Knee Brace*, 2020. Revision A.
- [25] TDK InvenSense, *ICM-42688-P: High-Precision 6-Axis MEMS MotionTracking Device Product Specification*, 7 2023. Revision 1.8.
- [26] same sky, *SERIES: AMT22 / DESCRIPTION: MODULAR ABSOLUTE ENCODER*, 12 2024. Revision 1.09.
- [27] espressif, *ESP32-S3 Series Datasheet Version 2.2*, 2026.
- [28] B. Fan, L. Zhang, S. Cai, M. Du, T. Liu, Q. Li, and P. Shull, “Influence of sampling rate on wearable imu orientation estimation accuracy for human movement analysis,” *Sensors*, vol. 25, p. 1976, Mar 2025.

A

Appendix 1

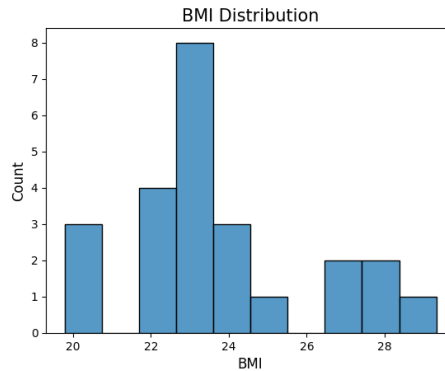
In this appendix, the supplementary methodological insights are provided.

In figure A.1 below, the age, leg length, and BMI distribution of the study population is presented in histograms.



(a) Age distribution of the study population.

(b) Distribution of leg lengths within the study population.



(c) Distribution of BMI within of the study population.

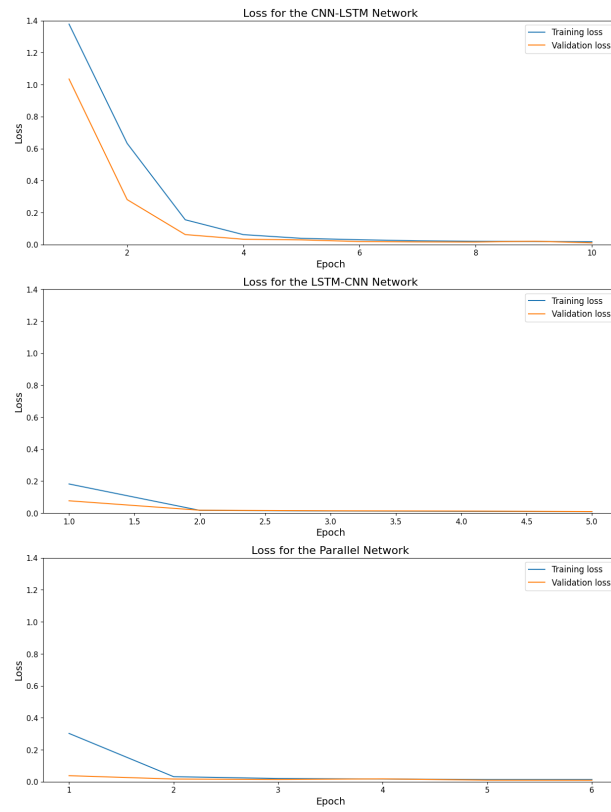
Figure A.1: Distribution of age, leg length and BMI within the study population. The y-axis in the histogram shows the number of participants in each of the 10 boxes.

In table A.1 below, the hyperparameters and their range tested during the heuristic derivation of the model architectures can be seen.

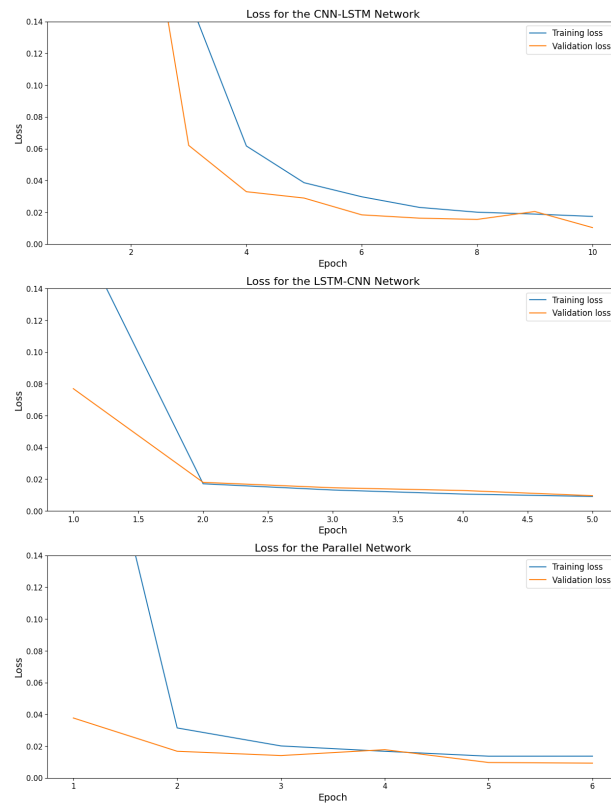
	Interval
Units (CNN)	16-256
Hidden Layers (CNN)	3-6
Kernel size (CNN)	3x3-10x10
Batch norms	1 layer-all layers
Units (LSTM)	32-128
Hidden Layers (LSTM)	1-3
Units (Fully connected layer)	64-256
Drop out	0.2-0.8
Learning rate	0.00001-0.01

Table A.1: All test-ranges for the hyperparameters during the development of the finalized models.

In figure A.2 below, the training- and validation loss curves for the three model architectures are presented. Figure A.2a shows the full scale loss curves, while figure A.2b shows a zoomed in graph of each loss curve.



(a) Full scale loss curves.



(b) Zoomed in loss curves to better visualize the loss for the last epochs.

Figure A.2: The loss curves for the CNN-LSTM, LSTM-CNN and parallel CNN and LSTM networks.

B

Appendix 2

In this appendix, the supplementary results are provided.

Figure B.1 below shows the results from the supplementary frequency analysis carried out for the desired sampling frequencies of 150 Hz, 200 Hz, 300 Hz, 400 Hz, 750 Hz, and 1000 Hz when utilizing UDP.

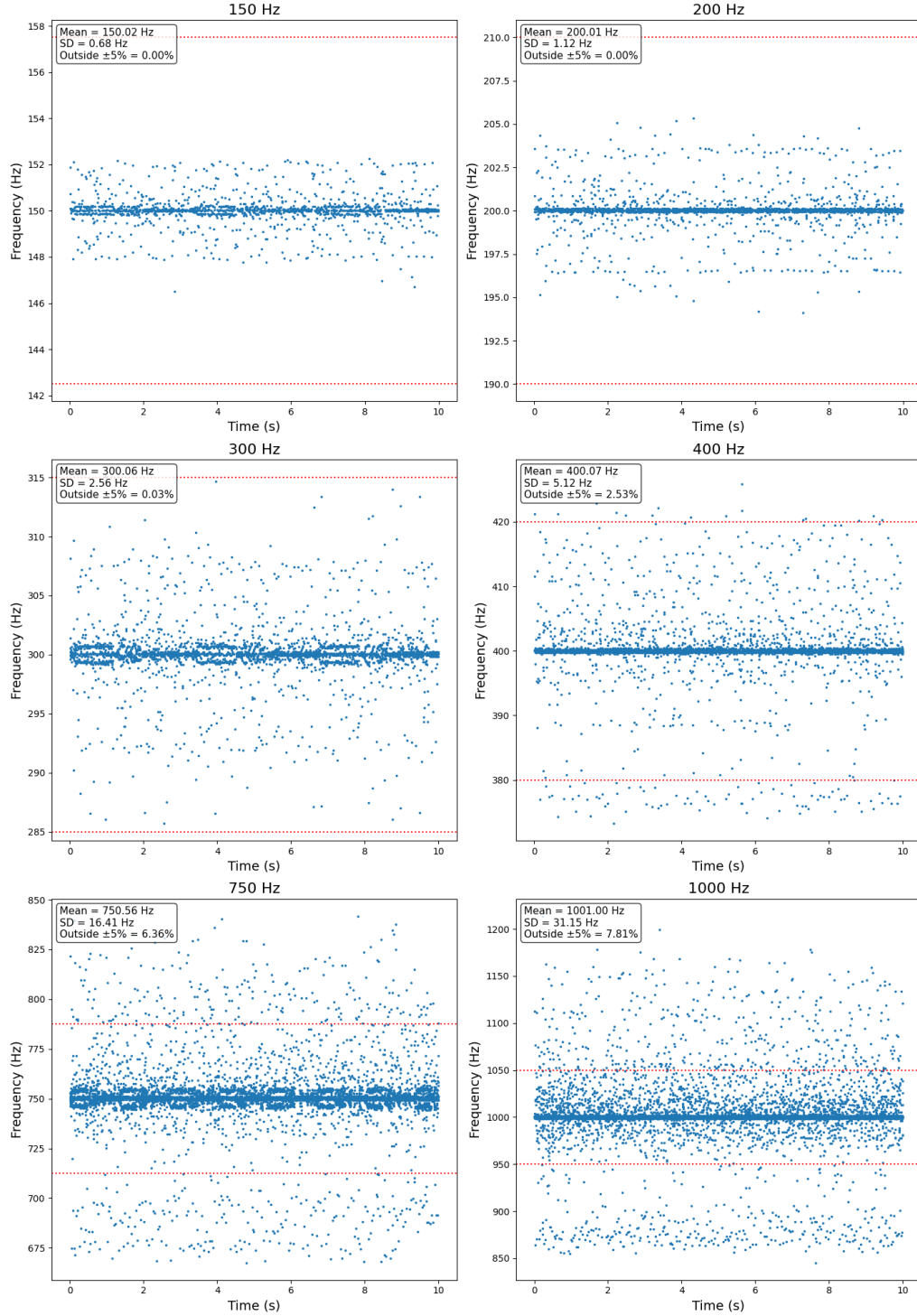


Figure B.1: Sampling frequencies of the system when writing the data directly to a computer utilizing UDP for a desired frequency of 150 Hz, 200 Hz, 300 Hz, 400 Hz, 750 Hz, and 1000 Hz. The box in the north-west corner of each plot presents the mean sampling frequency, standard deviation, and proportion of samples outside of the $\pm 5\%$ marks (red dotted line).

Figure B.2 below shows the results from the supplementary frequency analysis carried out for the desired sampling frequencies of 150 Hz, 200 Hz, 300 Hz, 400 Hz, 750 Hz, and 1000 Hz when writing the data over to the locally connected microSD card.

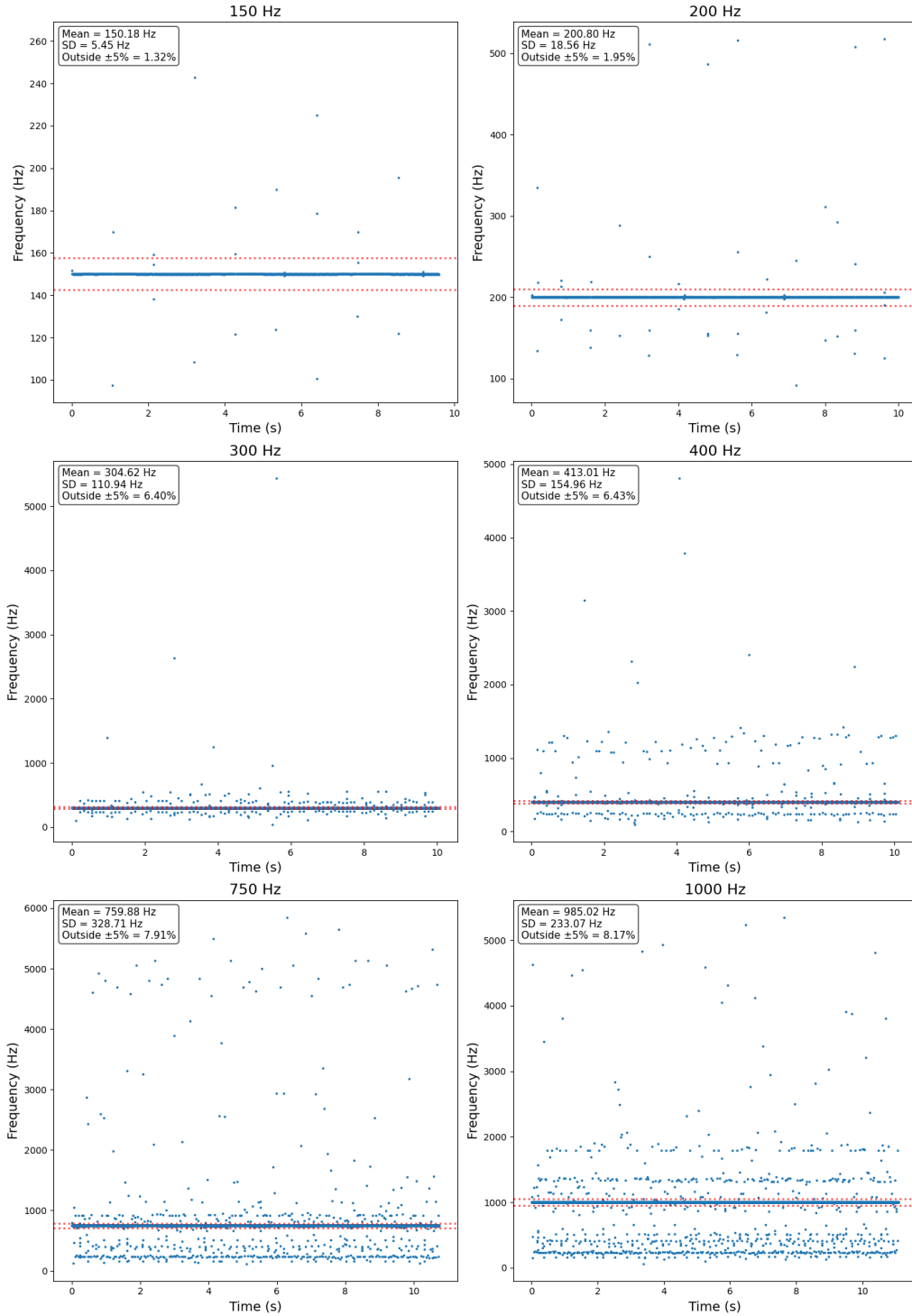


Figure B.2: Sampling frequencies of the system when writing the data over to a locally connected microSD card for a desired frequency of 150 Hz, 200 Hz, 300 Hz, 400 Hz, 750 Hz, and 1000 Hz. The box in the north-west corner of each plot presents the mean sampling frequency, standard deviation, and proportion of samples outside of the $\pm 5\%$ marks (red dotted line).

In table B.1 and figure B.3 additional results are presented for the offline performance of the CNN-LSTM model. Table B.1 presents the accuracy and F1-score on the training, validation, and test datasets for each individual class for the CNN-LSTM network. Figure B.3 shows the confusion matrix of the CNN-LSTM network when evaluated on the test dataset. The confusion matrix has been normalized along the true label.

	Training data		Validation data		Test data	
	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score
Other	99.45	0.9950	99.68	0.9957	99.39	0.9939
Heel-slide	99.58	0.9956	99.37	0.9968	98.98	0.9911
Sitting Leg Raises	99.58	0.9946	99.68	0.9968	100.0	0.9962
Lateral Leg Raises	99.68	0.9963	100.0	0.9983	99.73	0.9986
Squat	99.74	0.9977	99.38	0.9953	99.47	0.9960

Table B.1: Accuracy and F1-score for the CNN-LSTM network for each individual class.

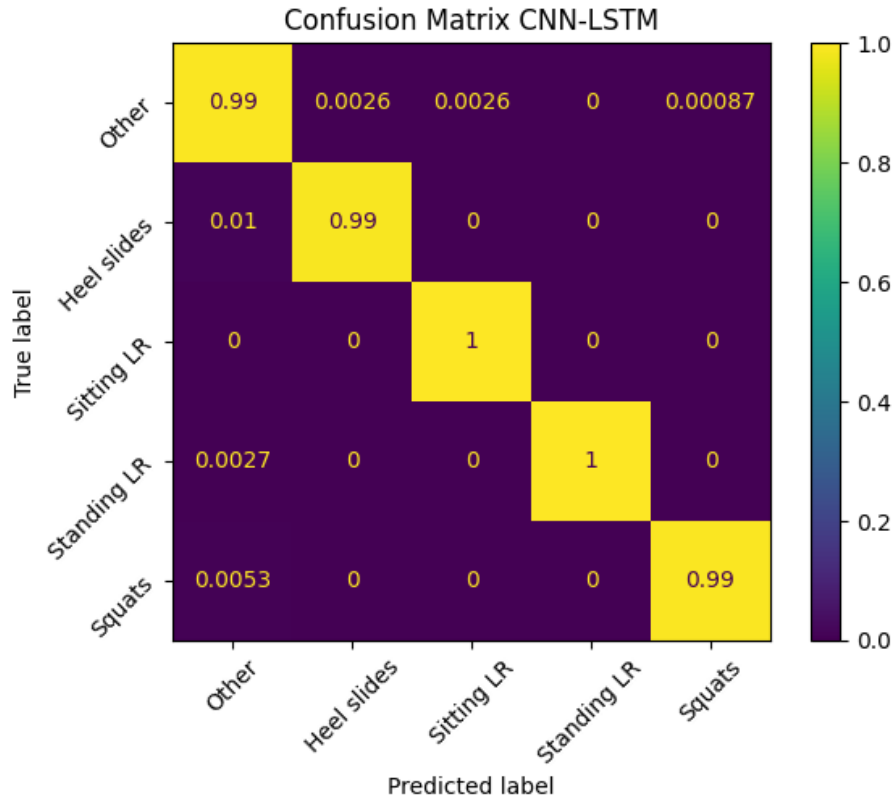


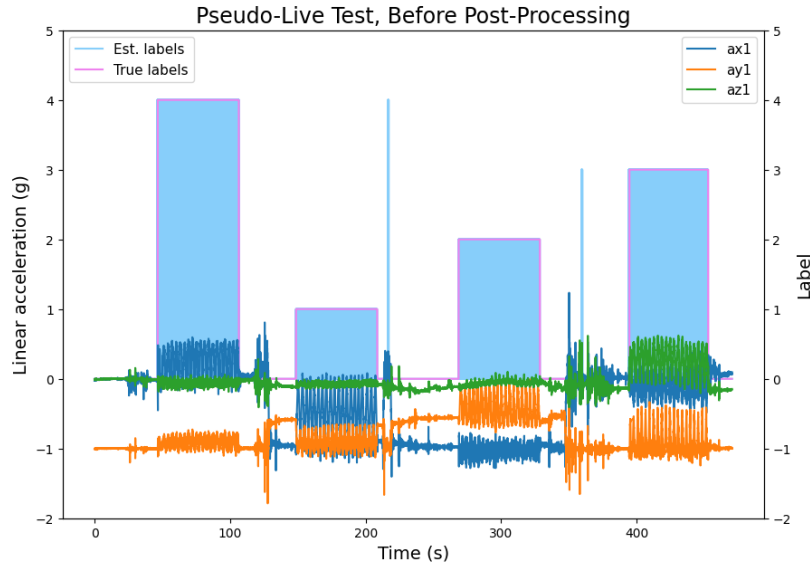
Figure B.3: Confusion matrix of the CNN-LSTM network when evaluated on the test dataset. Normalized along the true label.

Table B.2 contains additional information describing the box plots for the LOSO cross validation accuracy (figure 4.3).

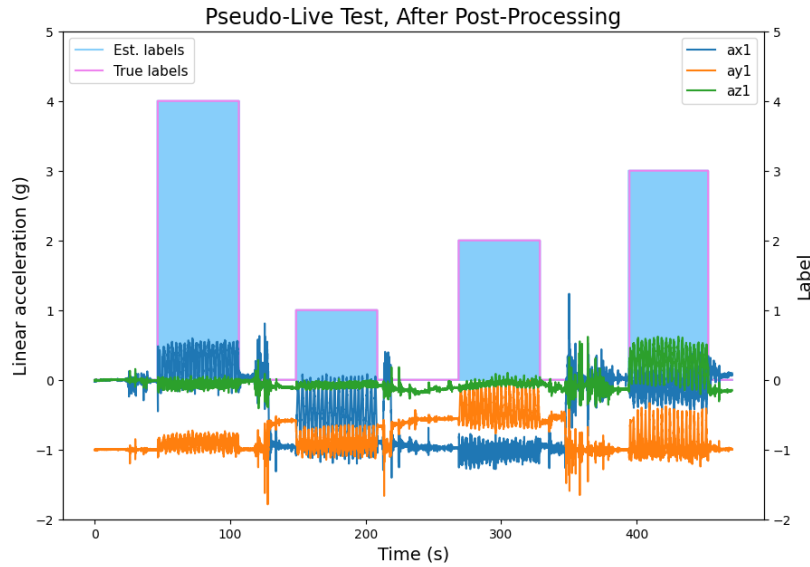
	Mean $\pm$ SD (%)	P25 (%)	Median (%)	P75 (%)
Training data	99.59 $\pm$ 0.08	99.56	99.61	99.63
Test data	99.42 $\pm$ 0.35	99.26	99.49	99.64

Table B.2: The mean $\pm$ SD and percentiles of the LOSO accuracy for the training and test datasets.

In figure B.4 an example of the pseudo-live classification, before and after post processing, is presented.



(a) Example of the segmentation before post-processing.



(b) Example of the segmentation after post-processing.

Figure B.4: Example of the segmentation before and after post-processing. The participant has consented to the publication of the data segmented in this figure.

Table B.3 below contains additional information describing the box plots for the range of motion during heel slides and standing leg raises (figure 4.6).

Feature	Variation	Mean $\pm$ SD	P25	Median	P75
Mean Range Heel Slides ($^{\circ}$)	Correct	88.07 $\pm$ 8.53	84.21	87.44	93.77
	Less Optimal	74.94 $\pm$ 8.92	67.57	75.83	83.28
	Poor	59.29 $\pm$ 7.11	53.73	59.10	65.07
Mean Range Standing Leg Raises ($^{\circ}$)	Correct	50.23 $\pm$ 4.73	47.52	50.98	53.61
	Less Optimal	34.88 $\pm$ 2.78	33.29	34.81	35.74
	Poor	18.85 $\pm$ 2.73	16.96	17.73	20.39

Table B.3: The mean $\pm$ SD and percentiles of the range of motion depending on the execution criteria for heel slides and standing leg raises.

Table B.4 below contains additional information describing the box plots for the maximum and minimum movement amplitude during heel slides and standing leg raises (figure 4.7).

Feature	Variation	Mean $\pm$ SD	P25	Median	P75
Maximum Flexion Heel Slides ($^{\circ}$)	Correct	92.34 $\pm$ 5.15	90.34	92.56	93.97
	Less Optimal	77.41 $\pm$ 6.42	75.86	77.40	81.74
	Poor	59.00 $\pm$ 5.97	56.59	59.03	62.02
Maximum Abduction Standing Leg Raises ($^{\circ}$)	Correct	51.15 $\pm$ 7.30	48.91	52.58	55.01
	Less Optimal	35.65 $\pm$ 3.96	33.36	35.54	38.62
	Poor	19.19 $\pm$ 2.97	17.63	18.91	20.98
Minimum Flexion Heel Slides ($^{\circ}$)	Correct	4.27 $\pm$ 7.86	-1.73	5.69	7.46
	Less Optimal	2.47 $\pm$ 6.80	-2.84	3.71	5.99
	Poor	-0.29 $\pm$ 5.15	-3.61	0.05	3.00
Minimum Abduction Standing Leg Raises ($^{\circ}$)	Correct	0.91 $\pm$ 6.04	-1.09	1.07	3.91
	Less Optimal	0.77 $\pm$ 3.31	-1.77	0.28	2.12
	Poor	0.33 $\pm$ 2.22	-1.00	0.31	1.83

Table B.4: The mean $\pm$ SD and percentiles of the maximum and minimum joint rotation depending on the execution criteria for heel slides and standing leg raises.

Table B.5 below contains additional information describing the box plots of the maximum knee drop during sitting leg raises (figure 4.8).

Feature	Variation	Mean $\pm$ SD	P25	Median	P75
Max Knee Drop Sitting Leg Raises (per person) ($^{\circ}$)	Correct	7.91 ± 3.76	6.42	9.73	10.36
	Less Optimal	20.94 ± 1.01	20.05	20.90	21.73
	Poor	29.54 ± 3.46	29.81	30.25	31.50
Max Knee Drop Sitting Leg Raises (per repetition) ($^{\circ}$)	Correct	7.91 ± 3.98	5.94	9.80	10.43
	Less Optimal	20.94 ± 1.09	19.98	20.89	21.76
	Poor	29.54 ± 3.49	29.62	30.28	31.69

Table B.5: The mean $\pm$ SD and percentiles for the observed knee drop for sitting leg raises during all three execution criteria, for each participant and repetition.

Table B.6 below contains additional information describing the box plots of maximum inwards knee rotation and total knee rotation during squats (figure 4.9).

Feature	Variation	Mean $\pm$ SD	P25	Median	P75
Maximum Knee Drop Squats (per person) ($^{\circ}$)	Correct	-1.91 ± 1.08	-2.40	-1.70	-1.23
	Less Optimal	-6.33 ± 3.75	-8.76	-5.62	-2.89
	Poor	-9.00 ± 4.75	-10.94	-7.76	-6.09
Maximum Knee Drop Squats (per rep) ($^{\circ}$)	Correct	-1.91 ± 1.51	-2.81	-1.63	-0.74
	Less Optimal	-6.33 ± 4.53	-9.13	-5.66	-2.79
	Poor	-9.00 ± 5.68	-12.68	-8.01	-5.06
Mean Leg Rotation Squats (per person) ($^{\circ}$)	Correct	6.62 ± 2.56	4.85	6.25	7.49
	Less Optimal	16.39 ± 3.93	14.42	15.94	17.59
	Poor	20.20 ± 3.91	18.14	18.95	22.39
Mean Leg Rotation Squats (per rep) ($^{\circ}$)	Correct	6.62 ± 2.83	4.66	6.22	7.75
	Less Optimal	16.39 ± 4.53	13.47	16.01	18.50
	Poor	20.20 ± 4.58	16.89	19.84	23.42

Table B.6: The mean $\pm$ SD and percentiles for the observed knee drop and total leg rotation during squats for all three execution criteria, per person and per repetition.

Figure B.5 below shows the scatter plots of the mean difference of LDLJ computed for heel slides for the different execution criteria.

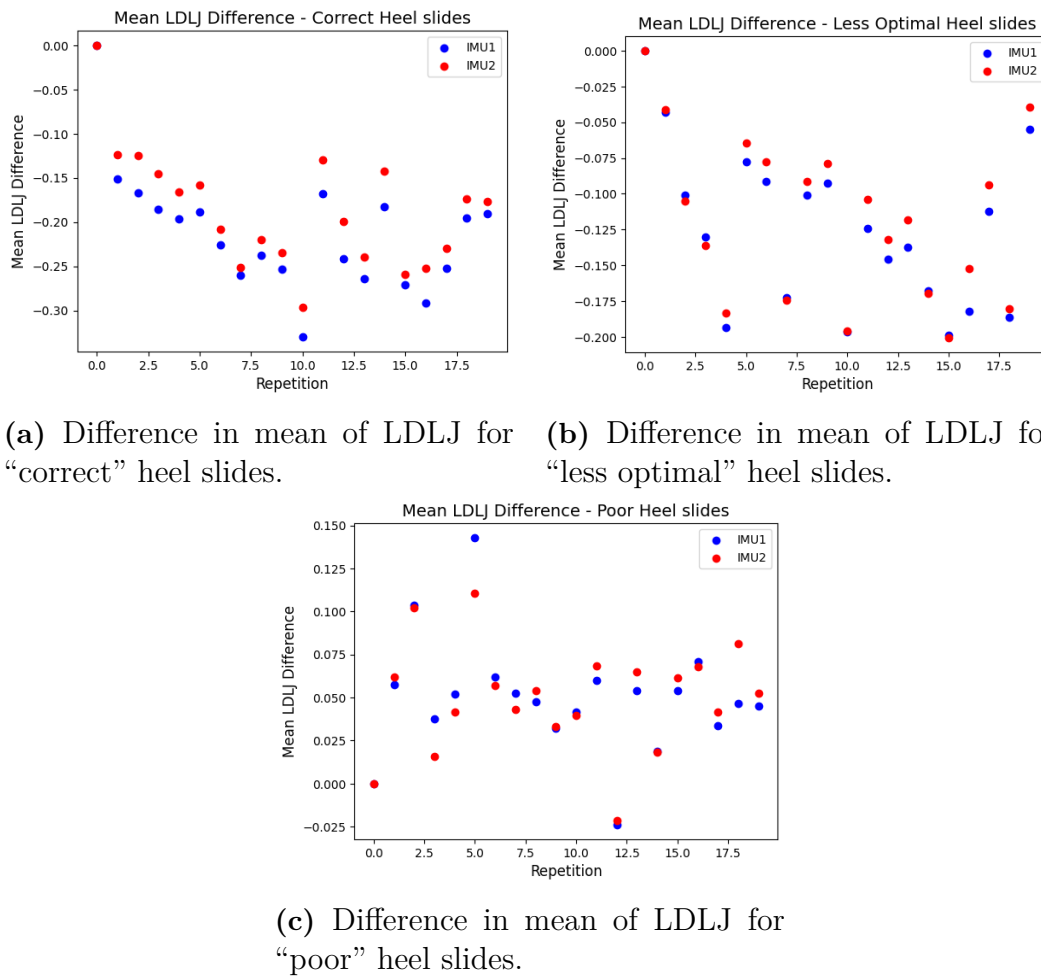


Figure B.5: Difference in mean of LDLJ for heel slides during the three execution criteria.

Figure B.6 below shows the scatter plots of the mean difference of LDLJ computed for standing leg raises for the different execution criteria.

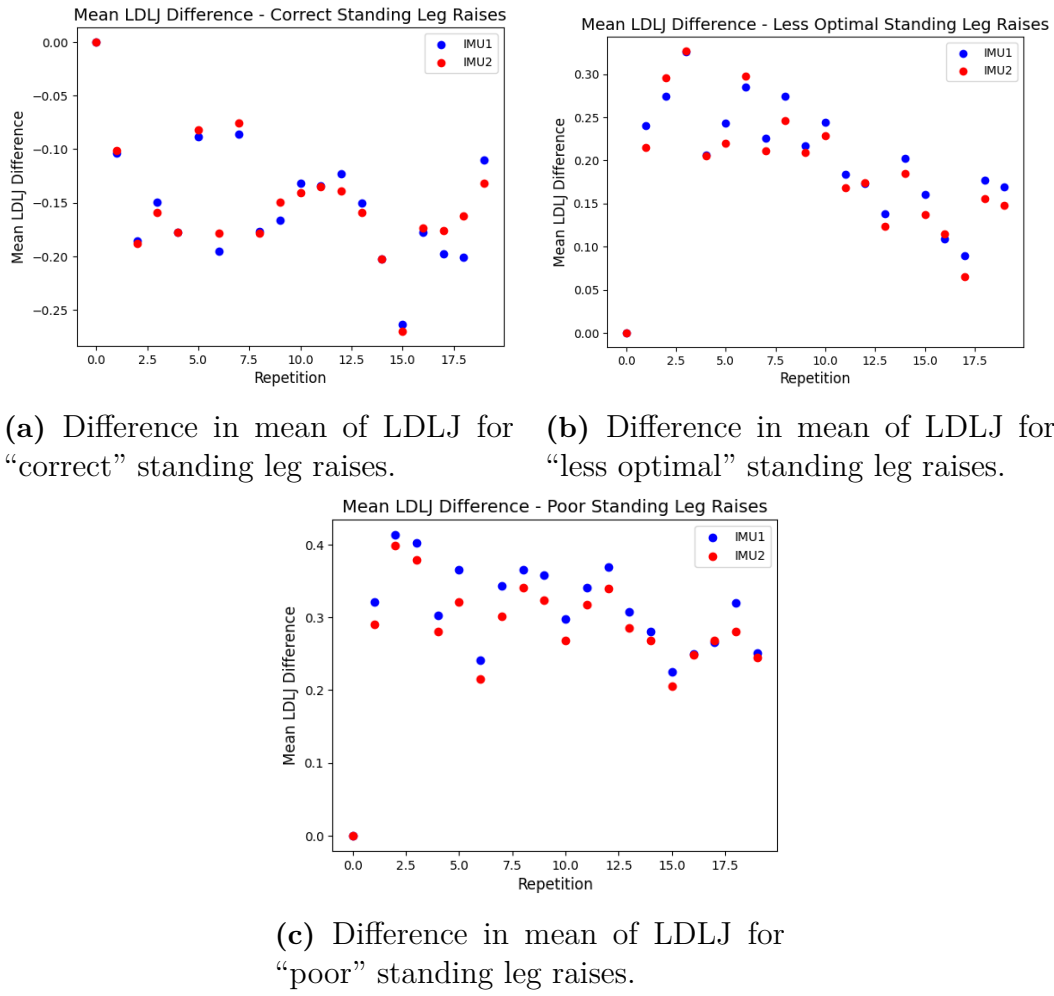
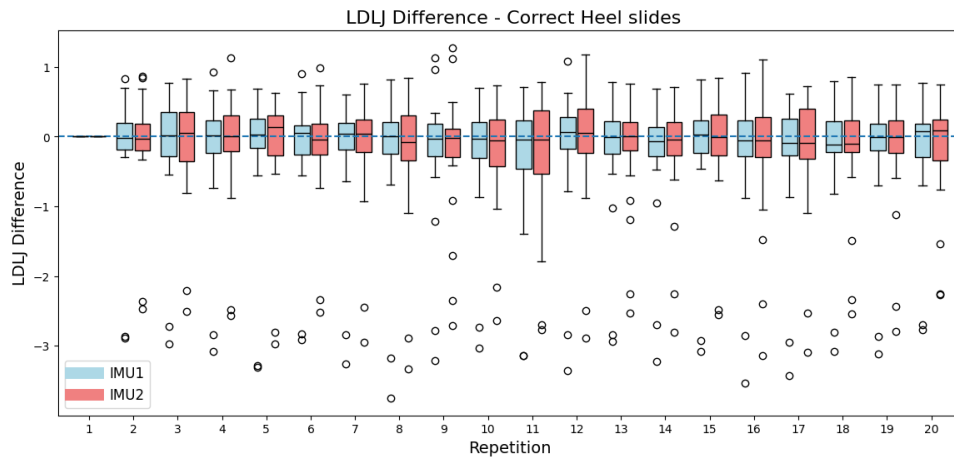
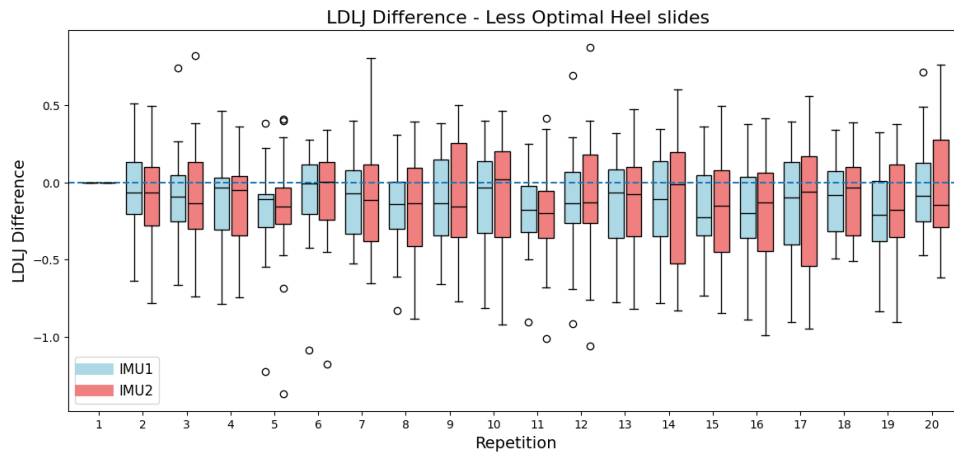


Figure B.6: Difference in mean of LDLJ for standing leg raises during the three execution criteria.

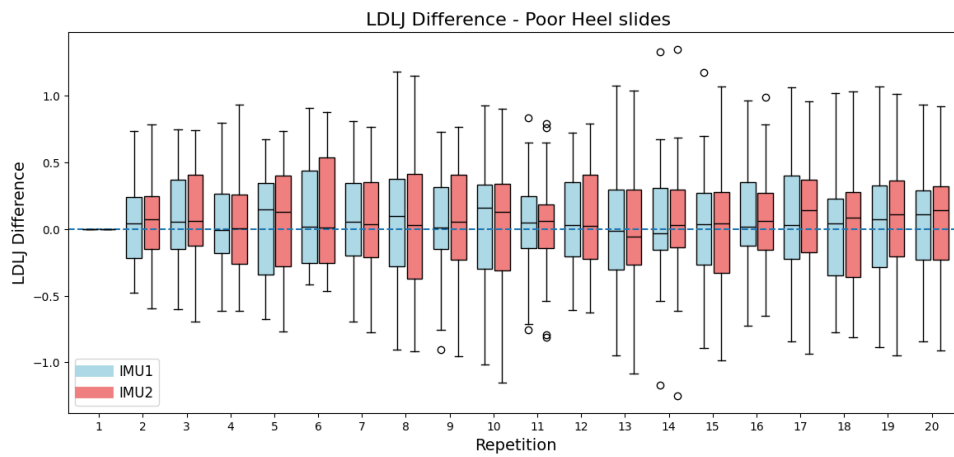
Figure B.7 below shows the box plots for the difference of LDLJ computed for heel slides for the different execution criteria.



(a) Difference of LDLJ for “correct” heel slides.



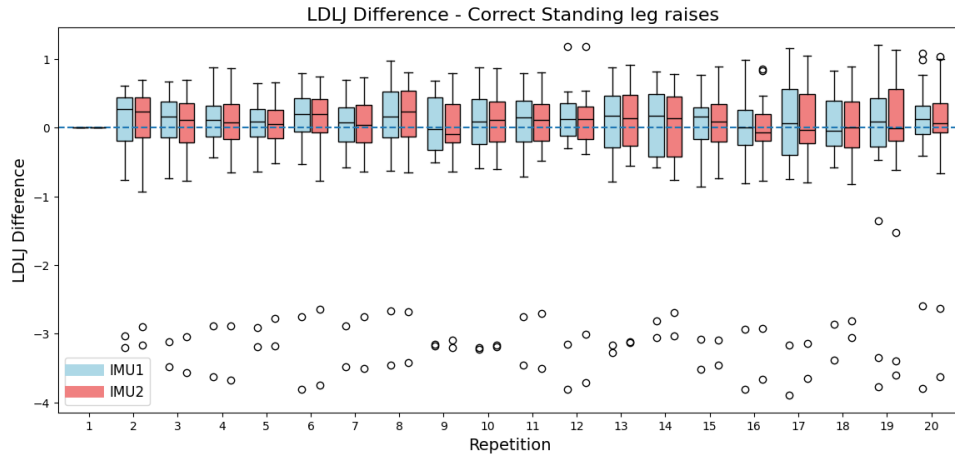
(b) Difference of LDLJ for “less optimal” heel slides.



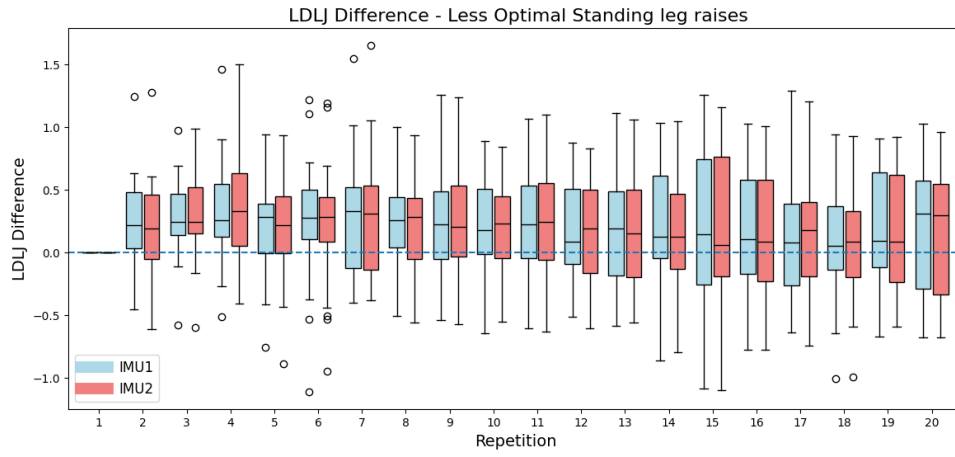
(c) Difference of LDLJ for “poor” heel slides.

Figure B.7: Difference of LDLJ for heel slides during the three execution criteria.

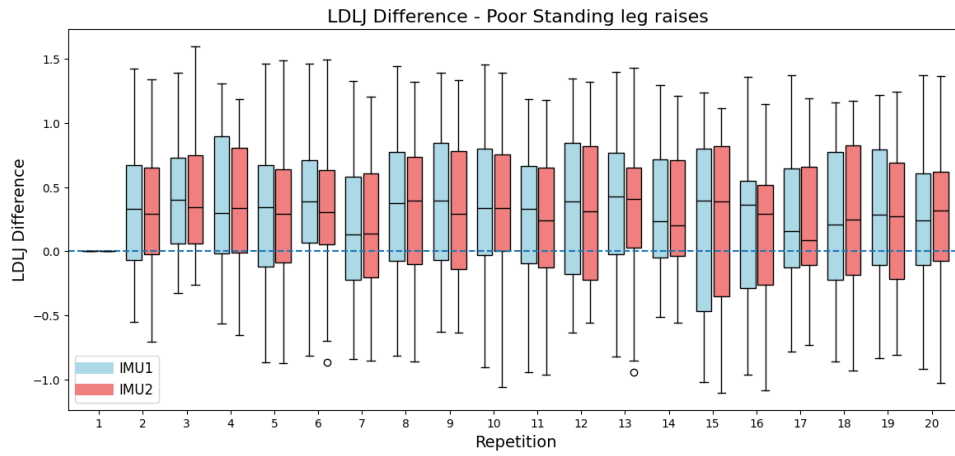
Figure B.8 below shows the box plots for the difference of LDLJ computed for standing leg raises for the different execution criteria.



(a) Difference of LDLJ for “correct” standing leg raises.



(b) Difference of LDLJ for “less optimal” standing leg raises.



(c) Difference of LDLJ for “poor” standing leg raises.

Figure B.8: Difference of LDLJ for standing leg raises during the three execution criteria.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY