

USING DEEP LEARNING FOR MOTION ANALYSIS OF 3D MOTION CAPTURE
DATA FOR FORECASTING MOTION AND FATIGUE

by

Geovanni Hernandez Jr., B.S.

A thesis submitted to the Graduate Council of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Engineering
May 2021

Committee Members:

Damian Valles, Chair

Francis A. Méndez

Jesus A. Jimenez

COPYRIGHT

by

Geovanni Hernandez Jr.

2021

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

This work is protected by the Copyright Laws of the United States (Public Law 94-553, section 107). Consistent with fair use as defined in the Copyright Laws, brief quotations from this material are allowed with proper acknowledgment. Use of this material for financial gain without the author's express written permission is not allowed.

Duplication Permission

As the copyright holder of this work I, Geovanni Hernandez Jr., authorize duplication of this work, in whole or in part, for educational or scholarly purposes only.

.

DEDICATION

My education career in this path of life is dedicated to many individuals, that I would have never made it to this point without them. I hope that this work will provide some form of inspiration and a sense of determination.

To my parents, who have always been there for me through many difficult moments in my life and pushed me to always work hard. They have always believed in me when I did not believe in myself. While both did not receive a higher education, they have always strived to guide me to always get an education and pursue life to be someone in the world. Although their only expectation was for me to pursue an undergraduate education, I sought to learn more and pursue a master's because I wanted to make them proud. Thank you for believing in me and always supporting me unconditionally.

To my girlfriend, who stood by my side throughout this whole experience. Not only supporting my decision but also pushing me when I no longer felt like I was capable of pursuing a master's degree. These past two years, you have helped me better myself and approach my situations differently. Thank you for helping me grow and not give up on myself.

To my younger brother, thank you for giving me motivation so that we can build something together.

To all the professors who have supported me from my undergraduate career at Texas State University, to the professors in this journey.

To all mentioned, I dedicate this work to you.

ACKNOWLEDGEMENTS

The completion of this Thesis could not have been made possible without the assistance of funding opportunities, the participation of so many individuals from professors to participants. I would like to identify the following opportunities and individuals for acknowledgment and my deepest appreciation.

Without the funding opportunities of Toyota Material Handling North America (TMHNA), the project will not be made possible. The funding provided an opportunity to obtain motion capture cameras that were used for the data collection process for each experiment. Without these cameras, there will have not been the opportunity to develop the Thesis idea and be able to conduct experiments associated with motions. In addition, I want to acknowledge the team of professors involved in obtaining the funding which includes, Dr. Jesus A. Jimenez, Dr. David Wierschem, Dr. Francis Mendez, Dr. George Koutitas, Dr. Damian Valles, Dr. Semih Aslan, and Dr. Rachel R. Koldenhoven. Without them, this thesis would have not existed.

I also want to acknowledge another funding opportunity from Research Enhancement Program, that was made possible by Dr. Damian Valles, and Dr. Francis Mendez. Thank you for providing me with the opportunity to continue my work throughout my graduate school career.

I would like to start to thank my committee members for my thesis. Starting with Dr. Damian Valles, I want to thank him for believing in me and providing me an opportunity to work with him. He always provided me not only with vast knowledge in

machine learning. Through his experience and guidance, he has empowered me to recognize each problem with a purpose. Without the opportunities he has presented I would not be where I am in my life and without his mentorship, it would not have been made possible. Thank you for allowing me to give me a chance to learn and acquire a new set of skills that I never thought I would enjoy. Next, I would like to acknowledge Dr. Francis Mendez for accepting to be part of my committee a year into my thesis. He has provided me the opportunity to learn from a different side in education that engineers might not have a chance to learn. He has shown me what it is like to view data from a statistical standpoint. He is highly skilled and passionate about statistics and data analytics which proved to be true when working with him. Thank you for taking the time to be on my committee. Lastly, from my committee, I would like to thank Dr. Jesus Jimenez, for providing me an opportunity to work with him in his lab and allowed me to join a team of researchers. His lab provided a space to conduct the motion capture experiments. With his leadership, support, and encouragement to explore new opportunities, this research provided a sense of accomplishment.

In addition to the committee member involved in the project, I would like to thank Dr. David Wierschem for their constant support. Without his involvement in the experiments, he helped build different apparatuses for the cameras and also helped with the data collection process. His leadership in the lab help drive a direction to the experimental procedures. I would like to thank him for his involvement in experiments and for being there when the motivation was low.

I would also like to acknowledge, Dr. Vishu Viswanathan, the Graduate Advisor of the Ingram School of Engineering for giving me and other students for the constant guidance in pursuing master's degree. To my research colleagues at CHiPS laboratory and HiPE research group, thank you for giving me a place to grow and build ideas.

Finally, thank you to all my family and friends who supported me to finish what I started and to not give up no matter how much I wanted to. Thank you for your prayers.

TABLE OF CONTENTS

| | Page |
|--|-------------|
| ACKNOWLEDGEMENTS | v |
| LIST OF TABLES | x |
| LIST OF FIGURES | xi |
| ABSTRACT..... | xiv |
| CHAPTER | |
| I. INTRODUCTION..... | 1 |
| II. BACKGROUND INFORMATION..... | 4 |
| Industry 4.0 | 4 |
| Issues Associated with Material Handling Labor | 6 |
| Motion Capture Technology | 9 |
| III. REVIEW OF LITERATURE | 11 |
| Motion Capture and Motion Analysis..... | 12 |
| Motion Capture and Manual Material Handling..... | 16 |
| Motion Capture and Analysis of Fatigue | 18 |
| IV. DATA COLLECTION | 24 |
| Overview..... | 25 |
| Preparations for Setting Up Data Collection | 29 |
| Post Processing of Data with QTM | 36 |
| Selected Data from Motion Capture Experiments | 41 |
| V. MACHINE LEARNING DESIGN AND IMPLEMENTATION | 42 |
| Background of Artificial Neural Network | 42 |
| Recurrent Neural Network..... | 44 |
| Preparing the Sequence of Data..... | 46 |

| | |
|---|-----|
| VI. EXPERIMENTAL DESIGN | 50 |
| Computation Resources | 51 |
| Process of Merging Motion Capture and Borg RPE Values..... | 53 |
| Normalization of Motion Capture Data | 55 |
| Metrics for Evaluating Machine Learning Performance | 56 |
| Univariate Human Motion Forecasting with Machine Learning | 58 |
| Multivariate Human Motion to Predict Borg RPE Value..... | 64 |
| VII. RESULTS AND DISCUSSION | 68 |
| Univariate Analysis of Human Forecasting..... | 68 |
| Predicting Borg RPE value using MoCap Data..... | 103 |
| VIII. CONCLUSIONS..... | 113 |
| Limitations of Research | 113 |
| Delimitations..... | 114 |
| Conclusions..... | 114 |
| Contributions..... | 116 |
| Future Work | 116 |
| REFERENCES | 118 |

LIST OF TABLES

| Table | Page |
|---|-------------|
| 1. Cameras in the Optical Marker Motion Capture System Environment..... | 27 |
| 2. Lifting Experiments for Participants to Perform | 28 |
| 3. LEAP Cluster System Details [67] | 51 |
| 4. HiPE Computation Server System Details [68]..... | 52 |
| 5. Python libraries used in Thesis. | 53 |
| 6. LSTM & GRU parameter values. | 62 |
| 7. LSTM Univariate Forecasting Metrics for Subject 007 | 84 |
| 8. LSTM Univariate Forecasting Metrics for Subject 1016 | 85 |
| 9. GRU Univariate Forecasting Metrics for Subject 007..... | 102 |
| 10. GRU Univariate Forecasting Metrics for Subject 1016..... | 102 |
| 11. Borg Prediction Metrics for LSTM & GRU..... | 111 |

LIST OF FIGURES

| Figure | Page |
|---|-------------|
| 1. Industrial Revolution Timeline adapted from [1] | 4 |
| 2. Representation of Developing MSD from Fatigue of MMH Motions | 9 |
| 3. Optical Marker Motion Capture System..... | 26 |
| 4. Flow for Data Capturing | 30 |
| 5. Calibration Kit: Top Item is L-Frame, Bottom Item is 300 mm Fiber Wand..... | 31 |
| 6. XYZ Direction Corresponding to MoCap Environment that is Viewed by the Cameras..... | 32 |
| 7. Marker Set Based on Qualisys..... | 34 |
| 8. Motion Capture Lifting Experiment | 35 |
| 9. Trajectory Editor of the Polynomial fill type in QTM..... | 37 |
| 10. Trajectory Editor of the Relational fill type in QTM..... | 38 |
| 11. Lifting Experiments for Participants to Perform | 39 |
| 12. Borg RPE Value at Every Minute from Subject..... | 40 |
| 13. ANN Visual of Layers and Back-Propagation | 44 |
| 14. Feedforward vs recurrent neural network..... | 45 |
| 15. LSTM Architecture..... | 48 |
| 16. GRU Architecture..... | 49 |
| 17. Flow of Merging MoCap and Borg to CSV | 55 |
| 18. Ways to partition the dataset..... | 58 |

| | |
|--|----|
| 19. Univariate Approach for MoCap Motion Forecasting..... | 59 |
| 20. Preprocessing of MoCap data into univariate timestep sequence, an example of five timesteps ahead. | 60 |
| 21. Multivariate MoCap Data to Predict Borg Values..... | 65 |
| 22. Multivariate data preparation to create a sequence of data given 10 msec history to predict 10 msec Borg RPE value. | 66 |
| 23. LSTM 200 msec train and validation loss curves of using batch size 1..... | 70 |
| 24. LSTM 50 msec train and validation loss curves of data using batch Size 10..... | 71 |
| 25. LSTM motion forecasting 200 msec of 007 with batch size 1 | 72 |
| 26. LSTM motion forecasting 200 msec of 1016 with batch size 1 | 73 |
| 27. LSTM motion forecasting 50 msec of 007 with batch size 10 | 75 |
| 28. LSTM motion forecasting 50 msec of 1016 with batch size 10 | 76 |
| 29. LSTM 50 msec train and validation loss curves using batch size 1. | 77 |
| 30. LSTM 100 msec train and validation loss curves using batch size 10. | 78 |
| 31. LSTM motion forecasting 50 msec of 007 with batch size 1 | 79 |
| 32. LSTM motion forecasting 50 msec of 1016 with batch size 1 | 80 |
| 33. LSTM motion forecasting 100 msec of 007 with batch size 10 | 82 |
| 34. LSTM motion forecasting 100 msec of 1016 with batch size 10 | 83 |
| 35. GRU 10 msec train and validation loss curves of using batch size 1. | 86 |
| 36. GRU 50 msec train and validation loss curves of using batch size 10. | 86 |
| 37. GRU motion forecasting 10 msec of 007 with batch size 1 | 88 |
| 38. GRU motion forecasting 10 msec of 1016 with batch size 1 | 89 |

| | |
|---|-----|
| 39. GRU motion forecasting 50 msec of 007 with batch size 10 | 91 |
| 40. GRU motion forecasting 50 msec of 1016 with batch size 10 | 92 |
| 41. GRU 50 msec train and validation loss curves of using batch size 1. | 94 |
| 42. GRU 10 msec train and validation loss curves of using batch size 10. | 94 |
| 43. GRU motion forecasting 50 msec of 007 with batch size 1 | 96 |
| 44. GRU motion forecasting 50 msec of 1016 with batch size 1 | 97 |
| 45. GRU motion forecasting 10 msec of 007 with batch size 10 | 99 |
| 46. GRU motion forecasting 10 msec of 1016 with batch size 10 | 100 |
| 47. LSTM MoCap Borg Prediction Train Metrics Plot | 104 |
| 48. GRU MoCap Borg Prediction Train Metrics Plot | 104 |
| 49. Borg Prediction of Subject 1016 using LSTM. | 106 |
| 50. Borg Prediction of Subject 007 using LSTM. | 107 |
| 51. Borg Prediction for Subject 1016 using GRU. | 108 |
| 52. Borg Prediction for Subject 007 using GRU. | 109 |
| 53. Self-Reported Borg Values from Each Subject used during training. | 111 |

ABSTRACT

Industrial Revolution 4.0 is defined as the interconnection of Information, Communications Technologies (ICT) within the industry. In the occupation of laborers, stock, and material mover they are often subjected to repetitive motions that cause exhaustion (or fatigue) that could potentially lead to work-related musculoskeletal disorder (WMSD). The most common repetitive motions are lifting, pulling, pushing, carrying, and walking with load, which are also known as Manual Material Handling (MMH) operations. There has been work using a machine learning technique known as Recurrent Neural Network (RNN) to predict short and long-term motions from motion capture measurements but research in using motion capture data related to MMH to measure the fatigue needs exploration. For this research, only the lifting motion is considered. Motion data is collected as time-stamped motion data using infrared cameras at a rate of 100Hz of a subject performing repetitive lifting motion. The data is a combination of XYZ coordinates from 39 reflective markers. Along with motion data, the subject will self-report the perceived level of fatigue using the Borg scale every minute. All this data can be merged into one to further be used for analysis. Since motions occur over time for a duration of time, this data is used as input to a time-series deep learning technique known as Long Short-Term Memory and Gated Recurrent Unit models. Using these models, this research will evaluate the deep learning technique and motion capture data to perform motion analysis to forecast univariate motion data and to also predict the fatigue based on the displacement movement from each marker.

I. INTRODUCTION

We are currently in the Fourth Industrial Revolution or Industry 4.0 (I4.0), bringing the interconnection of Information Technology (IT) to create a holistic better-connected ecosystem between human-technology interactions. I4.0 uses emerging digital technology such as the Internet of Things (IoT), machine learning (ML), collaborative robots, augmented reality (AR), and big data from real-time sensors to create a cyber-physical system (CPS). This creates the interaction or connection between physical (aspects from the First and Second Industrial Revolution) to digital (aspects from the Third Industrial Revolution) allowing humans and machines to collaborate across the current industry. With ML being part of the I4.0 process, a value can be added to create solutions to augment human performance with tasks that are usually repetitive that will still require human interaction, such as in the material handling industry. Manual Material Handling (MMH) intensive companies like Amazon, Walmart, HEB, FedEx, and Toyota will require humans to continue to be part of the process as machines cannot do much of the process without human interaction. This research focuses on the Human aspect and a solution to improve their performance and safety can be integrated with I4.0 smart solution using ML.

I4.0 is centered around the idea of augmenting humans performing MMH tasks within the material handling industry. With MMH labor, the worker must repeatedly lift-lower, push-pull, walk, and carry material with different loads. Having to do these fundamental movements for a given day causes the operator to experience fatigue, which can put the worker at risk of developing a musculoskeletal disorder (MSD), often involving strains or sprains of the wrists, knees, shoulders, lower back, and upper limbs.

MSD is usually developed from overexertion the manual material handling operator endure to complete their given duties, specifically overworking their muscle-skeletal system to the point of hurting themselves. Therefore, having a sense of the exertion (fatigue) level can give a worker an indicator of performance to increase human safety by reducing their exertion levels when needed. The goal of this research is to create a smart manual material handling solution in which motions conducted by a worker can be used to predict the level of exertion (fatigue) to give them an indicator to know that they should rest and recover their musculoskeletal system reducing the amount of the individual overexerts themselves.

Using only the motion displacement (XYZ values) of the individual as an indicator of fatigue in a specific motion using ML, rate of perceived exertion (RPE), and motion capture technology, a smart solution will be developed. Motion Capture (MoCap) technology comes in different forms, but the most common for recording MMH motions use either MoCap cameras with passive markers attached to the individual or wearable Inertial Measurement Unit (IMU) sensors. For this research, the data collection process is done in a real-time motion capture environment using twelve motion capture cameras in which participants wear 39 reflective markers to track their motions, along with the Borg RPE scale representing fatigue levels as they perform an MMH operation. This research is set out to investigate the combining of the motion data (XYZ coordinates) of MMH motions conducted by subjects, with the RPE per minute of a subject to create a Recurrent Neural Network (RNN)-based ML solution (Long Short-Term Memory and Gated Recurrent Unit algorithms) to predict the fatigue based on the RPE values associated with the motion(s) and to also perform univariate human motion forecasting.

The objective is to create a solution associated with I4.0 to create a smart, holistic system between human technology to increase worker safety in the material handling industry resulting in the augmentation of the human worker.

II. BACKGROUND INFORMATION

Industry 4.0

Starting from the advancement of the steam engine in the First Industrial Revolution to the manufacturing mass production process developed in the Second Industrial Revolution to the integration of computer systems to help drive the assembly line in the Third Industrial Revolution, bringing us into the present the Fourth Industrial Revolution or I4.0 with the use of Information and Communication Technology to create Cyber-Physical Production Systems (CPSS) in the manufacturing/production domain Fig. 1 [1, 2].

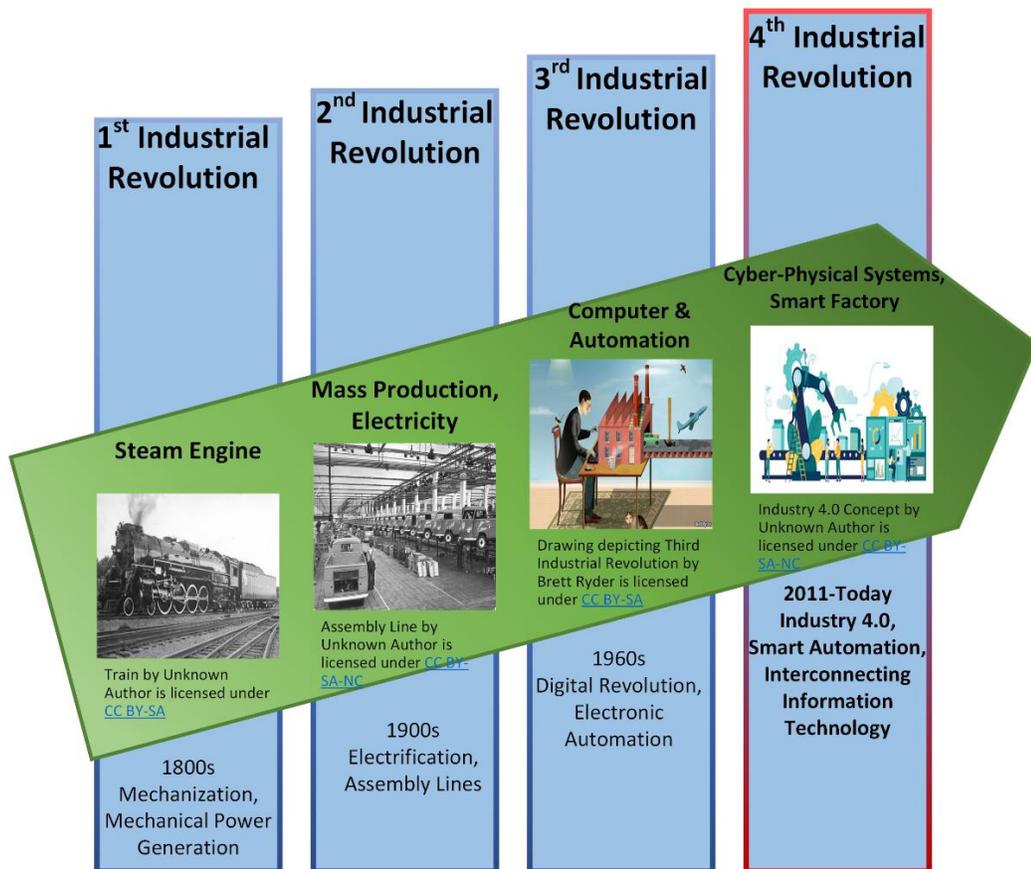


Figure 1 Industrial Revolution Timeline adapted from [1]

I4.0 was first introduced at the Hannover fair in 2011 in Germany, which has grown into what is known today as the Fourth Industrial Revolution that involves automation and data exchange of traditional industrial processes [1]. The main idea for I4.0 is to produce a 'smart' factory to improve the interconnectivity of process within the factory between IT, and the physical operational technology systems to create a CPSPS [1-3]. The goal is to use the data collection process and exchange it in real-time with the help of enabling technologies. These include IoT, Big Data analytics, ML, AR, and advancement in collaborative robots [1-3]. The use of these digital technologies will make it viable to create flexible automation processes within the industrial settings to create an intelligent process, increase product quality, increase productivity, and create a safer environment for humans. By making machines and production more autonomous, the ability of self-monitoring, self-predictive, and self-organize can reduce humans' being at risk on the factory floor [3].

Another motivation is to improve the MMH safety where operators are performing repetitive manual operations. When performing repetitive manual operations, operators may experience fatigue but will continue to work resulting in the overuse of their musculoskeletal system, leading to injuries and human error [4] in the factory. Therefore, to augment human-technology interaction, sensors can be added to the worker to monitor their fatigue with the use of real-time data analytics. The accumulation and collection of sensory data fit the criteria of what I4.0 brings to the future of smart factories.

Issues Associated with Material Handling Labor

Fatigue

Fatigue can be derived between two types, physical and mental fatigue that an individual develops over a duration of time giving the individual the sense of being tired, weak, or exhausted [5-9]. It often comes from a prolonged physical or mental activity that can be resolved or reduced with rest. In the material handling industry, workers usually perform repetitive MMH labor that involves a range of physical activities. The activities can range from motions that make them lift, lower, push, pull, walking, and/or carry objects with different weights for an extended period throughout their daily demanded tasks [5, 10, 11]. The fatigue considered for this research is tied to the physical aspect typically associated with these MMH labor motions. According to literature, fatigue is defined as the impairment of a person's capacity or performance with the task [5-9]. Fatigue can lead to numerous symptoms including discomfort that may affect the person's motor function, low motor control (increase reaction time), and reduction in physical strength that impacts the performance of a given activity [6].

Multiple measurement tools have been introduced for measuring fatigue. There are objective methods that involve the use of biometric wearable sensors (e.g., surface electromyographic, heart rate monitor) [12-14] to capture physiological indicators. These types of sensors include indicators such as heart rate, breathing rate, minute ventilation, and muscle activity [14, 15]. Fatigue can also be measured by calculating the amount of energy, measured in calories, for a given activity the person will use in activity, also known as energy expenditure, to measure the accumulation of fatigue [16].

Other forms of measuring fatigue, considered a subjective measure, is the use of self-reported measures using questionnaires, surveys, or Likert scales [7, 17, 18]. Self-reports rely on the person's perception of the activity. To name a few fatigue ratings used are Rapid Upper Limb Assessment (RULA), Brief Fatigue Inventory (BFI) chronic fatigue scale, Fatigue Symptom Inventory (FSI), and Likert Rate of Perceived Exertion (RPE) scale [7, 17]. For this research, a Likert RPE scale known as the Borg rate of perceived exertion scale is used as the measure of fatigue from the individual. The Borg scale is scaled from six representing no exertion to twenty meaning maximal exertion [18]. As previously mentioned, a Borg value is the perceived exertion by the individual themselves and where they think they stand on the scale. A Borg value is dependent on how hard the activity is to the individual performing it, and their perception of how it is affecting them physically. Both types, subjective and objective, of measurement of fatigue, will be helpful to create a solution for monitoring the human worker performance in the material handling industry.

Furthermore, incomplete recovery from fatigue can increase the risk of injury, increase human error, and decrease work efficiency [6, 9]. In the long term, if fatigue levels continue to increase in an MMH operator, without adequate time to rest, they will remain within their exhausted state causing them to develop a work-related musculoskeletal disorder (WMSD) resulting in increased worker's compensation claims and lost days [4]. Therefore, human safety in the material handling industry environment can be increased with the emerging technologies in I4.0.

Work-Related Musculoskeletal Disorder

According to the Center for Disease Control and Prevention (CDC) for Occupation Safety and Health (NIOSH), Musculoskeletal Disorder (MSD) are considered injuries of the musculoskeletal system (e.g., muscles, joints, tendons, cartilage) [19]. WMSD is the same just a factor created by the work environment and conditions. A WMSD starts in a region of the body that is overexerted in a worker to the point of creating serious bodily harm. When an MMH worker has to repeatedly use the same body region for handling heavy loads, if the skeletal muscles do not have enough recovery time or there is improper posture, the individual will be at risk of injuring themselves [10, 20].

This disorder is prevalent in the manufacturing, and material handling industry domain involving repetitive work [11]. According to the U.S. Department of Labor, Bureau of Labor Statistics [21], in 2015 20,990 workers in the occupation of laborers, stock, and material movers exhibited MSD or ‘chronic’ fatigue, which was the largest occupation in the list [10] This number increased by 4,120, to 25,110 cases in 2018, indicating that this issue is still a concern in the industry. Workers usually experience injury in their upper limbs (e.g., tendonitis, tenosynovitis bursitis), lower limbs (e.g., knees), and back [4, 11, 15, 22]. WMSD does not appear overnight and gradually develop if the overuse of the body region is continued without sufficient recovery. Hence, knowing when the operator is fatigued, they can be given adequate resting time from the motions. Fig. 2 gives a visual representation of the manual material handling motions increasing fatigue up to the point of exhaustion (represented by the arrow leading to red) and developing a WMSD.

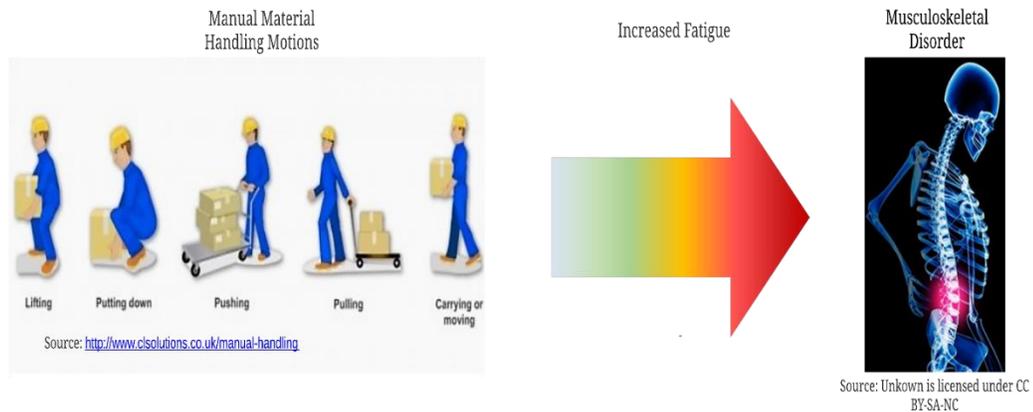


Figure 2 Representation of Developing MSD from Fatigue of MMH Motions

Motion Capture Technology

Motion Capture (MoCap) Technology was inspired by the movie industry to help speed up their animation process for developing new animation entertainment [23].

MoCap is prevalent in the video gaming industry with the intent to create realistic motions associated with human physics. MoCap is defined as the process of recording movements of an object of interest, at high frequency, in a real-time environment by tracking the position of points of interest on the object [23, 24]. This can be used to capture the posture and location of a human during physical activity for further analysis.

MoCap technology has various tracking systems that include mechanical, acoustical, inertial, magnetic, and optical systems [23, 24]. The Acoustical Systems use sound transmitters and microphones placed on specific locations to estimate the position of the points of interest. Inertial motion capture systems use an inertial sensor, containing an accelerometer and gyroscope, and rely on acceleration and angular velocity [24]. Magnetic-based MoCap systems use a set of receptors to obtain the magnetic field given by the joint position, angles, and orientation on the body. Mechanical tracking systems

are the oldest and simplest method of capturing motion by using potentiometers to measure joint orientation displacement at each point. Lastly, Optical systems can be divided into two categories, marker, and marker-less tracking systems.

An Optical Marker System (OMS) has two types of markers: passive and active markers. The passive markers are reflective markers attached to a point of interest on the actor. Using infrared high-speed cameras to capture the light reflective from the passive markers to triangulate the positions, producing 2D coordinates of the data. Proprietary software is used to compute the 3D coordinates of the markers [24] based on the *XYZ* plane in the designated environment. Alternatively, active markers use Light Emitted Diodes (LED) markers that emit light instead of reflecting it. Both are OMS but there is also an option for marker-less optical MoCap systems. These types of systems typically use computer vision algorithms for tracking the postures of the movements using a camera. The OMSs are the most accurate motion capture technology from all the ones mentioned beforehand [24].

From the mentioned MoCap technology, the two types of motion capture technology used for collecting the data from MMH motions being performed by a participant are the optical marker system and inertial motion systems. Using the motion data can be used to evaluate how the body responds to the repetitive motions conducted throughout their given task(s). Currently, this motion capture provides data for ergonomic evaluations for MMH operations [22, 25-29].

III. REVIEW OF LITERATURE

WMSD is a major health concern in the occupation of laborers and freight, stock, and material movers. In 2015-2018 these occupations accounted for the most WMSD cases compared to the other occupations [21]. In 2018, 30% of the MSDs cases involved work-related injuries that required a median of 12 days away from work (DAFW) [21]. Factors that increase the risk of MSDs include the increase of fatigue due to the intensity of the work, the frequency of repetitive motions, the duration, and the postures during each task.

In other words, the operator for a duration of time performs repetitive motions with objects of different loads meaning while constantly changing posture to adapt to the task at hand. This will cause the level of fatigue to change over time, usually in an increasing manner. Overexertion combined with repetitive movements affects the workers' performance and increases the probability of getting a work-related injury that will be considered a WMSD. Being able to monitor the fatigue during a repetitive lifting operation can be beneficial for the worker in the field of work that deals with manual material handling. In this chapter, a review of the type of human motion analysis conducted on operations with the use of MoCap technology to obtain the motions conducted within different environments. MoCap technology such as OCMS and Inertial Measurement Units (IMUs) have previously been used to conduct experiments to evaluate the motions conducted by individuals. Some of this analysis is related to forecasting motion, identifying motion, and fatigue experienced from the motions.

Motion Capture and Motion Analysis

Generation of human motion from motion capture data is a difficult task to replicate. In the past years, various motion capture datasets have been created to help model and reconstruct these motions [30-34]. In this section, a review of the research found that use motion capture datasets to perform forecast motion, detect different motions, and identify posture.

Optical Motion Capture system using markers has been widely accepted as a popular methodology for obtaining pose data. The Human3.6M (H3.6M) [32] dataset has been one of the driving factors to advancements in simulating the 3D posture of different tasks. This dataset contains about 3.6 million 3D poses collected from eleven actors (6 male, 5 female) using four cameras at a rate of 50Hz, accurate 3D joint positions, and angles from ten motion cameras and 3D laser scans of the actors. Their motivation to include all types of data was to create a dataset that simulates 3D postures seen in daily life in the world. The researchers conducted experiments on fifteen scenarios of different postures that will be seen in a real-world setting (i.e., walking, eating, smoking, waiting, greeting, etc.). Recent developments in human motion prediction have been made possible using this dataset and machine learning [19, 35-39]. There three notable prior works related that provide solutions for forecasting short-term and long-term sequences of natural human motion [19, 37, 38].

The authors in [19] exhibit the use of a recurrent neural network (RNN) model with a nonlinear encoder and decoder (ERD) for producing realistic motion from the H3.6M pose dataset. Learning from ground-truth data during training, RNNs will tend to have issues fitting to unseen data and not able to recover from the errors, therefore, they

gradually add noise to the ERD network [19] adding noise to the input when training will help with the issue (known as curriculum learning). The researchers provide a comparison with ERD and prior work that use Restricted Boltzmann Machines (CRBMs), Gaussian Process Dynamic Model (GPDM), nearest neighbor N-gram model (NGRAM), and 3 Long-Short Term Memory (LSTM) layers of 1000 units each with linear encoder and decoders to show the importance of the using a nonlinear encoder and decoder. In comparison, the N-gram model cannot generate anything outside of the training set, GPDM cannot handle the breadth of styles in training and produces unrealistic motion. The LSTM-3LR does perform relatively better than the ERD implementation for short-term motion generation but soon seems to converge to a mean pose with long-term motion forecasting because it did not encounter similar examples in training. The motion synthesis when using ERD showed that short-term motion (80msec, 160msec, 250 msec, and 320 msec) is mimicked well when comparing it to the ground- the truth of motion. Overall ERD only performs better on periodic activities (i.e., walking, smoking, etc.) than non-periodic (i.e., sitting) and produces fewer smooth completions with long-term motion predictions not staying close to the ground truth. Yet, according to the motion prediction error table in the paper, LSTM proves to have the lowest prediction error.

In [37] they introduced a structural RNNs (SRNN's) which uses Spatio-temporal graphs decomposed into different factors with each factor being assigned an RNN. Each of the RNN's is interconnected semantically to represent the structure and interaction of Spatio-temporal graphs. SRNN model adapts ideas from [19] such as noise sampling during training input to help the prediction of motion a realistic human motion. In this research, they also use the H3.6M dataset to compare their S-RNN implementation with

ERD for motion forecasting of the same experiments (i.e., walking, eating, smoking, discussion). The short-term predictions stay close to the ground truth data and the SRNN can produce a long-term motion for periodic actions and unlike ERD implementation it can forecast human motion for aperiodic actions. Comparing this to the 3-layer LSTM, it also does a slightly better job (smaller error) in motion forecasting.

Although S-RNN is a better solution compared to ERD another recent work focusing on creating a better performing long-term by authors in [38]. A single Gated Recurrent Unit (GRU) with 1024 units was created to simplify the solution. It will make the architecture simpler by reducing the number of gates, and layers. They create a sequence-to-sequence architecture with sampling-based loss to eliminate the need for hyper-parameter tuning[38]. According to the author a sequence-to-sequence architecture trains two networks: one network uses an encoder to generate an internal representation of the data inputted, the second network uses a decoder that takes the first network as input to produce the maximum likelihood estimation for prediction. They compare their implementation to LSTM-3LR, Encoder-Recurrent-Decoder (ERD) and structural RNNs (SRNNs) and the end results showed that the model suffers from discontinuities in short-term prediction but produces a realistic motion longer duration. For short-term motion dynamics a large amounts of training data are needed for ML model to be able to distinguish the motions, but this model could potentially be useful for longer sequences of motion.

Another dataset that provided advancement in simulating 3D motion is the Carnegie Mellon University (CMU) motion capture dataset [31]. This dataset contains an abundant number of motions by 144 participants, making this one of the largest motion

capture datasets. The data was captured using a Vicon motion capture system which consisted of 12 infrared MX-40 cameras at a rate of 120 Hz with a 4MP resolution and 41 markers placed on each actor. This dataset encompasses six different categories with twenty-three subcategories including human interaction, interaction with the environment, locomotion (i.e., running, walking), sports, and common situations and scenarios in the real world. Although this dataset does contain a higher number of subjects, each subject performed a different set of actions that may or may not have been performed by others. This dataset has also been used to further pursue the advancement in human motion analysis [40-42]

In [40] researchers, discuss the use of the Dynamic Forest Model (DFM) to recognize motions and predict what the action is. The authors created DFM for modeling human motion using a non-linear and non-parametric Markov model approach to form a linear system that can predict the next frame based on previous observations. They combine this Markov model with autoregressive trees (decision trees) and to overcome overfitting they use dynamic forest by creating an ensemble of trees. Each tree will then produce its prediction and average. They used the CMU motion capture dataset to generate the model sequence of motions and classified gestures. Another previous work that uses the CMU motion capture dataset for modeling human action recognition uses an unsupervised approach for detecting and isolating athletic movements [43]. The author used a PCA-like technique based on the kinematic structure of the movement data known as kinematic dimensionality reduction (KDR). The technique reduces the dimensionality of motion data to estimate the kinematic synergy for each limb joint motion They used the CMU motion capture dataset sports motions category such as jumping, kicking a

soccer ball, baseball pitching, and golf. Along with the motions of different sports, they also considered non-sports motions such as painting, hand waving, and window washing which might exhibit sport-like actions. The paper classifies an athletic movement as a movement that starts from a flexed pose, followed by quick and coherent motions. The flexed pose is detected by calculating the manipulability of the pose, which is a measure of the difference in eigenvalues of the Jacobian, therefore if the magnitude is low the pose of the limb is stretched and if the magnitude is high the pose of the limb is flexed. A limb movement is referred to as a motion of serial links connected by rotational joints and its kinematics were expressed as products of exponentials (POE). The proposed method is tested with non-athletic motion data which show that movements are detected as athletic movement due to the range of motion being performed. For example, spray painting motions are shown to be an athletic movement since it like disc throwing motion. This indicates there is a problem with their threshold and needs modifying. It also illustrates that the method does not make false-positive results since it is activated by athletic motions. Although repetitive manual handling labor can be considered non-athletic, there are still athletic motions being conducted by the operator like squatting to pick up material.

This is a good indication that joint movement obtained from motion capture data and the use of machine learning, actions can be isolated and detected.

Motion Capture and Manual Material Handling

Manual material handling workers are at risk of WMSDs due to the repeated demand on their spines and lower backs in posture from manual labor. This occupation requires workers to have to lift, lower, push, pull and carry material. This exposes the

human body to sustain awkward posture, perform repetitive motions, exposure to loads that may be unstable and hard to hold for a duration of time. Performing extraneous work like this can cause injuries, these include sprains and strains, back, wrists, arms, shoulder, neck, or legs injuries [4, 44-46]. It is important to study the postures and create a solution to monitor the risks in overexertion caused throughout the day.

Posture Recognition in Manual Material Handling Tasks

Several methods have been created in the past that used motion capture technology to collect data and provide some form of evaluation in the motion. Posture recognition and classification have been accomplished using motion capture cameras [41, 47], or wearable devices like inertial measurement units.[22, 27, 48-54]

There have been studies in which they assess in an MMH setting how the posture distribution in the height of the box and distances of a lift that could occur in factories [55]. Force sensors have on the feet while performing lifting tasks to measure the distribution of weight between each foot, concluding that one foot would have the most load during the lifting [56]. Using machine learning it is possible to accomplish posture prediction [47] or classification [25]of MMH tasks.

According to this research [47] created a solution using Artificial Neural Networks (ANN) to predict posture from motion capture data from the HUMOSIM dataset for non-repetitive manual material handling tasks. Their goal was to identify the lifting posture to assess the risk of physical injury from the MMH tasks. The authors used the initial and final hand locations of the MMH tasks to train the model with 2D and 3D coordinates. While their solution, compared to inverse kinematics, is equal in performance, the error percentage in predicting the posture was around 5 – 20%.

In [25], the authors explore the use of three classification algorithms: Linear discriminant analysis (LDA), k-nearest neighbor (k-NN), and multilayer feedforward neural network (MFNN) for classifying MMH tasks using inertial motion data and/or in-shoe pressure measurements (IPM). A self-organizing map (SOM), a neural network based on an unsupervised clustering algorithm with 1024 neurons in a 32x32 grid as the machine learning technique was used on features extracted using a sliding window of ten frames obtaining five descriptive statistics: mean, variance, min, max, and kurtosis. For the data sets that contain IPM outputs, PCA was also used to reduce the dimensions of the feature vector. They achieved a precision percentage greater than 90 percent (>90%) but their data size was relatively small with only ten participants. Each of the task classifications was performed individually on each participant.

Motion Capture and Analysis of Fatigue

In this section a review of studies that used motion capture to identify the fatigue. Some of the literature that will be discussed uses machine learning to identify the relationship between the subject's exhaustion level and motion data through time. Most of the research used a subjective measurement known as a rate of perceived exertion scale to obtain the participants level of exhaustion during the activity to predict fatigue from squat exercise[15] fatigue from outdoor running [57] fatigue related to a material handling process [22], classifying exhaustion in gait patterns [15, 58] and statistical analysis between first and last movements of manual labor [26].

As stated in [15] a set of 10 Motion Analysis Eagle cameras with 35 passive markers on the subject performing a squat exercise. The study used human-motion analysis for analyzing fatigue on a continuous level and estimating the increase of fatigue

whilst performing squatting. A parametric hidden Markov model (PHMM) and linear regression are used to estimate the level of fatigue when performing a squat exercise. Since the joint in the ankle, knee, hip, lower body, and upper body are predominantly used in the exercise, they used the angles from each of these joints. The first approach used sequential data of the squat exercise and linear regression is applied to the features. The second approach used PHMM for modeling the joint angles throughout the squat with a continuous parameter to indicate the motion change because of fatigue. Linear regression and PHMM produced similar results in accuracy for predicting the level of fatigue but PHMM is less prone to overfitting.

A paper that is more related to sports biomechanics rather than material handling operations [57] illustrates a solution for predicting fatigue in outdoor runners using an RPE scale, IMUs, and machine learning. In this research, fatigue is measured RPE from an individual running. The RPE scale the authors decided to use was the Borg scale, which is a subject fatigue measure that is indicated by the subject's perceived exertion level between 6 (no exertion), to 20 (max exertion). The RPE was measured after every lap producing nine values per trail. The inertial motion data from the inertial motion units (IMU) attached to the runner's which contains an accelerometer, gyroscope, and magnetometer. The IMUs were attached to the left/right shin bone (tibia), wrist, and arm along with a heart rate monitor. Only the accelerometer signals were considered from the IMU device attached to the runner. The RPE is added to each 10-second window sample that is collected from the IMU signals (RPE becomes the label). Since the RPE measurement is known as a linear change with the increase of time, they used linear interpolation of the values reported at the end of the previous and current lap to add to

each of the samples. They test four regression machine learning techniques to evaluate the prediction of the RPE range for the runners. The four techniques used were: 1) Gradient Boosted Regression Trees (GBRT), 2) Artificial Neural Network (ANN), 3) Linear Regression using penalty parameter of Elastic Net, and lastly 4) a Linear Regression with Least Absolute Shrinkage and Selection Operator regularization (LASSO). GBRT proved to perform the best indicating that this ML technique modeled the movement changes better using non-linear relationships. Using data only from one sensor, proved to be performing well-meaning there might not need to have multiple sensors when predicting the RPE values of the runner. Once again, as mentioned before, the Borg scale is a subjective measure of fatigue so it all depends on the participants' perceived exertion which can, in turn, be a different level of fatigue for every individual running.

In [22, 47] they use a data-driven implementation using wearable sensors to detect the physical fatigue in the subjects and to estimate the level of whole-body fatigue during a manual material handling process. They also used the Borg scale since it has been used by past literature to measure the physical fatigue in the individual. This RPE scale is asked every 10 min and written down. The participants wore four inertial measurement units (IMUs), strapped onto their right ankle, right wrist, hip, and torso. The wearable sensors recorded the XYZ data of acceleration, angular velocity, and the magnetic field, which will be used to measure the change in motion that occurred from fatigue. The RPE was defined as two binary decision rules one rule being RPE is greater than or equal to 13 (≥ 13 , conservative rule), and the second rule is RPE is greater than or equal to 15 (≥ 15 , standard based on Borg). In other words, if RPE is ≥ 13 the binary

decision is 1 same goes for ≥ 15 . The occurrence of the fatigue was modeled using these two different binary decision rules. They used the RPE Borg rating with linear regression to predict the rating. The features were extracted from the IMU sensor using the Least Absolute Shrinkage and Selection Operator (LASSO). A penalized logistic and multiple linear regression were used to perform the prediction and estimation of fatigue. The LASSO model with sampling performs better than current mathematical models and they claim to have an estimation of 100%. This creates a correlation between the IMU data and the self-reported RPE value in the participant in that it is possible to identify fatigue in MMH tasks.

Another unrelated to MMH study that classifies fatigue is [15]. The authors used machine learning to identify common features of exhaustion in gait (walking) patterns. They used Principal Component Analysis (PCA) and Fourier Transformation (FT) on all motion capture measurements obtained to reduce the dimensionality of the features. The experiments were captured using 6 VICON IR cameras with 35 passive markers attached to the subjects. They used a combination of features that include the XYZ coordinates or the joint angles from the data to classify the gait of the walkers. They used several classifiers for classifying normal and exhausted gait patterns such as Linear Discriminant Analysis (LDA), 1-Nearest Neighbor Clustering (1NN), Naïve Bayes, and Support Vector Machine (SVM). They state that they achieved 100% correct classification when using LDA, 1NN, and SVM but not Naïve Bayes. This means that an unknown gait pattern might not fit well with the model.

Next is a recent study conducted in 2019 which uses motion data from smartphone to classify the fatigue experienced from the gait patterns of individuals [58].

The motion data consisted of twenty-four participants from the university population (12 M, 12F) performing a walking pattern of 20-32 steps, followed by 16 squats. At end of each squat set, the participant gave their RPE value which is based on Borg's scale rating and then performed the same procedure again. The smartphone was attached to the shank of the participants which acted as an Inertial Measurement Unit (IMU) collecting at a frequency of 100Hz and the data was collected until they reached a Borg value of 17 or greater. Each IMU data was segmented into windows of fixed sizes along the sequence of data points to identify the segments of acceleration in the gait pattern. Then features were extracted which included, mean, coefficient of variation, maximum acceleration, acceleration range, second acceleration peak, etc. that will be used as input for an SVM classifier with a Radial Basis Function kernel. They experimented with two different numbers of classes for fatigue classification, 2 class SVM and 4 class SVM. The 2-class SVM is based on binary classification where 0 is classified as normal walking and 1 is classified as a range of 15-20 Borg RPE rating. For the 4-class SVM, the classification is based on four categories, 0 is classified as normal walking, 1 classified as a range of 7-10 Borg RPE rating, 2 classified as a range of 11-15 Borg RPE rating, and 3 classified as a range of 15-20 Borg RPE Rating. They performed gait analysis using these two types of classes to classify the fatigue which they achieved accuracies of 91% for 2-class SVM and 61% for 4-class SVM. The binary classification performed better than the 4-class SVM because there are fewer gaps between the classes making it difficult to separate them in the hyperplane. The 2-classes gap had a much bigger gap between classes when looking at the RPE value so there is higher sensitivity for identifying the gait pattern of RPE 15-20.

Lastly, research related to the manual handling of loads [26], focuses on the fatigue that is caused by manual labor. The goal of the project was to identify an indicator that shows fatigue in the participant using physical symptoms (i.e., posture change, facial expression). They used two webcams to capture the recordings of the experiments with the volunteers repetitively lifting a load. The participants had markers placed only on the upper body and the reference points were manually placed in each frame of the video and the real-world coordinates of the body position were reconstructed using Direct Linear Transformation. They gathered rates of perceived exertion (RPE) using the modified Borg scale at the end of the experiment. The Borg scale indicates the level of fatigue from 0 to 10 (0 being none, and 10 being maximal exertion). Once the experiment was over the researchers watched the recordings to observe the physical changes throughout the recording. To improve this aspect, they could have also recorded the biometrics of the individual (i.e., heart rate, breath pattern, etc.) to help validate their observations from the recordings. They performed statistical motion analysis between the first and last movement of the wrists, elbows, and shoulders using XYZ, velocity, and acceleration. They observed that the range of movement during the lifting changed individually with others having different acceleration when lifting. This research has proven to show that movement changes are prone to change when performing lifting and lowering movements.

IV. DATA COLLECTION

MoCap was invented to help speed up the animation process for developing future content and is widely used in the gaming, entertainment, biomechanics, and ergonomic industries [22-25, 28, 29]. MoCap is defined as recording movements at a high frequency in a real-world setting by tracking the positions of points of interest attached to an object or person [23, 24, 59]. There are various tracking technologies used for motion capture; they include acoustical, inertial, magnetic, mechanical, and optical systems [24]. The acoustical systems use sound transmitters and microphones placed in specific locations to estimate the points of interest. Inertial motion capture systems use an inertial sensor (containing an accelerometer and gyroscope) that relies on acceleration and angular velocity [22, 27, 28]. Magnetic-based motion capture systems use a set of receptors to obtain the magnetic field given by the body's joint position, angles, and orientation. Mechanical tracking systems are the oldest method of capturing motion by using potentiometers to measure joint orientation displacement at each point of interest on the actor [24].

Finally, the Optical Motion Capture System (OMCS) is divided into passive and active markers. The passive markers are reflective indicators attached to the main points of interest on the person or object. Passive marker positions are triangulated using measurements captured from infrared high-speed camera reflections off the marker producing 2D coordinates of the data. Proprietary software is used to compute 3D coordinates of the markers [23]. Alternatively, active markers use light-emitting diode (LED) indicators that emit a light of their own instead of reflecting light. Some OMCS do not require physical markers, known as optical marker-less capture systems that use

computer vision algorithms to track the movements. OMCS technologies have been implemented to collect data from motions performed by participants performing manual handling tasks for Industry 4.0 applications [25, 27-29]. It can also analyze the correlation between exhaustion and manual handling operations[22, 60]. This research aims to analyze motion data to predict fatigue levels based on the Borg rate of perceived exertion. An Optical Marker Motion Capture System (OMMCS) system with passive markers and infrared cameras has been developed, as shown in Fig. 3. The following chapter is split into four subsections describing the various components of data collection for the research. The first section, *Overview*, gives an overview of the OMMCS technology involved in this research. The following section, *Preparations for Setting Up Data Collection*, is a detailed breakdown for performing an experiment and the various aspects involved to capture the data. The third section, *Post Processing of Data Using Qualisys Tracking Manager (QTM)*, discusses proprietary software paired with the OMMCS to process and clean the data after capture is completed and describes the type of data is captured. Lastly, in the *Selected Data from Motion Capture Experiment* section, a description of the data selected from the experiments to develop a machine learning model is discussed.

Overview

The data collection environment contains various components for capturing the movements associated with Manual Material Handling. An OMMCS is used to obtain participants' motions in a real-time environment, as shown in Fig. 3. In the OMMCS, three additional Qualisys (Oqus 110+) cameras were added to the previous group of cameras, making up a total of twelve infrared cameras (Oqus 110+, 510+, and M3) [60].

One video camera (210c) is shown in Table 1.

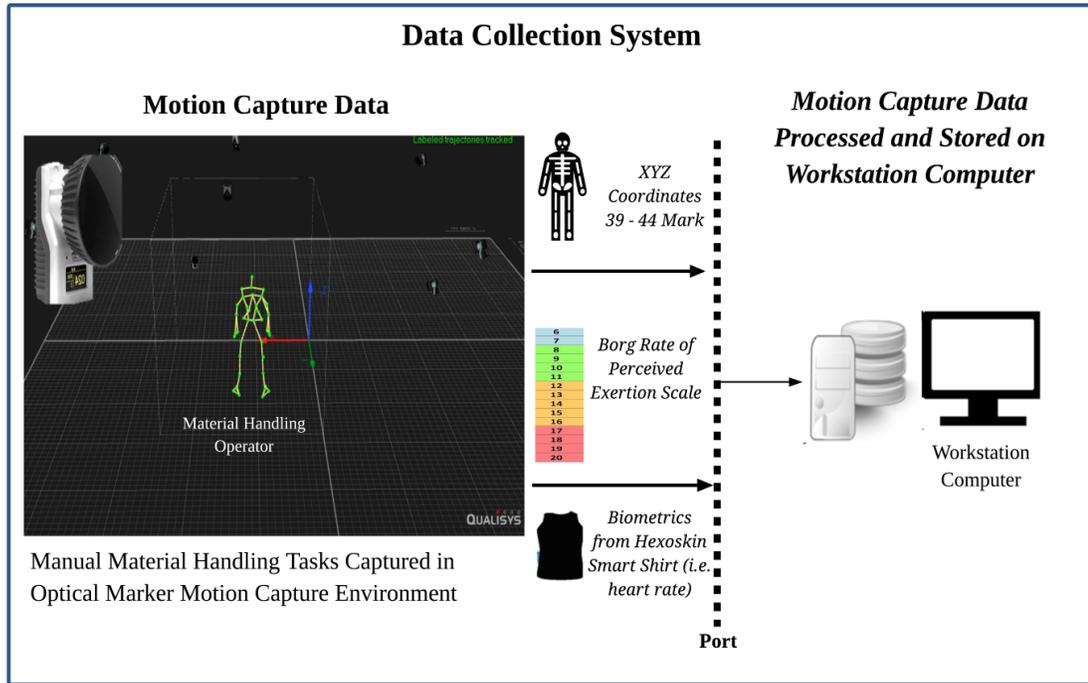


Figure 3 Optical Marker Motion Capture System

These cameras are confined to space in the Ingram School of Engineering lab designated for the subject to perform the tasks, so adding more cameras added vantage points to the lower portion of the body that was continually missed while the participant performed experiments. This research involves human subjects; therefore, the experimental procedures had to be approved by the Texas State University Institutional Review Board, and informed consent was obtained from the participant before conducting any experiments.

Table 1 Cameras in the Optical Marker Motion Capture System Environment

| Qualisys Camera Type | Quantity |
|-----------------------------|-----------------|
| Oqus 110+ | 3 |
| Oqus Video 210c | 1 |
| Oqus 510+ | 6 |
| Miquis M3 | 2 |

The task(s) needed to be performed by a subject essentially mimics a manual material handling movement in which the performer operates based on various heights and weights shown in Table 2. For measuring the body positions, 39 markers were attached with double adhesive-sided tape on the participant. Using QTM, proprietary software gives the ability to use the Qualisys cameras to capture 2D/3D/6DOF motion capture data in real-time [61]. The QTM software with the cameras will capture and generate real-time 3D coordinates of each marker position at a rate of 100 Hz (100 frames per second) while the participant performs each motion. Each experiment conducted by the participant was based on the “Snook Table” at various heights and weights [62]. The motion reference will be on the lifting motion from the floor to the knuckle height for these experiments.

Table 2 Lifting Experiments for Participants to Perform

| Gender | Range | % Industry | Weight (kg) | The distance of Lift (cm) | Interval Between Lift (sec) |
|---------------|---------------|-------------------|--------------------|----------------------------------|------------------------------------|
| Female | Floor-Knuckle | 50 | 15 | 25 | 9 |
| Female | Floor-Knuckle | 50 | 13 | 51 | 9 |
| Female | Floor-Knuckle | 50 | 12 | 76 | 9 |
| Female | Floor-Knuckle | 50 | 16 | 25 | 14 |
| Female | Floor-Knuckle | 50 | 14 | 51 | 14 |
| Female | Floor-Knuckle | 50 | 13 | 79 | 14 |
| Male | Floor-Knuckle | 50 | 24 | 25 | 9 |
| Male | Floor-Knuckle | 50 | 20 | 51 | 9 |
| Male | Floor-Knuckle | 50 | 19 | 76 | 9 |
| Male | Floor-Knuckle | 50 | 28 | 25 | 14 |
| Male | Floor-Knuckle | 50 | 24 | 51 | 14 |
| Male | Floor-Knuckle | 50 | 22 | 76 | 14 |

Each participant also provided their fatigue level using a rating of perceived exertion based on the Borg scale from six to twenty every 60 seconds in the experiment. In this case, six was no exertion perceived by the individual, while twenty is maximal exertion [18]. Every participant performed the lifting task for a different duration of time until they perceived on their own that they could not go any further in the experiment. In other words, from the data points standpoint, every participant motion data produced a different number of frames/samples with a mean of 106,025, a minimum of 25116, and a maximum of 195,799 frames between the ten participants for lifting only motion. The OMCS can capture each of the 3D coordinates of the markers(s) using *Qualisys* proprietary software, *Qualisys Tracking Manager (QTM)* [63]. Another type of data

collected from the individual that is not going to be used for this research is the biometrics data, i.e., the heart rate and breathing rate in real-time using a Hexoskin Smart Shirt[64]. The biometric data is part of the contributed data collected while performing these experiments that will not be used to develop a predictive fatigue model.

Preparations for Setting Up Data Collection

The starting basics will be discussed further along with the activities done to get to the point of capturing data and as represented by Fig 4. The setup ranges from starting the system to calibrating the data to preparing the experiment and the subject's reflective markers' placement. Lastly, the general flow of what the experiment process is during the experiments. Fig. 4 shows the flow for the beginning MoCap session in the lab. As shown, there are five steps to get to the point of start for capturing the data. These steps must be performed chronologically to complete the MoCap setup. Once these steps are completed, the data capture can begin.

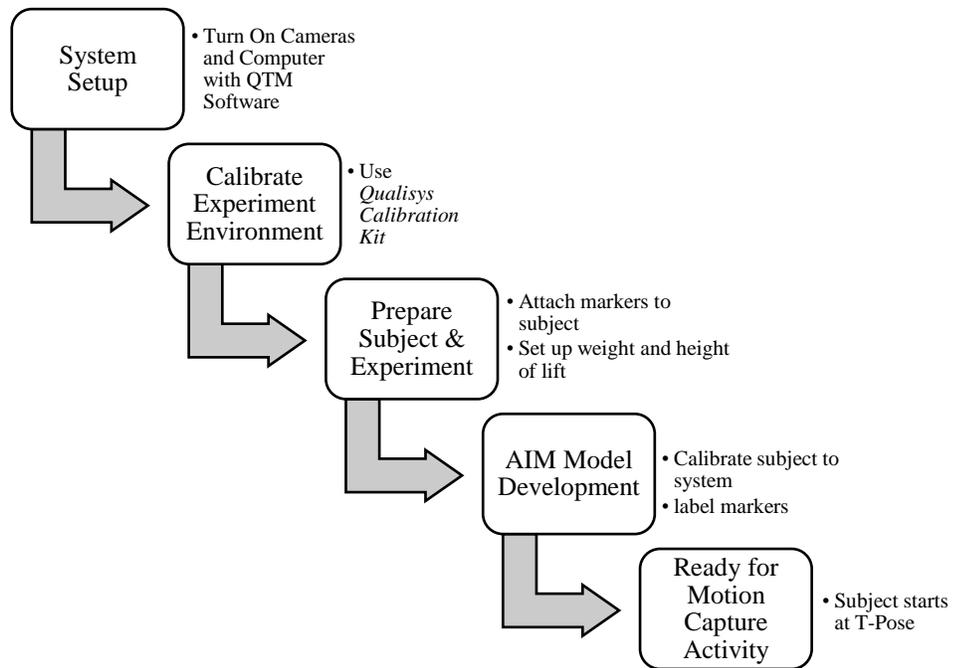


Figure 4 Flow for Data Capturing

Calibration of Environment

Firstly, the environment must be calibrated to give the origin point where the experiment area begins to give the bounding area where the motions will be captured. This is done using the calibration kit provided from *Qualisys* containing two apparatus, an L-frame and a 300 mm carbon fiber wand, as shown in Fig. 5 [63].



Figure 5 Calibration Kit: Top Item is L-Frame, Bottom Item is 300 mm Fiber Wand.

The L Shape is placed on the designated spot on the floor in the viewpoint of cameras' viewpoint, indicating the origin, which will let the calibration distinguish where the experiment area begins. Using the wand apparatus and moving it in the x (left to right), y (forward and backward), and z (up and down) direction indicated by the direction of the axis shown in Fig 6. This ensures while calibration mode is on in the QTM software, the cameras can triangulate the area of interest.

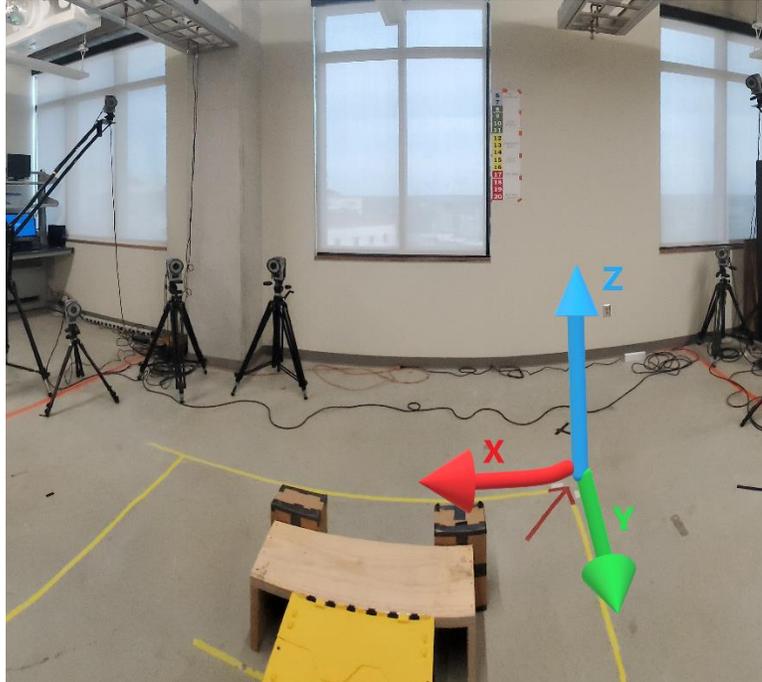


Figure 6 XYZ Direction Corresponding to MoCap Environment that is Viewed by the Cameras.

The software creates the calibration area by forming a bounded box in which the motions will only be captured. Therefore, any motions performed outside this bounded box will not be captured and will not show up in the data capture. This creates a concise way for cameras to view the markers attached to a subject in the experiment area.

Setting Up the Experiment and Subjects

Once the calibration is performed, the next step is to set up the experiment being performed by the subject based on the activity's parameters. In this case, it will be the lifting operations. Lifting operations involve loads with different weights based on a percentage of the gender population and distance of the vertical lift [1][3]. Each lifting task contained three parameters: the interval between each lift, the box's weight, and the distance of the lift from floor level to knuckle height. In the data collection process, the participant must lift a box with the load derived from the "Snook Tables" [3]. These

tables outline the maximum acceptable weight for specified manual material handling motion for the female and male population. Each participant was to perform the experiments described in Table 2, which lists the experiment's setup parameters by setting up the table, the box with corresponding weight, and the interval timer.

Once the experiment setup is completed, markers are placed on the participant based on a modified version of the Qualisys Animation Marker Set provided within the *Qualisys Tracking Manager* shown in Fig. 3. The marker set contains only 39 markers that will be used to capture the motions during the experiments. In Fig. 7, the marker set is edited to showcase the 39 markers, with left and corresponding right markers. Each of the subjects also wore a back-support brace to mitigate the effects of lifting the load provided for each experiment, which is also used by workers dealing with manual material handling operations in the industry. The double-sided adhesive tape was used for attaching the markers, with some attached to the skin, while others attached to the clothing and back brace.

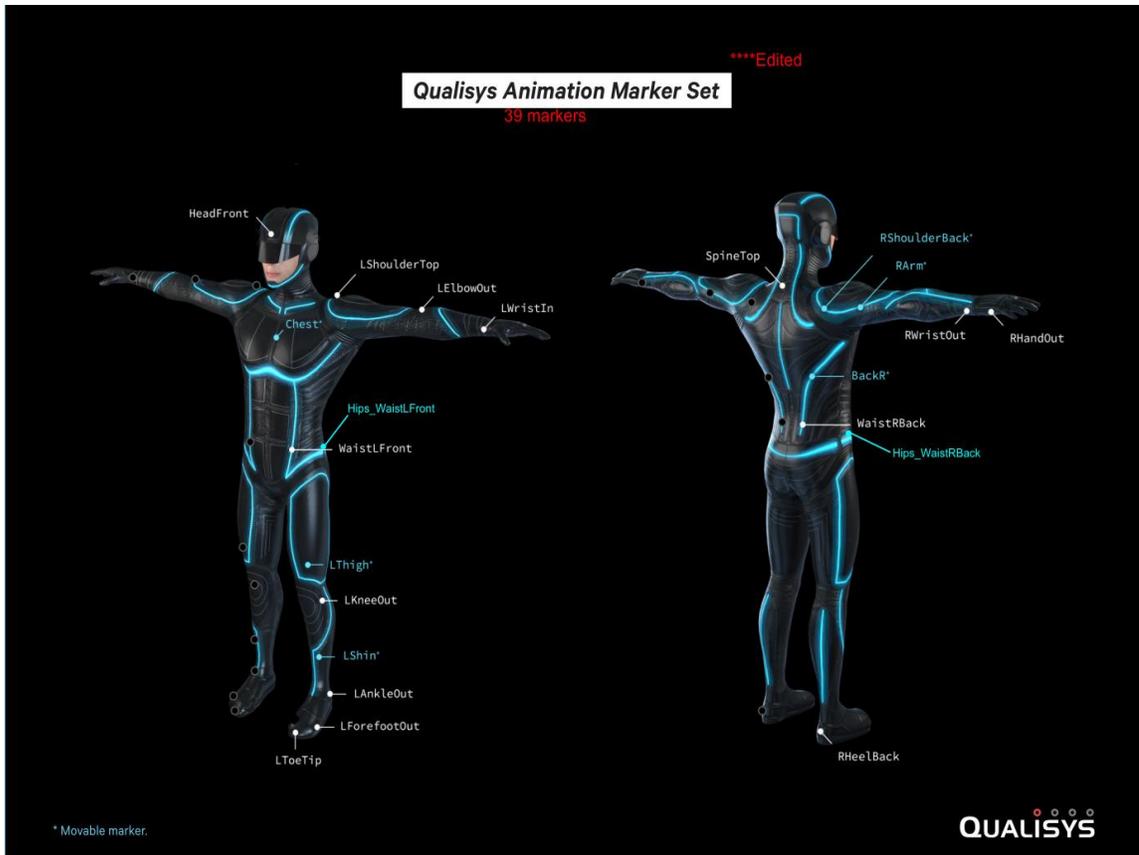


Figure 7 Marker Set Based on Qualisys

Automatic Identification Markers (AIM) of Subject

Once the calibration is set for the area and the subject is set up with the markers, the next step is to generate an Automatic Identification Markers (AIM) [63] model tailored to the person performing the motions. Essentially calibrating the system to the person and manually labeling the markers to the corresponding joint used in real-time marker identification. The AIM helps the data collection process that allows the system to quickly identify the markers attached to the individual in real-time; without this, the post-processing will have a significant number of unidentified trajectories that will have to be manually labeled for each marker on the person.

Capture of Experiments

The steps beforehand must be completed, and the data collected captured in real-time begins with the subject standing at T-Pose to start data collection. The motion positions are processed in the QTM software for the experiment's duration, and motion is conducted based on the interval. Unlike the QTM that QTM automatically processes, the Borg RPE values are manually collected onto an Excel spreadsheet every 60 seconds. Every 60 seconds, the subject self-reports the lab assistant what their RPE value is and notes it down into the Excel file. The experiment stops once the individual believes they are fatigued based on their own perceived motion in the activity and signals the lab assistant by standing back into T-Pose to stop the capture in QTM. Fig. 8 shows an example of the lifting motion capture experiment and how the environment looks.

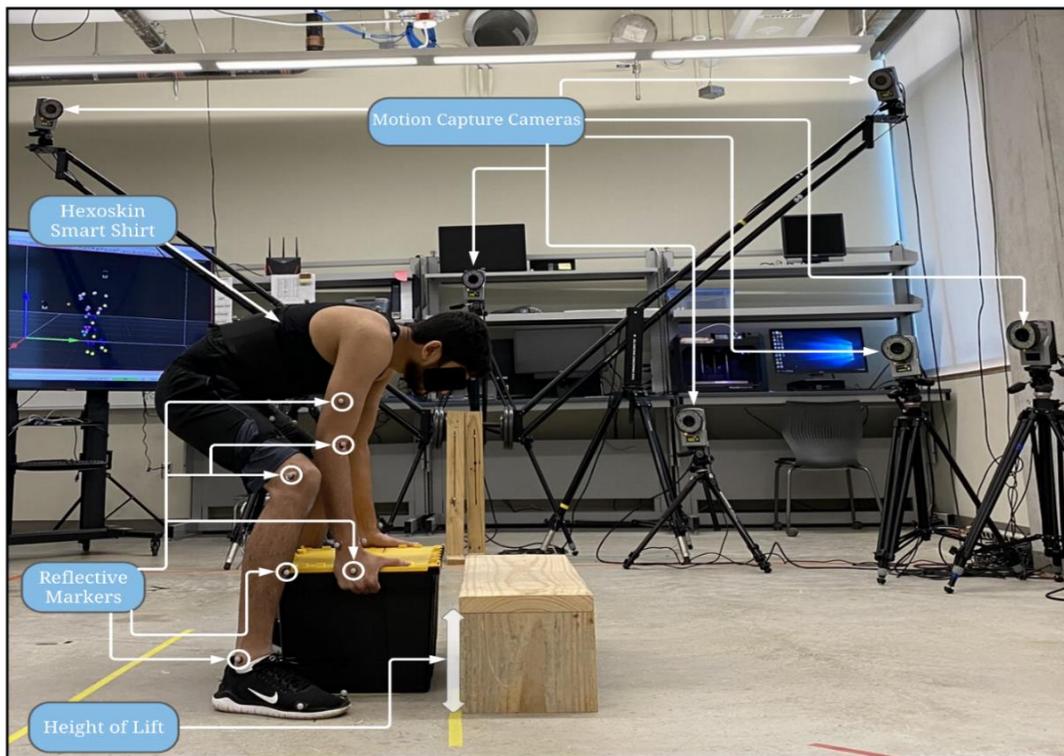


Figure 8 Motion Capture Lifting Experiment

Post Processing of Data with QTM

After completing the capture using QTM, the data is post-processed. The cameras capture every marker data at a speed of 100 frames every second, meaning every frame is captured every 0.01 seconds. Each frame contains each of the marker positions in x, y, and z-direction, with a total of 117 data points each frame (39 markers * 3 directions). The tracking system suffers from missing or gaps in the data for the marker(s). These gaps are usually automatically filled up to ten missing frames by the *QTM* software based on the AIM model created, but there are instances where the software cannot track the marker. The leading cause is by obstructing the view of marker(s) or having fallen when performing a movement, and the cameras lose track causing the system to have missing frames or gaps in the data. To overcome this issue of missing data/frames, *Qualisys* has provided tools (algorithms) to fill in the data's missing gaps. Looking at the trajectory editor, the options provided in QTM are Static, Linear, Polynomial, Relational, Virtual, and Kinematic. Filling in the missing gaps using QTM reconstructs the data for missing frames from the marker(s) that have gaps, all performed manually. After interpolating the data, the data is simulated on the 3D plane in QTM to visualize if the filled data's trajectory corresponds to the motion conducted.

Depending on the number of frames missing, for 100 or fewer missing frames, the Polynomial algorithm was used to fill in the gaps. Polynomial uses an algorithm to join the X, Y, and Z trajectory curves from the beginning of the gap to the other end[61]. This algorithm smooths the data, so this can only be used when there is trajectory data on both sides of the gap. It relies on the previous and later data to interpolate the missing trajectory curves. For this reason, long gaps more extensive than 100 frames are

disregarded because they will not reconstruct the trajectory of the curves that correspond to the motion being performed. Fig. 9 shows an example of the trajectory editor with the Polynomial fill type selected. For missing gaps of 100 frames or greater, Relational was used to fill in gaps for the X, Y, and Z trajectory curves[61]. Just like the Polynomial algorithm, the Relational fill type connects both sides of the gap. The difference is that it uses surrounding markers' movement to create a local coordinate system to follow the trajectory path and fill in the missing curves. Since this requires an origin marker to create the local coordinate system, this is ideal for more significant gaps because there will always be other markers in the surrounding area to fill in the missing data in the trajectory curves. An example of the trajectory editor for the Relational fill type is shown in Fig. 10.

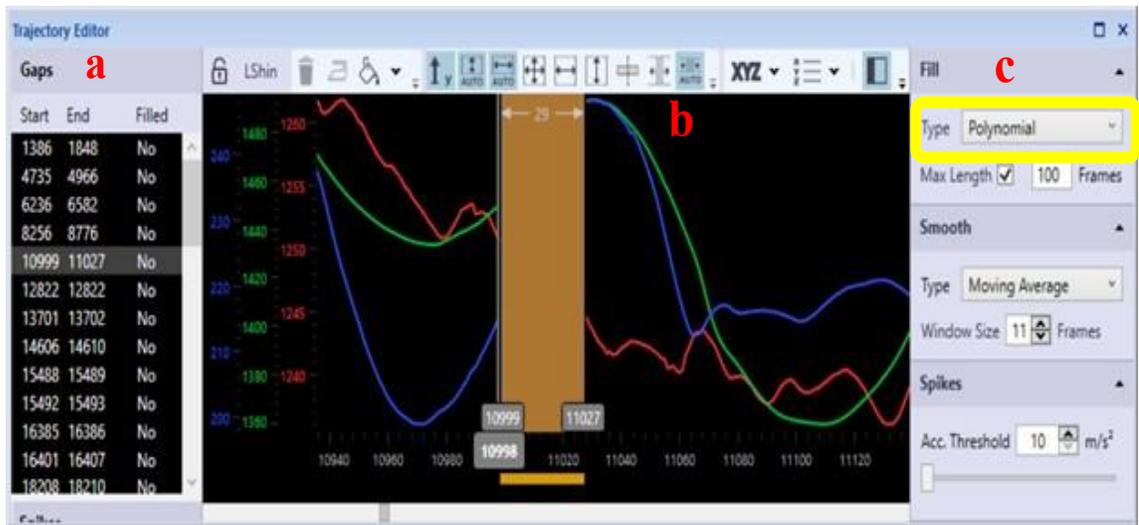


Figure 9 Trajectory Editor of the Polynomial fill type in QTM. a) Gaps to Fill shown in the left sidebar, b) the trajectory curve is shown in the middle with a missing gap, and c) gap-fill type is shown on the right sidebar.

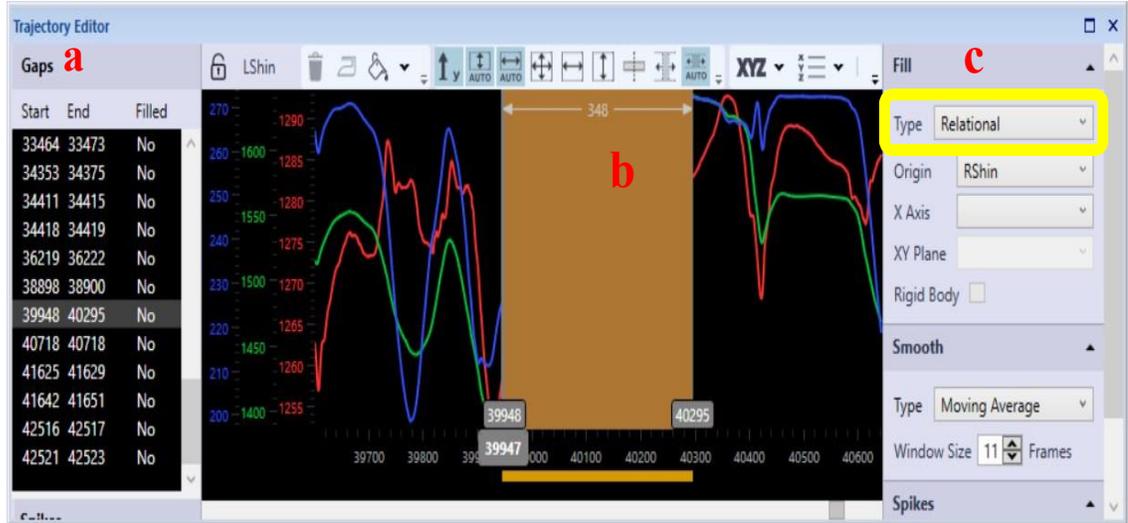


Figure 10 Trajectory Editor of the Relational fill type in QTM. a) Gaps to Fill shown in the left sidebar, b) the trajectory curve is shown in the middle with a missing gap, and c) gap-fill type is shown on the right sidebar.

After the data is post-processed in QTM, the data can be exported, and for this purpose, the data is exported into a Tab Separated Value (TSV) file, which holds the 3D XYZ positional coordinates of every marker separated by tabs. Every three columns contained the x, y, and z of each marker shown in Fig. 11. Since the cameras will capture and generate in real-time 3D coordinates of each marker position at a rate of 100 Hz (100 frames per second), the timestep of each frame is 0.01 seconds. QTM does have other formats to export, but for this purpose, the files with TSV format will be used for obtaining the motion capture data from the motions performed by the subjects. Each TSV

file corresponds to a separate experiment completed by a subject.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|----|------------------|------------|-------------|--------------|--------------|-----------|-----------|----------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | NO_OF_FRAMES | 157837 | | | | | | | | | | | | |
| 2 | NO_OF_CAMERAS | 9 | | | | | | | | | | | | |
| 3 | NO_OF_MARKERS | 46 | | | | | | | | | | | | |
| 4 | FREQUENCY | 100 | | | | | | | | | | | | |
| 5 | NO_OF_ANALOG | 0 | | | | | | | | | | | | |
| 6 | ANALOG_FREQUENCY | 0 | | | | | | | | | | | | |
| 7 | DESCRIPTION | -- | | | | | | | | | | | | |
| 8 | TIME_STAMP | 2019-10-11 | 718994.808 | | | | | | | | | | | |
| 9 | DATA_INCLUDED | 3D | | | | | | | | | | | | |
| 10 | MARKER_NAMES | HeadFront | Chest | LShoulderTop | RShoulderTop | LElbowOut | RElbowOut | LWristIn | RWristIn | WaistLFront | WaistRFront | LThigh | RThigh | LKneeOut |
| 11 | Frame | Time | HeadFront X | HeadFront Y | HeadFront Z | Chest X | Chest Y | Chest Z | LShoulderTop X | LShoulderTop Y | LShoulderTop Z | RShoulderTop X | RShoulderTop Y | RShoulderTop Z |
| 12 | 1 | 0 | 1190.372 | 2099.4 | 1757.102 | 1084.264 | 2132.29 | 1456.75 | 1250.35 | 2228.163 | 1573.157 | 915.767 | 2234.57 | 1554.248 |
| 13 | 2 | 0.01 | 1190.375 | 2099.412 | 1757.127 | 1084.203 | 2132.274 | 1456.71 | 1250.29 | 2228.188 | 1573.096 | 915.732 | 2234.509 | 1554.198 |
| 14 | 3 | 0.02 | 1190.339 | 2099.356 | 1757.119 | 1084.268 | 2132.334 | 1456.67 | 1250.3 | 2228.319 | 1573.175 | 915.699 | 2234.469 | 1554.139 |
| 15 | 4 | 0.03 | 1190.313 | 2099.339 | 1757.17 | 1084.306 | 2132.376 | 1456.63 | 1250.269 | 2228.409 | 1573.234 | 915.696 | 2234.421 | 1554.082 |
| 16 | 5 | 0.04 | 1190.36 | 2099.554 | 1757.358 | 1084.268 | 2132.289 | 1456.72 | 1249.988 | 2228.606 | 1573.186 | 915.588 | 2234.401 | 1553.907 |
| 17 | 6 | 0.05 | 1190.267 | 2099.471 | 1757.191 | 1084.249 | 2132.297 | 1456.59 | 1249.894 | 2228.679 | 1573.174 | 915.55 | 2234.362 | 1553.854 |
| 18 | 7 | 0.06 | 1190.269 | 2099.473 | 1757.242 | 1084.215 | 2132.269 | 1456.57 | 1249.804 | 2228.728 | 1573.185 | 915.506 | 2234.309 | 1553.787 |
| 19 | 8 | 0.07 | 1190.255 | 2099.448 | 1757.195 | 1084.235 | 2132.306 | 1456.51 | 1249.816 | 2228.823 | 1573.213 | 915.446 | 2234.225 | 1553.763 |
| 20 | 9 | 0.08 | 1190.269 | 2099.431 | 1757.27 | 1084.232 | 2132.316 | 1456.46 | 1249.756 | 2228.85 | 1573.273 | 915.41 | 2234.145 | 1553.74 |
| 21 | 10 | 0.09 | 1190.184 | 2099.31 | 1757.214 | 1084.216 | 2132.279 | 1456.41 | 1249.62 | 2228.812 | 1573.206 | 915.347 | 2234.066 | 1553.733 |
| 22 | 11 | 0.1 | 1190.169 | 2099.24 | 1757.167 | 1084.221 | 2132.289 | 1456.38 | 1249.661 | 2228.901 | 1573.276 | 915.364 | 2234.011 | 1553.721 |
| 23 | 12 | 0.11 | 1190.202 | 2099.222 | 1757.118 | 1084.197 | 2132.211 | 1456.33 | 1249.623 | 2228.905 | 1573.317 | 915.331 | 2233.919 | 1553.697 |
| 24 | 13 | 0.12 | 1190.208 | 2099.157 | 1757.018 | 1084.196 | 2132.198 | 1456.29 | 1249.556 | 2228.891 | 1573.261 | 915.299 | 2233.855 | 1553.665 |
| 25 | 14 | 0.13 | 1190.23 | 2099.08 | 1757.037 | 1084.215 | 2132.225 | 1456.24 | 1249.443 | 2228.845 | 1573.229 | 915.266 | 2233.792 | 1553.599 |
| 26 | 15 | 0.14 | 1190.199 | 2098.965 | 1757.009 | 1084.161 | 2132.138 | 1456.23 | 1249.395 | 2228.832 | 1573.261 | 915.234 | 2233.705 | 1553.593 |
| 27 | 16 | 0.15 | 1190.208 | 2098.858 | 1756.965 | 1084.104 | 2132.072 | 1456.17 | 1249.371 | 2228.835 | 1573.321 | 915.189 | 2233.602 | 1553.584 |
| 28 | 17 | 0.16 | 1190.197 | 2098.744 | 1756.876 | 1084.147 | 2132.09 | 1456.06 | 1249.326 | 2228.81 | 1573.298 | 915.139 | 2233.504 | 1553.512 |
| 29 | 18 | 0.17 | 1190.246 | 2098.66 | 1756.889 | 1084.152 | 2132.03 | 1456.05 | 1249.258 | 2228.774 | 1573.304 | 915.105 | 2233.388 | 1553.446 |
| 30 | 19 | 0.18 | 1190.196 | 2098.501 | 1756.844 | 1084.117 | 2132.01 | 1455.98 | 1249.237 | 2228.747 | 1573.248 | 915.054 | 2233.295 | 1553.401 |
| 31 | 20 | 0.19 | 1190.16 | 2098.349 | 1756.811 | 1084.103 | 2131.925 | 1456.02 | 1249.179 | 2228.684 | 1573.319 | 914.989 | 2233.151 | 1553.315 |
| 32 | 21 | 0.2 | 1190.216 | 2098.294 | 1756.815 | 1084.115 | 2131.865 | 1455.89 | 1249.041 | 2228.582 | 1573.241 | 914.954 | 2233.018 | 1553.226 |
| 33 | 22 | 0.21 | 1190.228 | 2098.196 | 1756.837 | 1084.084 | 2131.76 | 1455.85 | 1249.009 | 2228.534 | 1573.261 | 914.929 | 2232.904 | 1553.214 |
| 34 | 23 | 0.22 | 1190.236 | 2098.067 | 1756.804 | 1084.064 | 2131.681 | 1455.84 | 1248.957 | 2228.468 | 1573.236 | 914.882 | 2232.753 | 1553.144 |
| 35 | 24 | 0.23 | 1190.221 | 2097.935 | 1756.755 | 1084.108 | 2131.641 | 1455.78 | 1248.956 | 2228.432 | 1573.329 | 914.84 | 2232.624 | 1553.092 |
| 36 | 25 | 0.24 | 1190.196 | 2097.783 | 1756.762 | 1084.096 | 2131.545 | 1455.74 | 1248.915 | 2228.357 | 1573.369 | 914.819 | 2232.484 | 1553.063 |

Figure 11 Lifting Experiments for Participants to Perform

Along with the TSV files, the Borg values stored in the Excel sheet will indicate fatigue from the motions. An example of the Excel file is shown in Fig. 12, showing a column for a minute and a column for the Borg value corresponding to each minute the subject reported the data. These values will need to be interpolated to every frame with

every 6,000th frame starting with a different Borg RPE value. This is because 6,000 frames are equivalent to 60 seconds in the motion data since each frame's rate is 100 frames per second (100 frames/second * 60 seconds).

| | A | B |
|----|--------|--------------|
| 1 | Minute | Borg's Scale |
| 2 | 0 | 6 |
| 3 | 1 | 6 |
| 4 | 2 | 8 |
| 5 | 3 | 8 |
| 6 | 4 | 9 |
| 7 | 5 | 9 |
| 8 | 6 | 10 |
| 9 | 7 | 10 |
| 10 | 8 | 11 |
| 11 | 9 | 11 |
| 12 | 10 | 12 |
| 13 | 11 | 13 |
| 14 | 12 | 14 |
| 15 | 13 | 14 |
| 16 | 14 | 14 |
| 17 | 15 | 14 |
| 18 | 16 | 14 |
| 19 | 17 | 15 |
| 20 | 18 | 15 |
| 21 | 19 | 15 |
| 22 | 20 | 16 |
| 23 | 21 | 17 |
| 24 | 22 | 17 |
| 25 | 23 | 17 |
| 26 | 24 | 18 |
| 27 | 25 | 18 |
| 28 | 26 | 19 |

Figure 12 Borg RPE Value at Every Minute from Subject

Selected Data from Motion Capture Experiments

With the limited amount of participant and their availability, some of the experiments were not completed. Therefore, only one set of parameters is considered to create a solution as they have a more significant number of subjects. The vertical lift parameter is the distance from the floor level to the knuckle height of 51 cm with a nine-second interval for both genders, highlighted with grey on Table 2. This provides ten participants' data (seven males, three females) for lifting only experiments (motion data and RPE values) for this work. The data will be split for the machine learning process of training and testing the model, which will be discussed further using these lifting experiments.

V. MACHINE LEARNING DESIGN AND IMPLEMENTATION

In this chapter, the machine learning algorithms that were used for applying deep learning for two applications, 1) forecasting human motion and 2) predicting the exhaustion level (Borg scale values) based on the displacement of motions obtained from data collection. This covers a brief background on Artificial Neural Network which is the workhouse of deep learning algorithms like the Recurrent Neural Network (RNN). Further, into the chapter, the RNN is also discussed in-depth and how it can be used for time-series applications. Finally, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms were used to investigate the possibility to forecast motion using a single marker and to also use motion data to predict the fatigue level based on the changes in the motion.

Background of Artificial Neural Network

ML is a tool used to create new solutions that can be used to build applications for Artificial Intelligence. Machine learning is considered the first subset of artificial intelligence and is achieved by providing a system the ability to learn from a set of data patterns to predict an outcome without human intercession. One way to train an ML algorithm is supervised learning. For supervised learning, the machine is trained using a given input data that has been labeled to the correct output data. The machine will find patterns between the input and output values, which can then be used on a new set of input values.

Neural Networks can be used in a supervised setting which makes it ideal to use in this case since there will be a set of labeled data that will be used to create machine learning algorithms. A NN is a machine learning algorithm that recognizes patterns in a

set of data through the process that mimics the human brain processing data using our system of neurons. ANN in the computational world has nodes that act like neurons in the brain which are just the mathematical nodes that work in unison to find the patterns and change the internal weights within the nodes that can be stacked in layers with other nodes. A common way NN works is the feed-forward network type that follows one pattern. The data will begin in the Input layer that receives the input information, then follow into the hidden layers where all the computational nodes are located followed by the output layer which gives the result based on the data set provided as shown in Fig 13. The way NN learns is with an important aspect known as backpropagation. It is a way of tuning the internal weights and biases. NN uses backpropagation to calculate the loss backward layer by layer in which it calculates the gradient of error and aims to minimize the loss by adjusting the weights and biases.

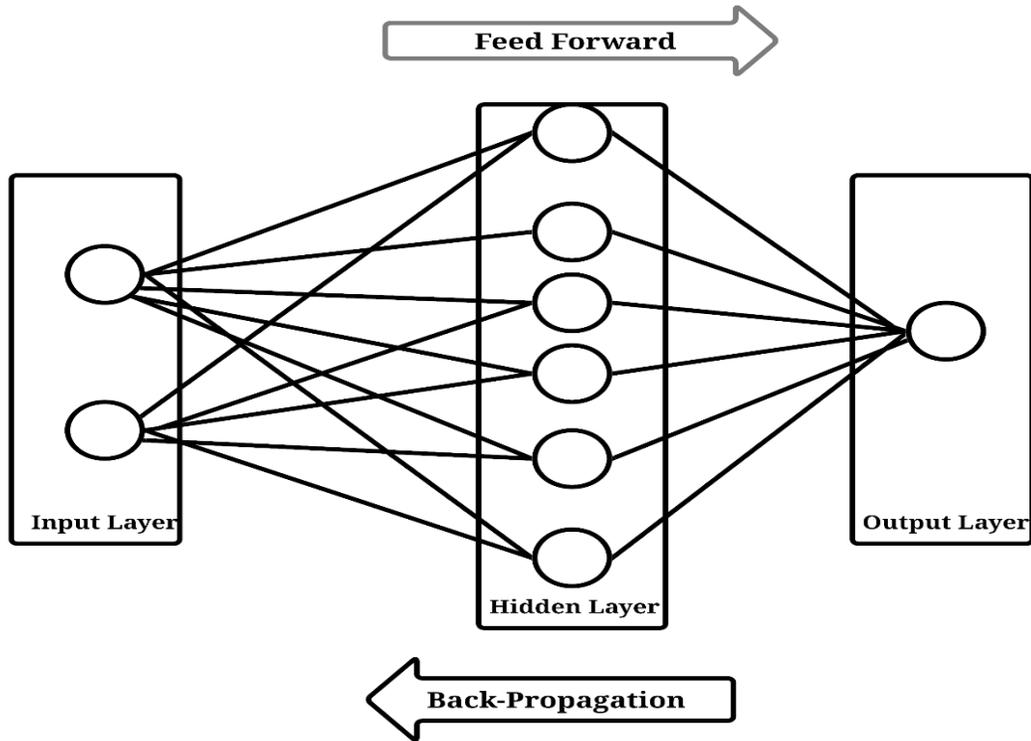


Figure 13 ANN Visual of Layers and Back-Propagation

This feedforward type network will only pass the information through these layers, therefore, a standard NN will not learn previous observations that are sequentially in order. NN is the foundation of deep learning a powerful tool in machine learning that can help solve complex problems related to computer vision, speech recognition, predictive modeling, etc.

Recurrent Neural Network

A type of deep learning technique is the RNN which is the descendent of NN that behaves differently from the feed-forward path. RNN has a straight path but it also contains a recursive component as shown in Fig. 14 below. This recursive component provides the RNN to build an internal hidden state (memory) that is used with the next

inputted data to learn the patterns. Essentially, the RNN uses previous observations with the new data to build features that distinguish what the model should learn and make an accurate prediction.

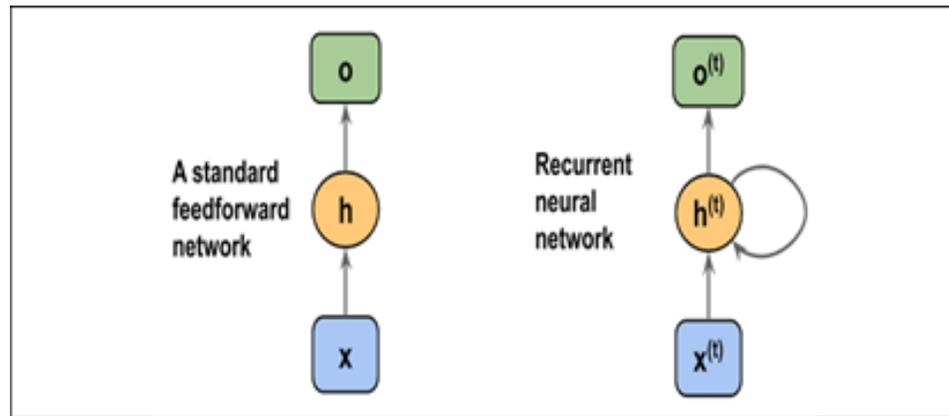


Figure 14 Feedforward vs recurrent neural network

Unfortunately, the RNN tends to suffer from short-term memory; if the sequence is long the network will have a difficult time carrying previous observations throughout the network because it encounters either vanishing or exploding gradient problems. When training an RNN, as the gradients are calculated when back propagated through the earlier layers, the gradients to update the network weights suddenly become too small (vanishing gradient, values are < 1) or become too large (exploding gradient, values are > 1) [65].

To overcome these issues two popular RNN based solutions that work well with time-series forecasting are known as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) which have been used for predicting the motion but not predicting fatigue in motion [19, 37, 38]. Since it is proven that RNNs can predict human motion, combining the motion data with fatigue analysis of motion it is possible to predict fatigue

for a specific material handling operation. A 1-Layer LSTM and 1-Layer GRU will be used for investigating two different scenarios; 1) human motion forecasting and 2) predict Borg RPE value using motion data at each frame captured.

Preparing the Sequence of Data

MoCap data and level of exhaustion increases over time as the individual performing the task will gradually experience discomfort. This will then lead to motion changing over time and the displacement of motions which can be used as an indicator of fatigue. When using time-series data, the samples are split into sub-samples taken at a timestep value creating a sequence of data. The individual features in the data appear to be in a certain order since it is sequential data dependent on the time just like the MoCap data and Borg RPE values. There are two separate scenarios in which the sequence of data plays a role. One is creating the sequence of data for forecasting the motion and the second one is creating the sequence of data for predicting RPE value. For this instance, let's assume there are n training samples for each subject of chronological order by time of each frame without shuffling the order to derive the trends within the data. Essentially using the historical sequential data, or the previous MoCap values to forecast 1) future MoCap values or 2) future Borg RPE values based on the historical data.

LSTM Architecture

The LSTM is an RNN type that contains gates that create and carry the information throughout the NNs [65]. Each of these gates represents a NN assigned to perform different tasks. The first gate is the forget gate which decides what information should be passed into the cell state or discarded. The data that passes through here is the previous hidden state and current input which in this case will be the previous. The input

gate also takes the previous hidden state and current input into a sigmoid function that decides what data is important to pass. Another factor for this is the output from the input gate is followed by a piecewise multiplication.

The output gate decides what the next hidden state should be, which contains the information on the previous input. In this case, the hidden state will contain the previous MoCap data. This information will be used with the cell state to create a long short-term memory which is ideal for predicting longer-timesteps ahead. This provides an advantage for forecasting MoCap data as this is an indicator for when a person experiences discomfort as they become fatigued.

The architecture for this model is represented in Fig 15. It will only have one LSTM layer to receive inputs, with a single dense neuron layer. This dense layer is a fully connected layer of neurons in the model that represents a matrix-vector multiplication that changes the dimension of the vector from the previous layer to a single dimension. The output size is represented by the number of neurons in the dense layer. In this case, only one neuron is needed since there is only a single forecasted output.

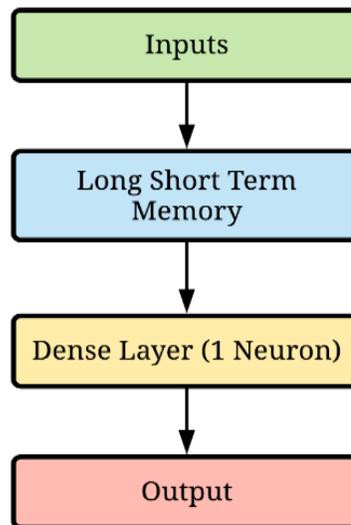


Figure 15 LSTM Architecture

GRU Architecture

The GRU contains fewer gates, making a simpler implementation of LSTM which also makes it less complex meaning it performs fewer mathematical operations [66]. This will decrease the amount of work and time compared to the LSTM. For the GRU the first gate is the Reset gate which just decides what past information to forget. Secondly, the update gate is the combination of the forget and input gate of an LSTM. They combined both these gates to give the GRU a way to decide what information is not useful to discard and what new information to add to the cell state. Just as the LSTM the hidden state is just the output of the previous time step from the MoCap data.

The architecture for this model is represented in Fig 16. It will only have one GRU layer to receive inputs, with a single dense neuron layer. This architecture is essentially identical to Fig. 14 architecture, but the LSTM is replaced with a GRU layer.

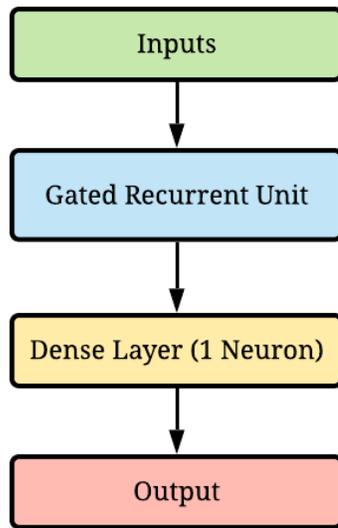


Figure 16 GRU Architecture

VI. EXPERIMENTAL DESIGN

In this chapter the review of experiment design for both the study of using the univariate approach in human motion forecasting and predicting the fatigue Borg RPE value based on the displacement of motions. For this implementation, only supervised learning will be considered for training the ML algorithm. There will be two different ways that supervised is implemented within the architecture of the ML algorithm. The first way is a univariate approach of using one single marker data and convert it into a time-series sequence in using previous observations and the next is to predict the future values. There are a total of ten subjects (seven males, three females) that are related to the lifting experiment of 51 cm height and the nine-second interval between lifts. Different model architectures using LSTM and GRU were experimented with using Tensorflow 2.2.0 and Keras 2.4.3 which provides a Python interface for deep learning neural networks. Only these two different deep neural networks for each scenario are configured and evaluated. The loss function used for training is the mean square error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (1)$$

The \hat{Y} is the predicted value, and Y represents the ground-truth value. Using MSE will indicate that this model is a regression analysis to allow to predict a continuous outcome variable.

Computation Resources

Hardware

For creating and executing the experiments the Texas State University Learning, Exploration, Analysis, and Process (LEAP) cluster and High-Performance research group's cluster were used in this research. These clusters are a group of servers that work together that increase computational capabilities. The LEAP cluster uses the SLURM batch system for submitting jobs, which execute the experiments. The system details of the LEAP cluster are listed in Table 3. The nodes are the inter-connected computers that work together to create a single powerful machine to perform highly intensive computational tasks.

Table 3 LEAP Cluster System Details [67]

| LEAP High-Performance Computing Cluster Hardware Details | |
|--|--------------------------|
| Operating System | Linux (Cent OS) |
| CPU Type | Two Intel Xeon E5-2680v4 |
| Processor Cores | 3,532 |
| Nodes | 123 |
| Memory | 18TB |
| Disk Size | 48TB |
| Memory per CPU | 4.5 GB |
| CPU Speed | 2.4 GHz |
| CPU Cores per Node | 28 cores |
| Memory per Node | 128 GBs |
| Batch System | SLURM |

The HiPE servers were also used to run experiments that simultaneously ran with the LEAP cluster. The HiPE servers were accessible by being part of the HiPE research group. These servers were equipped with graphics processing units (GPUs), unlike the LEAP cluster. The GPUs provided an advantage to training deep learning models as they provided better performance in computational time. Two HiPE servers that were used, and their specifications are listed in Table 4.

Table 4 HiPE Computation Server System Details [68]

| High-Performance Engineering Research Group Hardware Details | | |
|--|---|---|
| Servers | Two PowerEdge C4130 Rack Servers (HiPE1) | One PowerEdge R740 Rack Server (HiPE3) |
| Operating System | Linux (CentOS 7) | Linux (CentOS 7) |
| CPU Type | Dual Intel Xeon E5-2640 v4 | Dual Intel Xeon gold |
| CPU Speed | 2.4 GHz | 2.3 GHz |
| CPU Cores | 20 Cores | 18 Cores |
| Memory Size | 16GB RDIMM x8 Data Width (128GB) | 16GB RDIMM x12 Data Width (192GB) |
| Disk Size | Dual 800GB Solid State Drive uSATA | Dual 1.2TB Solid State Drive SATA |
| GPUs | Dual NVIDIA Teslas V100 | Dual NVIDIA Teslas V100 |
| Cores per GPU | 5,120 cores | 5,120 cores |
| Tensor Cores per GPU | 640 tensor cores | 640 tensor cores |
| Memory per GPU | 16GB HBM2 | 16GB HBM2 |

Software

For the development of software, Python 3.7 was used for the experimental design of the machine learning models. Python provides a range of libraries that are helpful for

reading/writing data from files and doing some preprocessing. Python is one of the popular computer programming languages used to create deep learning architectures.

Each python library and its version are displayed below in Table 5.

Table 5 Python libraries used in Thesis.

| Library Name | Description | Version |
|----------------|---|---------|
| Keras | Open-source software for deep learning that acts as an interface for running machine learning platform TensorFlow [69] | 2.4.3 |
| Matplotlib | Open-source library for creating plots in Python [70] | 3.3.0 |
| Numpy | A library that adds support for processing arrays and matrices on a large amount of data.[71] | 1.19.1 |
| Openpyxl | A library to read/write Excel files to save results [72]. | 3.0.4 |
| Pandas | Library software used for data manipulation and analysis. Using data frames and operations for manipulating tables obtained from excel files and to create time-series data [73]. | 1.1.0 |
| Scikit-learn | Library software for supporting machine learning algorithms. This library was used for scaling the raw MoCap data in-between ranges -1 and 1 [74]. | 0.23.1 |
| Tensorflow | An open-source software library for machine learning that is used as the backend for Keras. Used for the training of deep neural networks of Long-Short Term Memory and Gated Recurrent Unit. This specific library supports only the Central Processing Unit (CPU) [75]. | 2.2.0 |
| Tensorflow-gpu | Same as above but this is support for Graphics Processing Unit (GPU). Using GPU support will enable an improved model performance [75]. | 2.2.0 |

Process of Merging Motion Capture and Borg RPE Values

As discussed in Chapter 3 the datasets were obtained within the guidelines of the Snook Tables. The subjects were recruited at Texas State University and were mainly students, therefore there were no experts that had experience in manual handling materials.

Since the data is in TSV format, as shown in Fig 11, the file contained unnecessary data like the heading. Using Python 3 the TSV file was parsed to extract

only the following data, the frames, the time for each frame, and the columns for the marker XYZ data. Using the knowledge obtained from literature in what is known about MMH workers that experience a WMSD, only certain markers from the subject are selected. These markers are related to the parts in which the individual will likely experience physical discomfort or will use specific parts of the body while performing a lifting motion of the material. These markers of interest are the Back, Shoulder Back, Shoulder Tops, Knees, Shins, Thighs, and Elbows since there are cases of injury in the back, shoulders, elbows, and legs (thigh and shin). This means 14 different markers are only being considered out of the 39 as these are prone to show more of what the body is reacting to when the person will begin to experience fatigue. This creates about 42 different values (14 markers * 3 orientations) per frame since there are XYZ coordinates to each marker. Using the remaining data, the Borg RPE values can be interpolated for each frame in the MoCap file with every 6000th frame indicating the start of a new RPE value. Once all the data is merged it can be saved into a comma-separated variable (CSV) as shown in the figure below. Another thing to note is that the frame number and time per frame are also included within the file. A thing to note about time per frame since frames are captured at a rate of 100 Hz (100 frames/second), one frame is captured every 10msec, so when for example, when predicting one timestep it accounts for 1 frame ahead or 10msec ahead.

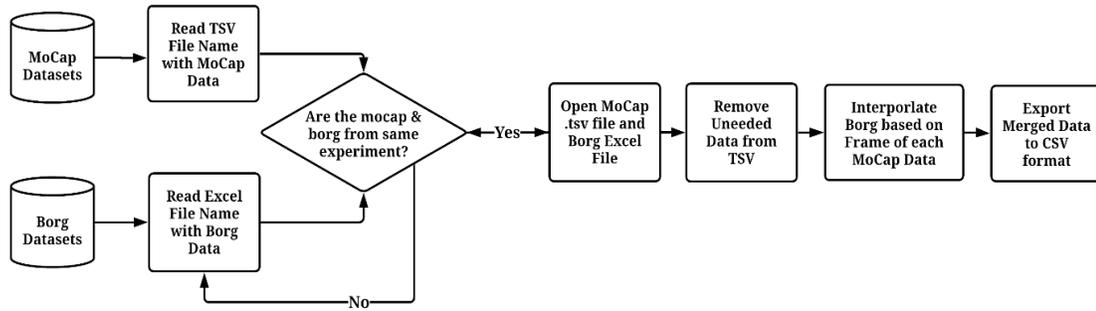


Figure 17 Flow of Merging MoCap and Borg to CSV

This CSV format is used in the machine learning process of creating each model. This will provide a smaller size file and a simpler process of opening the file using python panda's data frame for an efficient way of reading the data.

Normalization of Motion Capture Data

As explained in the section before, the MoCap data used contains XYZ coordinates for each marker captured throughout the experiment. Using only the MoCap data, as a raw format could provide an issue while training because each subjects' data will feature a different range because of their height difference, the marker placement, or even the range of motion conducted over time. The goal for normalizing the MoCap data is to rescale the features or numeric columns in the dataset to a common scale to use with the ML algorithm. For this research, the MoCap data is normalized to a range of $[-1,1]$ with the use of the **MinMaxScaler** estimator class from the scikit-learn library.

According, to the documentation [74], the transformation is calculated using equations (1) and (2) where min and max equal the boundary of the interval of the range given ($min = -1, max = 1$). The X_{min} is the smallest in a feature column, and the X_{max} is the largest value in the column.

$$X_{std} = \frac{(X - X_{min})}{X_{max} - X_{min}} \quad (2)$$

$$X_{scale} = X_{std} * (max - min) + min \quad (3)$$

The **MinMaxScaler** estimator is fit on the training data set which are the eight different subject files. Each subject file has 42 columns of data corresponding to each marker. Using the estimator each of these will be individually fitted using the given range on the training set, e.g., between negative one and one. The same estimator will be used to transform both the training and test data. Transforming the data will scale the data to be used for creating the models.

Metrics for Evaluating Machine Learning Performance

Along with the MSE loss function, other metrics that are considered to evaluate the performance are the root mean square error (RMSE) and mean absolute error functions (MAE). These metrics are typically used for regression-type models to help indicate how the model is performing. Just as MSE these will give a sense of how well the model can predict. Using these together with loss curves plot can provide a complete picture of the distribution in prediction errors [76].

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{Y}_i - Y_i)^2}{n}} \quad (4)$$

$$MAE = \frac{1}{n} \sqrt{\sum_{i=1}^n |\hat{Y}_i - Y_i|} \quad (5)$$

Training, Validation, and Testing

To design a machine learning algorithm a common practice is to split the data into two sections a training set and a testing set. The model is initially fitted on a training

dataset which provides a set of examples for the system to learn the patterns from pair of input to the known output values (target). The training dataset can optionally be split into a subset known as the validation dataset. A validation data split provides examples containing the unknown input and output pairs not used during training to provide an unbiased evaluation of the models fit on the training dataset. This is used for tuning the model hyperparameters and find the optimal parameters for the model. Comparing the training and validation error will provide a sign for the model overfitting or underfitting to the training dataset. Lastly, the test set provides a sample of unseen data to provide an unbiased evaluation of the final model. Fig. 18 shows how data can be partitioned to be used in machine learning design. For this research, the dataset of ten subjects is split into two ways, for forecasting motion, it is split into eight subjects for training with 15% of each subject data used for validation and two subjects for testing. For predicting the Borg RPE value, the data is split into eight subjects for training and two subjects for testing with no validation set since all the RPE values are needed to not miss the information provided by each subject. The MSE, RMSE, and MAE are calculated to measure the testing performance between prediction and true values in the.

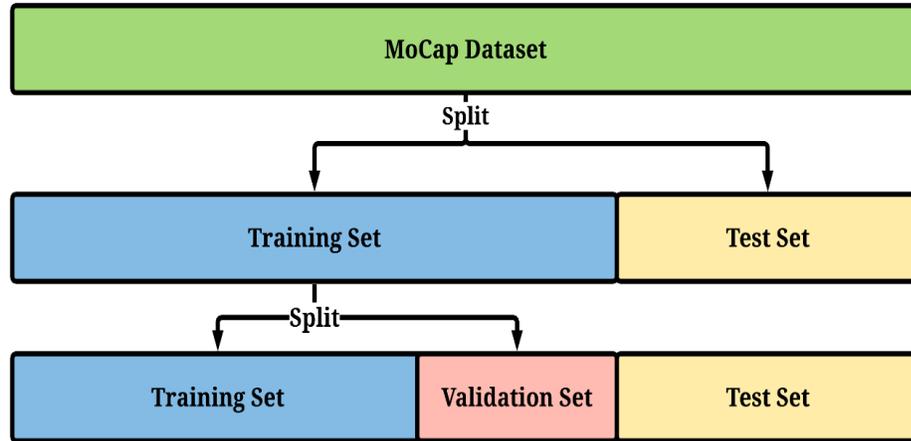


Figure 18 Ways to partition the dataset.

Loss Curve Plots

To measure the performance, besides using the metrics and picking the implementation that contains low MSE, RMSE, and MAE error between prediction and target values is using loss curve plots. A loss curve plot illustrates the training and validation losses against the number of epochs on the fitted model. The horizontal axis (x-axis) represents the number of epoch and the vertical axis (y-axis) represents the error values. This provides insight into how the ML model performance is fitted to training data and if any overfitting or underfitting is occurring. Overfitting is when a model has learned the trained data too well but fails to generalize on a new set of data that has not been seen by the model. Underfitting is when the model cannot learn the underlying patterns within the training data and fails to predict.

Univariate Human Motion Forecasting with Machine Learning

In this section, there are two different scenarios for using the Univariate approach for forecasting human motion. The first scenario begins with using a single file and single subject to help guide the training process in finding the hyperparameters for the machine

learning algorithm. The second scenario is using every single file that is presented with a lifting task at the height and interval discussed prior and evaluating on a single marker to forecast motion. Each of these scenarios will follow the flow shown below. Each of these scenarios was trained on 10 epoch and MSE loss function to keep consistency between the two algorithms. A validation split of 15 percent was used as well to perform validation of the model and indicate how the model was performing using the set of parameters.

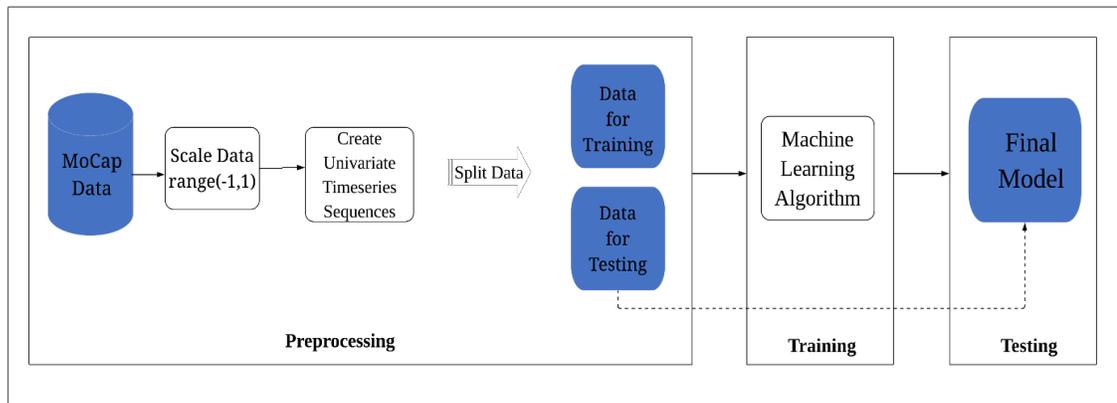


Figure 19 Univariate Approach for MoCap Motion Forecasting

As shown in Fig. 19, the data gathered from the database is scaled to a range between $[-1,1]$. Once the data is normalized to a common scale, the data must be transformed to represent a time series of sequences of consecutive observations before it can be modeled. This is a requirement for the model to learn from historical data to forecast the next value in the sequence of time-series data. LSTM and GRU will learn patterns that map a sequence of past observations as input to an output observation. Since this is a univariate time series forecasting using a single marker, the sequence of observations is transformed into multiple MoCap examples as input and a single output

prediction. For example, to make a single forecast 50 msec (five timesteps), given 50msec of history data in a sequence it will look like Fig. 20.

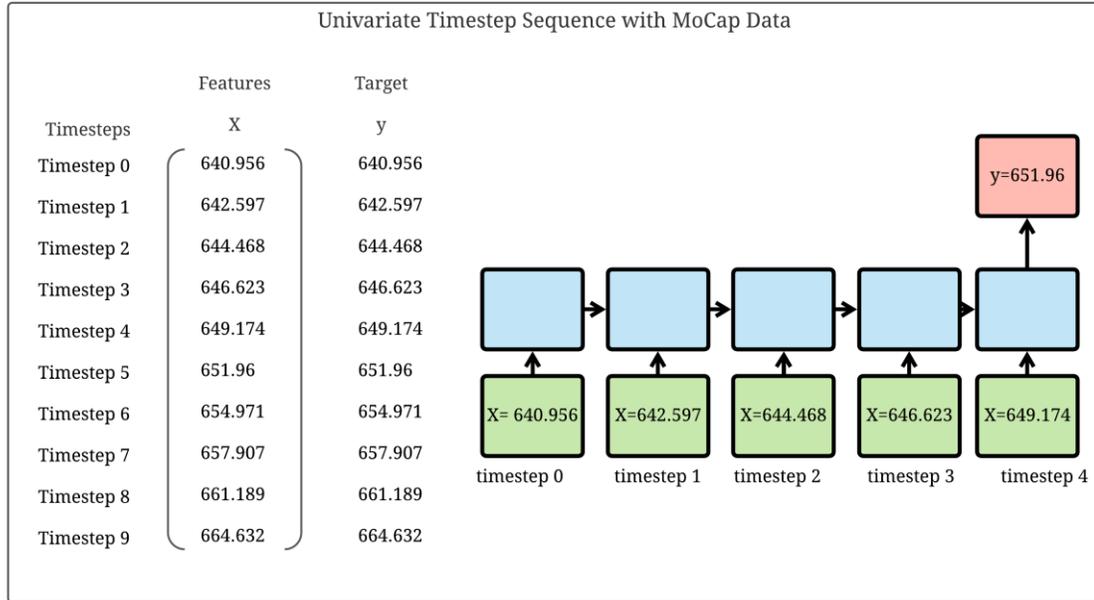


Figure 20 Preprocessing of MoCap data into univariate timestep sequence, an example of five timesteps ahead.

One File One Subject Hyperparameter Tuning

For this implementation, only one subject and one MoCap file were used for finding optimal parameters that will give insight into how the two algorithms will perform. Using hyperparameter tuning, a process used to find an optimal parameter, each of these algorithms presented different outcomes. There was a combination of the number of neurons ranging from [1,20], optimizer between Adam and stochastic gradient descent (SGD), and three different options for splitting training and testing data. One thing to note was the split difference did not present much information since the future implementation will use more subject files that will contain a different split between training and testing data. The split referred to here is splitting the single-subject data into

a certain percentage for training and the rest for testing. This is useful to indicate the number of neurons that will be ideal to use when using multiple files. Each experiment was conducted for 10 epochs which are the number iterations data are trained on with a batch size of one.

For this scenario, there was a combination of 20 neuron options * 2 optimizers * 3 split differences (60%/40%, 70%/30%, and 80%/20%) making a total of 120 different experiments to find optimal parameters. These experiments were performed on a singular marker for each marker, making it a univariate approach to forecasting. As discussed in the previous section, each of these marker data was converted into a time series sequence. The data was sequenced into predicting 10 msec (one timestep). Using only the MoCap data from the marker, for example, RElbowOut X position, the sequence could be constructed. Using t as a timestep factor, for instance, the input data (X) at $t=0$ will be the first value at frame 1 while the target value (Y) will be $t+1$. This process will be followed by scaling the data to then splitting it into training and testing. These models were trained on the LEAP cluster. The learning rate for Adam and SGD were kept in their default setting. The learning rate determines the step size in each iteration that will move the loss error to its minimum. A summary of the parameter values for both models is displayed in Table 6.

Table 6 LSTM & GRU parameter values.

| Parameters | LSTM | GRU |
|--------------------------|-----------------------------|-----------------------------|
| Batch Size | 1 | 1 |
| Loss function | mean square error | mean square error |
| Optimizer | Stochastic Gradient Descent | Stochastic Gradient Descent |
| Learning Rate | 0.01 (default) | 0.01 (default) |
| Number of Neurons | 20 | 17 |
| Number of Epochs | 10 | 10 |

For this scenario, both LSTM and GRU SGD were found to perform the best compared to using Adam optimizer. SGD was a more effective optimizer for the model to learn the patterns when using a single marker and its corresponding values. Based on the metrics for the LSTM 20 neurons seemed to capture the most benefit for optimizing the error between prediction and ground truth while for GRU it was 17 neurons.

Training and Evaluating with Multiple Subjects

Using the experiments from beforehand this scenario uses all the subject data available for lifting height of 51 cm and 9-second interval between lifts. For this scenario, the data is split between the ten subjects with the assumption that the participants will have similar movement patterns. Eight for training (6 male, 2 female) and two subjects for testing (1 male, 1 female). A male and female were placed in the testing set to represent the sample of data that will provide an evaluation on forecasting male and female subjects' motion. These models are trained on both the LEAP cluster and the High-Performance Engineering (HiPE) servers. Using the parameters discovered in the past experiments with a single subject, the LSTM uses 20 neurons and GRU uses 17

neurons for training and testing on all 10 subjects. For each experiment, the way each file was used to train depended on the number of epochs. As done in the previous scenario ten epochs were used for training and a validation split of 15% was also used for measuring the performance of using multiple subject files instead of one subject. To measure the performance of the models only one marker will be considered which is the RElbowOut X position. Using every single file, a loop is created to train individually each file between each epoch. At the beginning of every epoch, the file order is randomized with each epoch having a different subject order. This process also follows the same approach in Fig 19. Each subject data is fitted into the scale data to be used for future scaling of unseen test data. Different motion forecasting experiments were created to give a comparison between the forecasting length of sequence timesteps and the difference in batch size. Each training dataset will be transformed into a time-series sequence based on previous timesteps to predict future values as done in the example in Fig. 20.

- For these experiments, the goal is to make a single prediction:
 1. 1 Timestep (10 msec) given 10 msec of previous observations.
 2. 5 timesteps (50 msec) given 50 msec of previous observations.
 3. 10 timesteps (100 msec) given 100 msec of previous observations.
 4. 20 timesteps ahead (200 msec) given 200 msec of previous observations.
 5. 100 timesteps ahead (1 sec) given 1 sec of previous observations.

The timesteps considered in this scenario are predicting 1, 5, 10, 20, and 100 frames with a batch size of one and a batch size of ten. The purpose is to investigate the performance of each algorithm using different timestep sizes for forecasting t frames ahead and measure if there is a difference in performance when increasing batch size to 10. The reason for increasing batch size will be to decrease the computational time without losing the performance as a batch size of 1 only take a sample one at of time. With a batch size of 10, 10 samples from the training set will be used to calculate the gradient error value by comparing the prediction of those 10 samples to their expected output variables. Afterward, the internal weights and bias are updated in the algorithm, unlike a batch size of 1 where they are updated after each sample presented.

Once the training process is completed the remaining two datasets are used to test the model's performance. These two subjects as discussed previously are 1 male and 1 female subject. Just like the training dataset, each test set is scaled using the same scaler from training. This will normalize the data between $[-1,1]$ and will be used as the new input to predict based on the timestep provided. The data must also be formatted into a sequence of the previous timestep to forecast the value. Using the metrics MSE, RMSE, and MAE, each of these can indicate which experiment performed the best in LSTM and GRU.

Multivariate Human Motion to Predict Borg RPE Value

Using the experiments from beforehand this scenario uses all the subject data available for lifting height of 51 cm and 9-second interval between lifts which are in total ten subject files. The data is split between ten subjects. Using the ten subjects the assumption is participants will have similar fatigue based on the motion displacement

from lifting. Eight subjects are used for training (6 male, 2 female) and two subjects for testing (1 male, 1 female). The same process is conducted for this scenario as shown below Fig. 21 except for this time the target values are not MoCap values, the target values are Borg values.

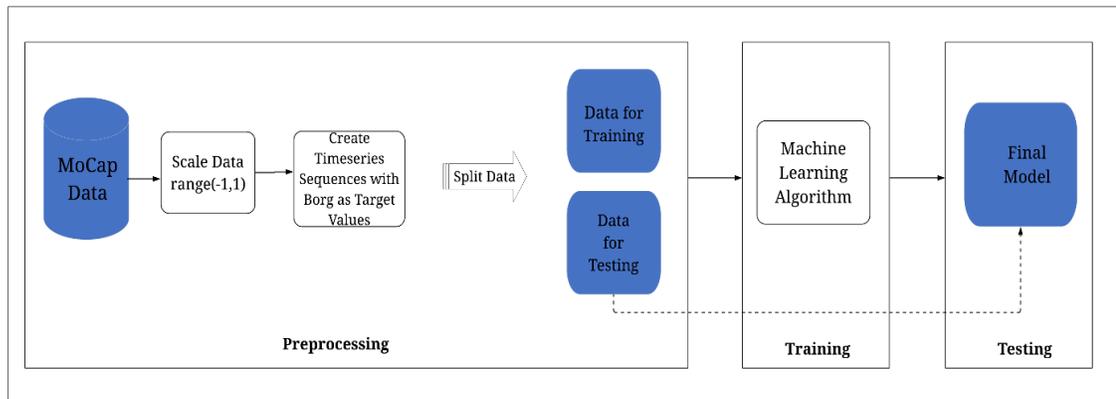


Figure 21 Multivariate MoCap Data to Predict Borg Values

The markers of interest for this case are the Back, Shoulder Back, Shoulder Tops, Knees, Shins, Thighs, and Elbows since there are cases of injury in the back, shoulders, elbows, and legs (thigh and shin). This means 14 different markers are only being considered to show more of what the body is reacting to when the person will begin to experience fatigue from lifting [4, 9, 46]. There are 42 different columns with values (14 markers * 3 dimensions) since there are XYZ coordinates to each marker. The time per frame was also included in the set of features that will be inputted into the computer algorithm. This will create a dimension of 43 different features or columns by n samples. Each row will contain the time per frame and MoCap data. A male and female were placed in the testing set to represent the sample of data that will provide an evaluation on male and female subjects. These models are trained on both the LEAP cluster and the

High-Performance Engineering (HiPE) servers. Using the parameters discovered in the past experiments, indicated by Table 16 the LSTM and GRU will remain the same using eight subjects for training and testing using two subjects, with an epoch of 10 and batch size 1. For each sequence, there will be 43 different inputs that will be used to predict the target values which is the Borg value at that frame in time.

Creating the sequence of time is essentially the same as in Fig. 20 except there are multiple features of input data and the single output data is the Borg RPE value instead of MoCap data. In this case, a single prediction 10msec into the future given 10msec of historical data which is 43 different points in the sequence transformed to be like in Fig. 22.

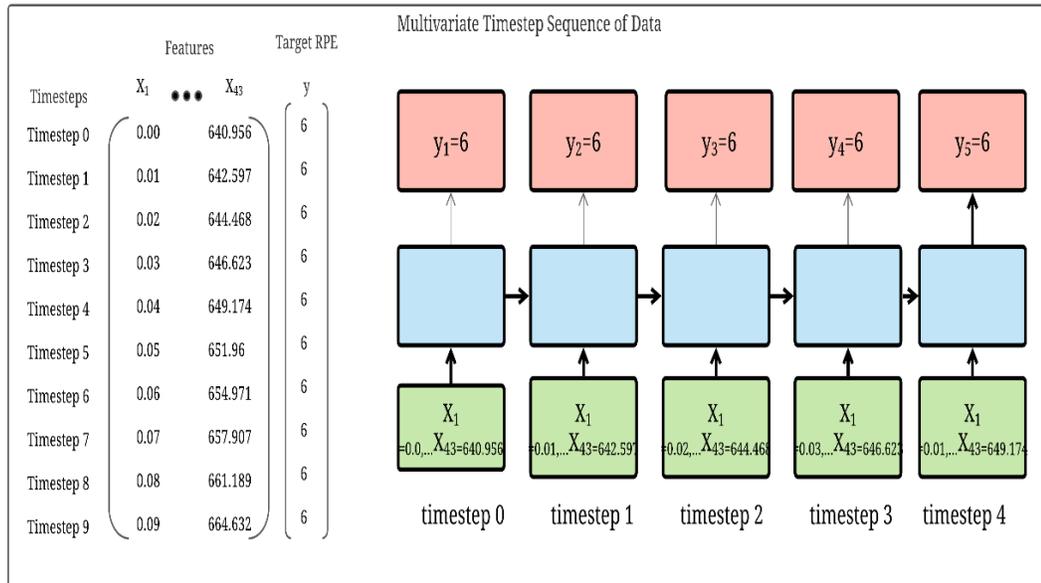


Figure 22 Multivariate data preparation to create a sequence of data given 10 msec history to predict 10 msec Borg RPE value.

Since the Borg value is in increasing order with dependent on a time in the activity, all the samples must be used as to not remove information needed to teach the

ML algorithm. Therefore, no validation split in data will be evaluated as described in the past experiment designs. In this case, test sets are going to be used to evaluate the performance of the trained model. This will be used to investigate if motion displacement is enough to give an indication of fatigue and be able to predict it.

VII. RESULTS AND DISCUSSION

In this chapter the experimental results for both the univariate human motion forecasting with machine learning using multiple subjects and predicting Borg RPE Values using multivariate motion capture data. Each evaluation or test is performed on the test set of data that contains only two subjects. subject 1016 a male subject and 007 a female subject. Both participants provide a sample of RPE data that ends in the same number of 19 which is close to the maximal exertion.

Univariate Analysis of Human Forecasting

This section first covers the performance on the test set using metrics of MSE, RMSE, and MAE. For the first model, the LSTM will be shown using the parameters of 20 neurons, an MSE loss function. For the second model, the GRU will be shown using 17 neurons with the same loss function. The loss function estimates how close the predictions made by the model match the target variable that was shown during training.

In this scenario, as described in chapter VI, a univariate approach with multiple subject data is considered with the use of one marker data. In this case, only the RElbowOut X data is used to create a time series sequence in which the input data is the previous RElbowOut X data, and the target value is the next measurement in the data. Since the model was trained on eight subjects using ten epochs, in which each subject is individually processed through the ML algorithm so each file shall be iterated ten times by the end of the tenth epoch.

To measure the performance, ten different sets of experiments are designed to find the optimal way of training the model and forecast the timestep t for RElbowOut X. Firstly, using a batch size of one, five different timestep predictions will be used to train

the model and then evaluated using the test set of data. The five different timestep predictions are $t+1$, $t+5$, $t+10$, $t+20$ and $t+100$. To reiterate this means a prediction of 10 msec, 50 msec, 100 msec ahead, 200 msec, and 1 sec. Additionally, using the same time step prediction, training the model with a batch size of ten is also evaluated and compared with the batch size of 1. Each setup is executed ten times and the metrics for each are calculated at the end of each experiment. Once all ten experiments were completed the average of all the ten runs was taken to measure the performance of the model. The difference between each run was the training process in which each training file is randomly selected. This will create a sense of how the model will perform using differently ordered data.

To measure the performance, besides using the metrics and picking the implementation that contains low error between prediction and target values there are two other ways to measure performance. The first way is to view the loss curves and metrics. The second way is to view the actual forecasted MoCap data plotting the raw data and forecasted values over the number of frames in the implementation. These plots contain a vertical (x-axis) that corresponds to the MoCap position values, and the horizontal (y-axis) corresponds to the frame number. For these plots, each data point is plotted for every 1000th frame to reduce the number of points and provide a better visualization between raw data and forecasted values. Only four sets of the experiments for each model will be showcased. Two will represent the worst-performing from each batch size and the other two will represent the best performing from each batch size. For the MoCap prediction plots, blue arrows will be used to show the spots where there are gaps and inconsistencies between the raw data and forecasted values. This will be used to

showcase each model’s performance between batch sizes and how well the fitted model predicted on the test set. Starting with LSTM the plots will be shown followed by a table of metrics from all experiments. Secondly, the GRU plots will be shown and also a table of the metrics from all experiments.

LSTM Worst Performance Loss Curve Plots: Batch Size 1 & 10

In Fig 23 and 24, the loss curves of two experiments were plotted for the LSTM implementation using batch sizes of 1 and 10. Fig. 23 is associated with forecasting 200 msec given 200 msec of previous data, using a batch size of 1. Fig 24, is associated with forecasting 50 msec given 50 msec of previous data, using a batch size of 10.

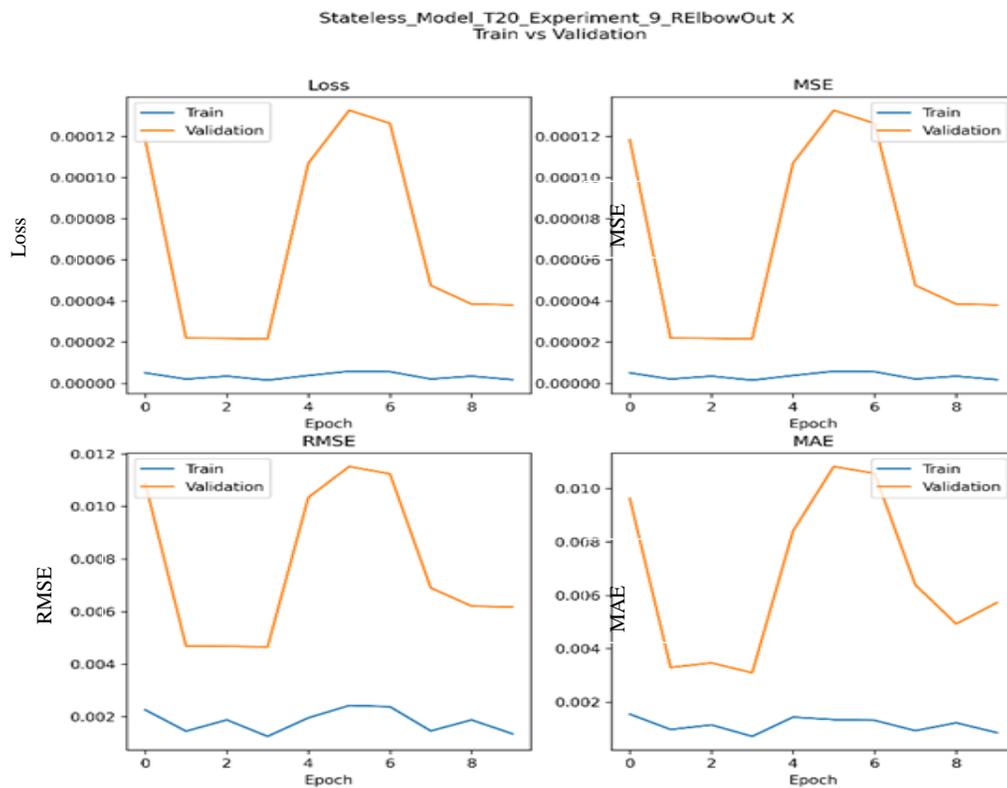


Figure 23 LSTM 200 msec train and validation loss curves of using batch size 1.

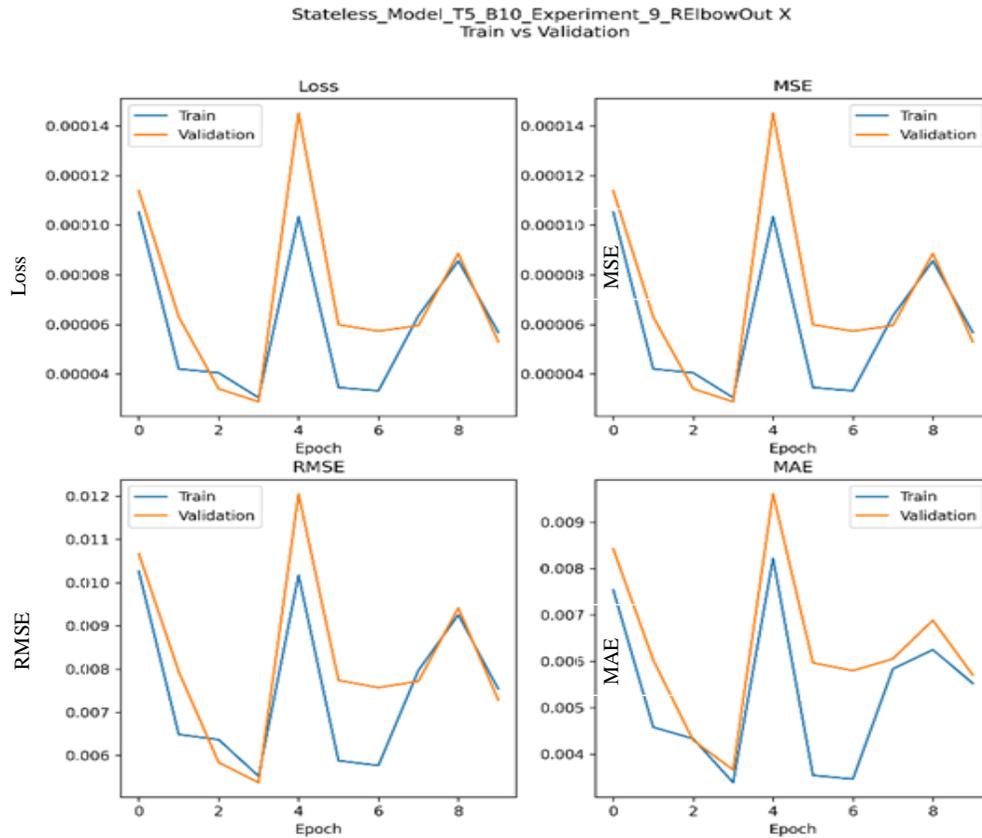


Figure 24 LSTM 50 msec train and validation loss curves of data using batch Size 10.

LSTM Worst Performance MoCap Forecasting with Batch Size of 1

The two plots that are shown in this section are for each subject in the test dataset using a batch size of 1. These plots in Fig. 25, and 26 are the experiment of forecasting 200 msec that also correspond to loss curves in Fig. 23. These were plotted to show the performance of the fitted model for forecasting MoCap data on the 007 subject and 1016 subject. The first figure shows the predictions for 007 and the second figure shows the predictions for 1016. The blue arrows will indicate where the fitted model underperformed.

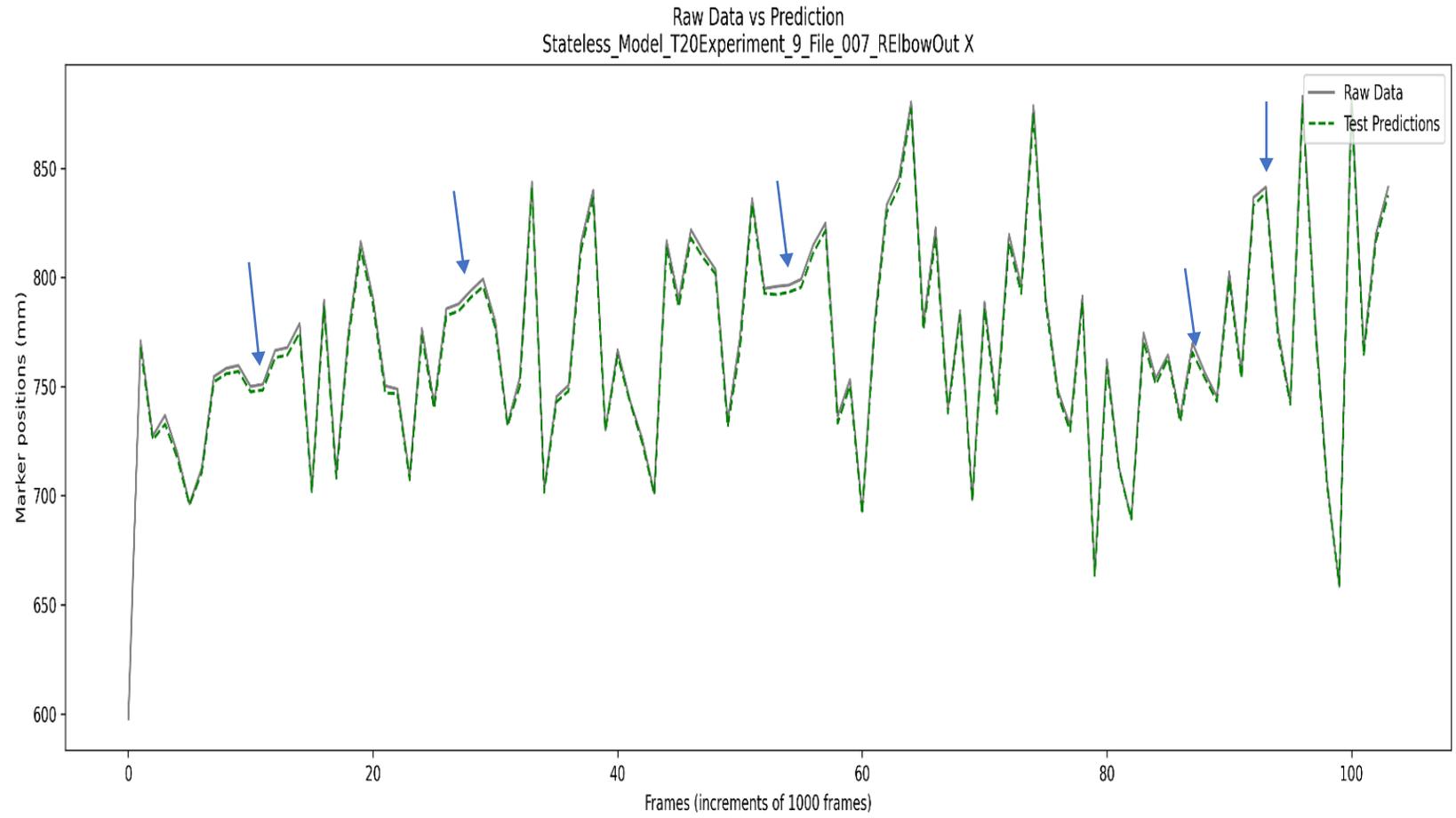


Figure 25 LSTM motion forecasting 200 msec of 007 with batch size 1

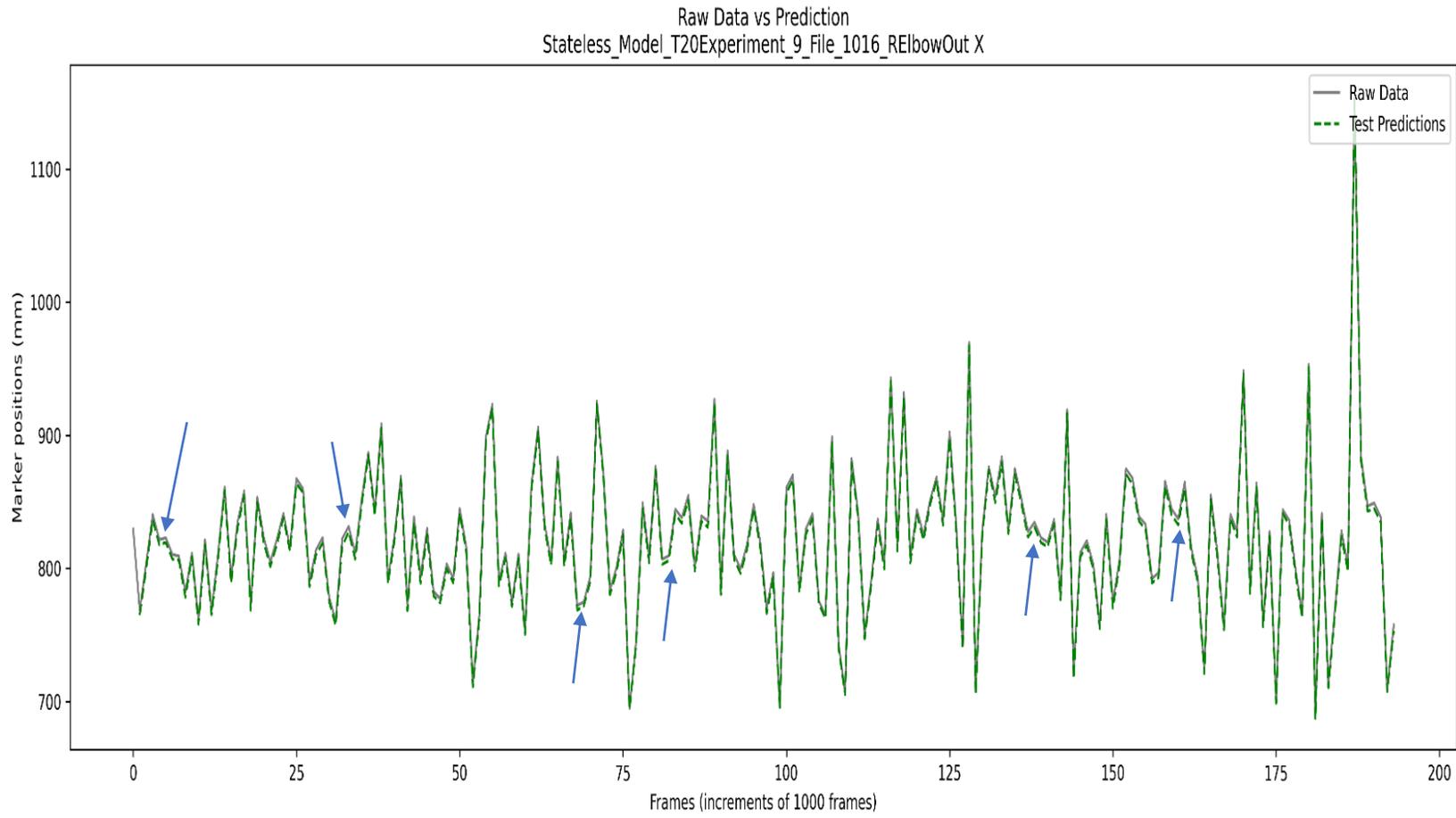


Figure 26 LSTM motion forecasting 200 msec of 1016 with batch size 1

As can be seen in Fig 23 the validation curve compared to the training, there are gaps. This gap indicates that there is some overfitting occurring within the data so there will be a possibility that new unseen data will not fit well the trained model. Looking at Fig 25 and 26, there are spots in which the predicted data is not fitted well with the raw test data. For each subject 1016 and 007, there are variations in errors between prediction and errors that are noticeable with underpredicting the data.

LSTM Worst Performance MoCap Forecasting with Batch Size of 10

The two plots are shown in this section with the worst performance in forecasting the test dataset using a batch size of 10. These prediction plots in Fig. 27, and 28 are for forecasting 50 msec of MoCap data. These were plotted to show how well the fitted model can forecast MoCap data using 007 subject and 1016 subject when using a batch size of 10 instead of 1. The first figure shows the predictions for 007 and the second figure shows the predictions for 1016.

75

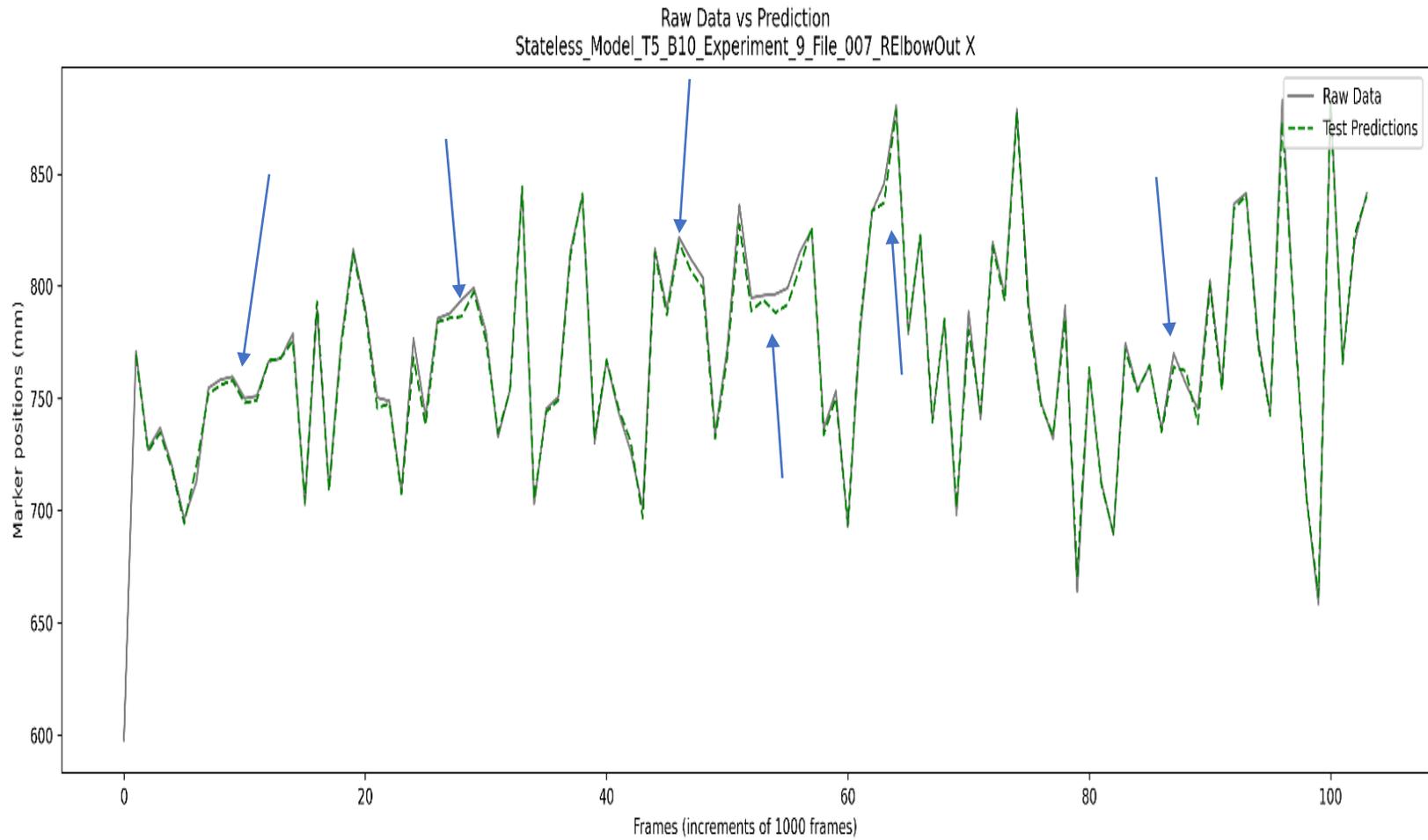


Figure 27 LSTM motion forecasting 50 msec of 007 with batch size 10

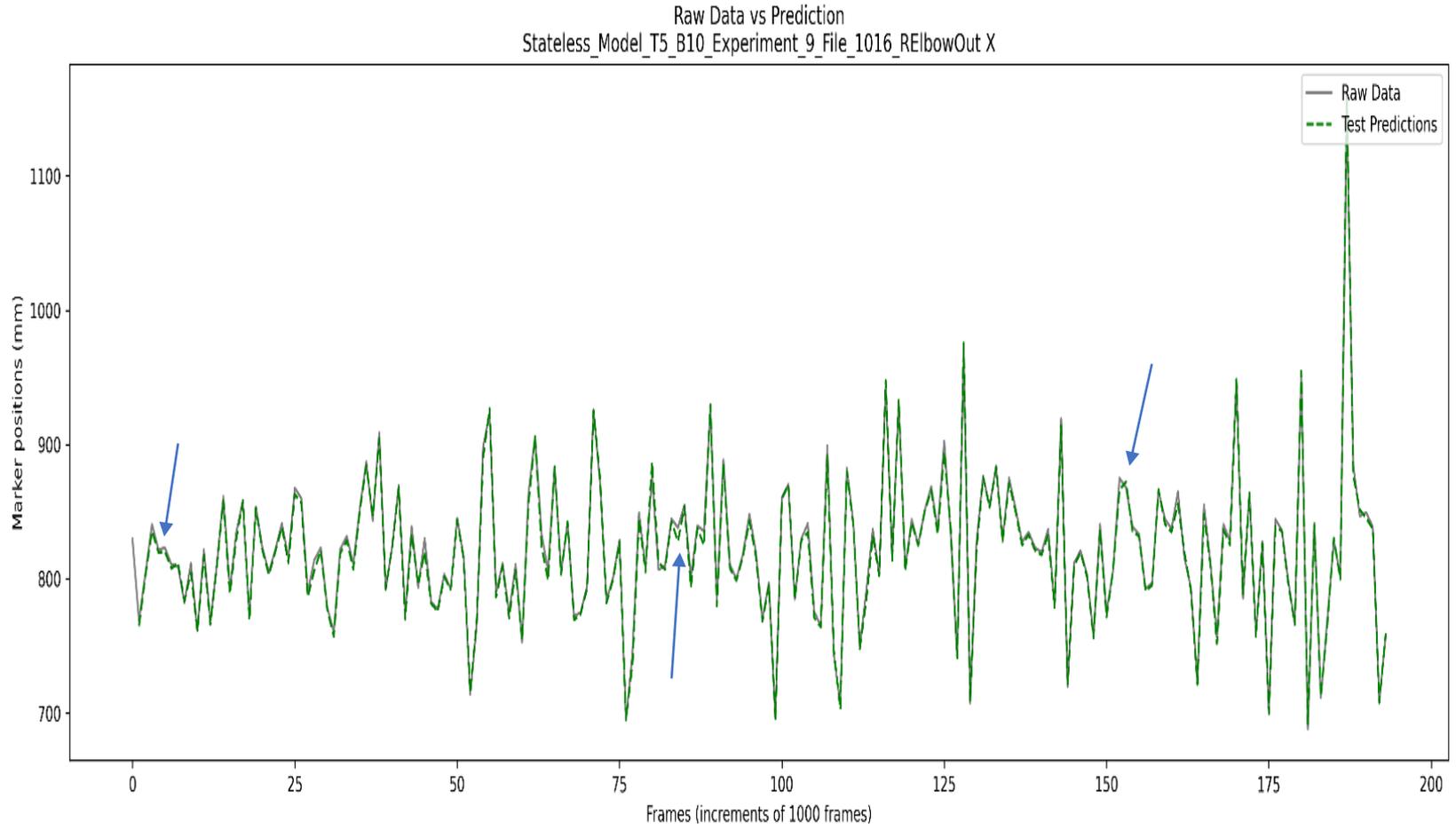


Figure 28 LSTM motion forecasting 50 msec of 1016 with batch size 10

Fig 24., indicates some issues between the training and validation. They follow each other but also experience a high value of error between epoch 3 and 6 with epoch 6 having a gap in between both which means there is some overfitting. Just as the previous figures that used a batch size of 1, Fig 27 and 28, have a variety of missed predictions indicated by the blue arrows. For subject 007 the data does not fit as well as for subject 1016. Since there is fewer female in training a 50msec timestep could be causing the fewer patterns to form for the ML algorithm to learn from.

LSTM Best Performance Loss Curve Plots: Batch Size 1 & 10

The loss curves of the two best-performed experiments were plotted for the LSTM implementation using batch sizes of 1 and 10. Fig. 29 is associated with forecasting 50 msec given 50 msec of previous data, using a batch size of 1. Fig 30, is associated with forecasting 100 msec given 100 msec of previous data, using a batch size of 10.

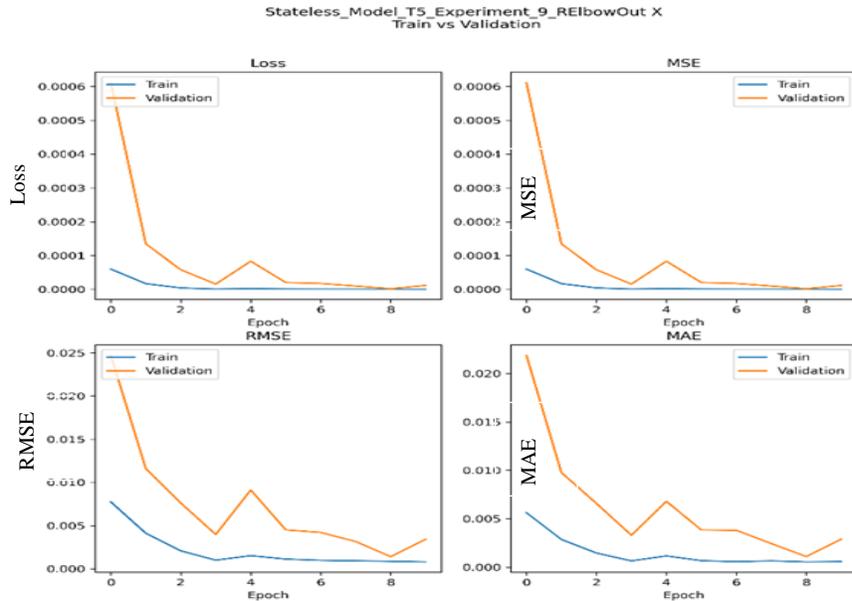


Figure 29 LSTM 50 msec train and validation loss curves using batch size 1.

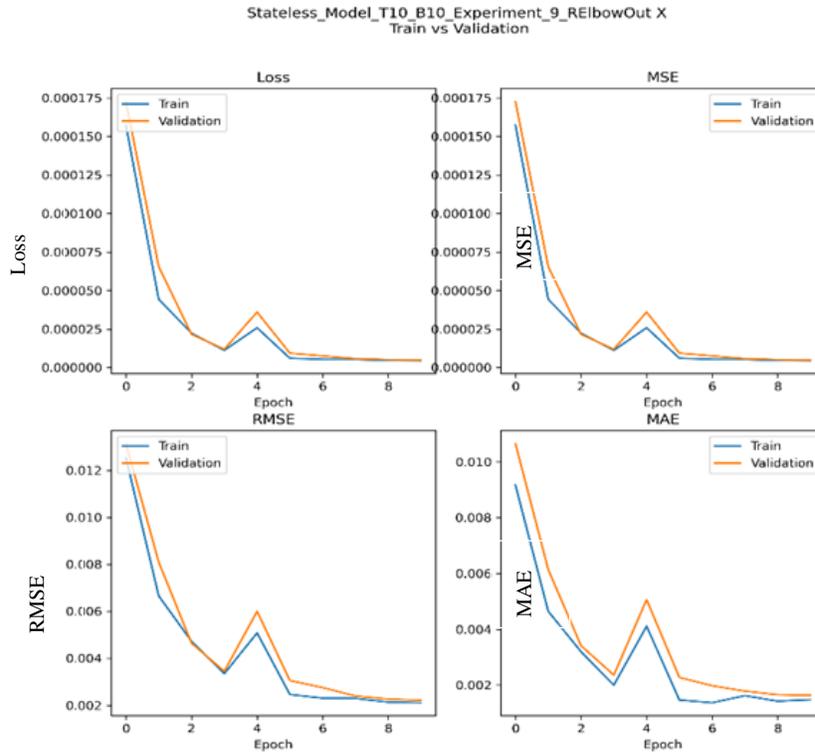


Figure 30 LSTM 100 msec train and validation loss curves using batch size 10.

LSTM Best Performance MoCap Forecasting with Batch Size of 1

In this section, the raw data against the predictive motion data of test data is plotted. These plots show the fitted model performance in predicting 50 msec, using LSTM with a batch size of 1. This is the best case in which the model predicted the best among all the experiments. Fig. 31 is the plot for subject 007 and Fig 32 is the plot for subject 1016.

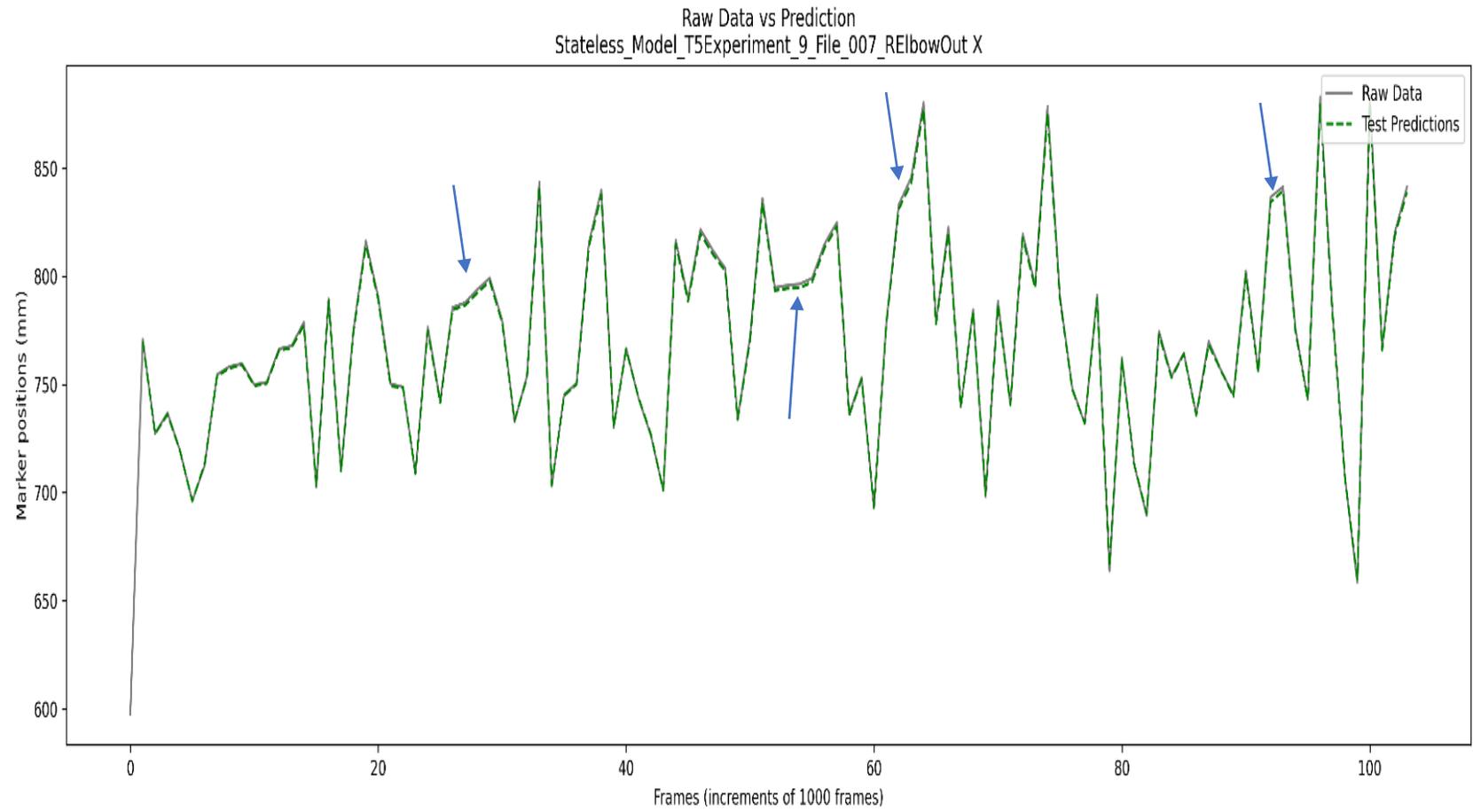


Figure 31 LSTM motion forecasting 50 msec of 007 with batch size 1

08

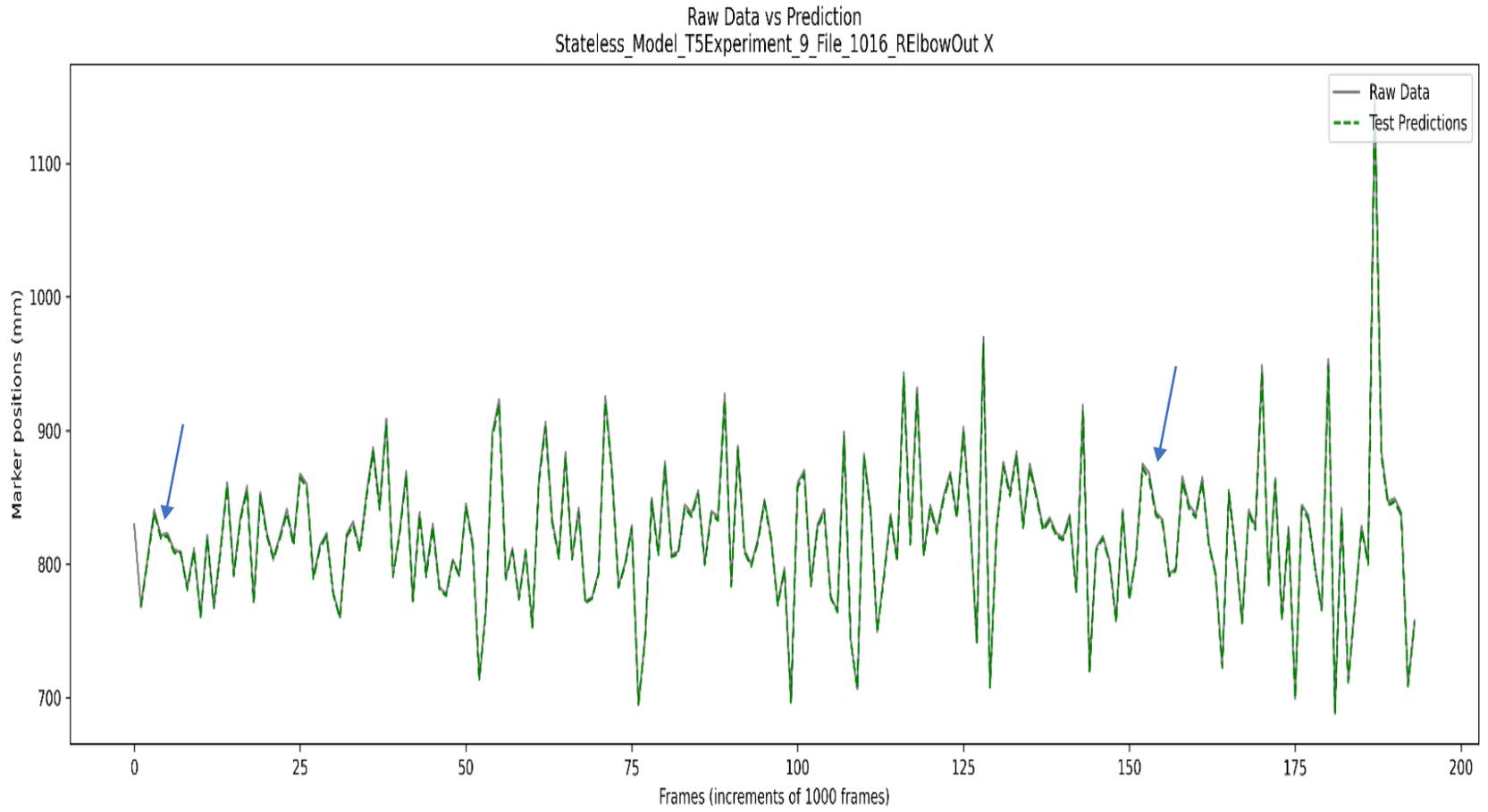


Figure 32 LSTM motion forecasting 50 msec of 1016 with batch size 1

As seen in Fig. 29, the training vs validation loss curve plots indicates that the model is fitting well with unseen data. They both converge to zero at the end of epochs if looking at MSE but with RMSE and MAE, the error was about to increase and there might have been some overfit occurring. For this reason, there is a small prediction error in the predictions of each subject. Fig. 31 and 32 show the visual plot of each subject when plotting the MoCap prediction. There are not as many wrong predictions visible to the naked eye for both subjects. subject 007 performance is better this time around than using 50msec timestep prediction.

LSTM Best Performance MoCap Forecasting with Batch Size of 10

In this section, the raw data against the predictive motion data of test data is plotted. These plots show the fitted model performance in predicting 100 msec, using LSTM with a batch size of 10. Fig. 33 is the plot for subject 007 and Fig. 34 is the plot for subject 1016.

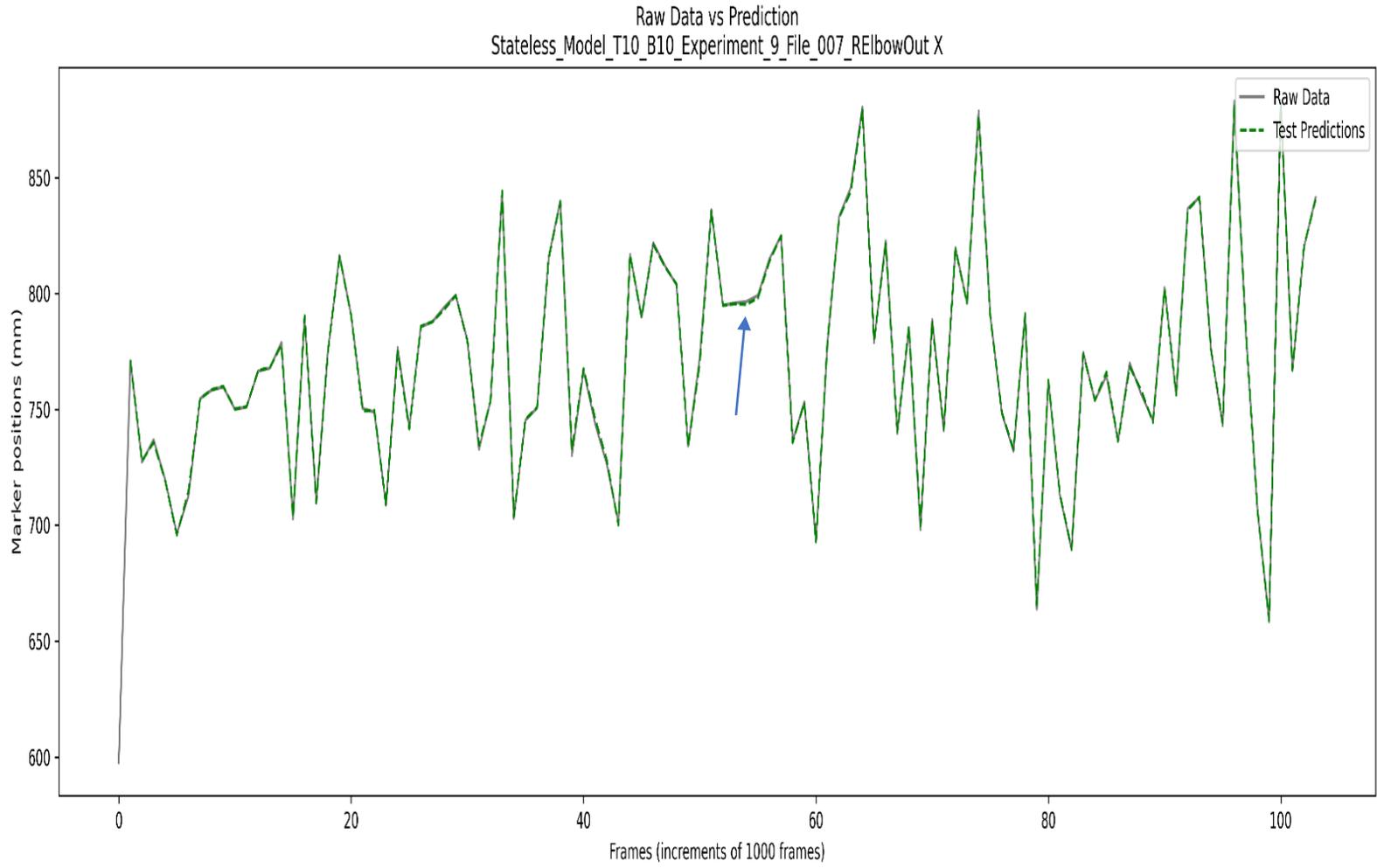


Figure 33 LSTM motion forecasting 100 msec of 007 with batch size 10

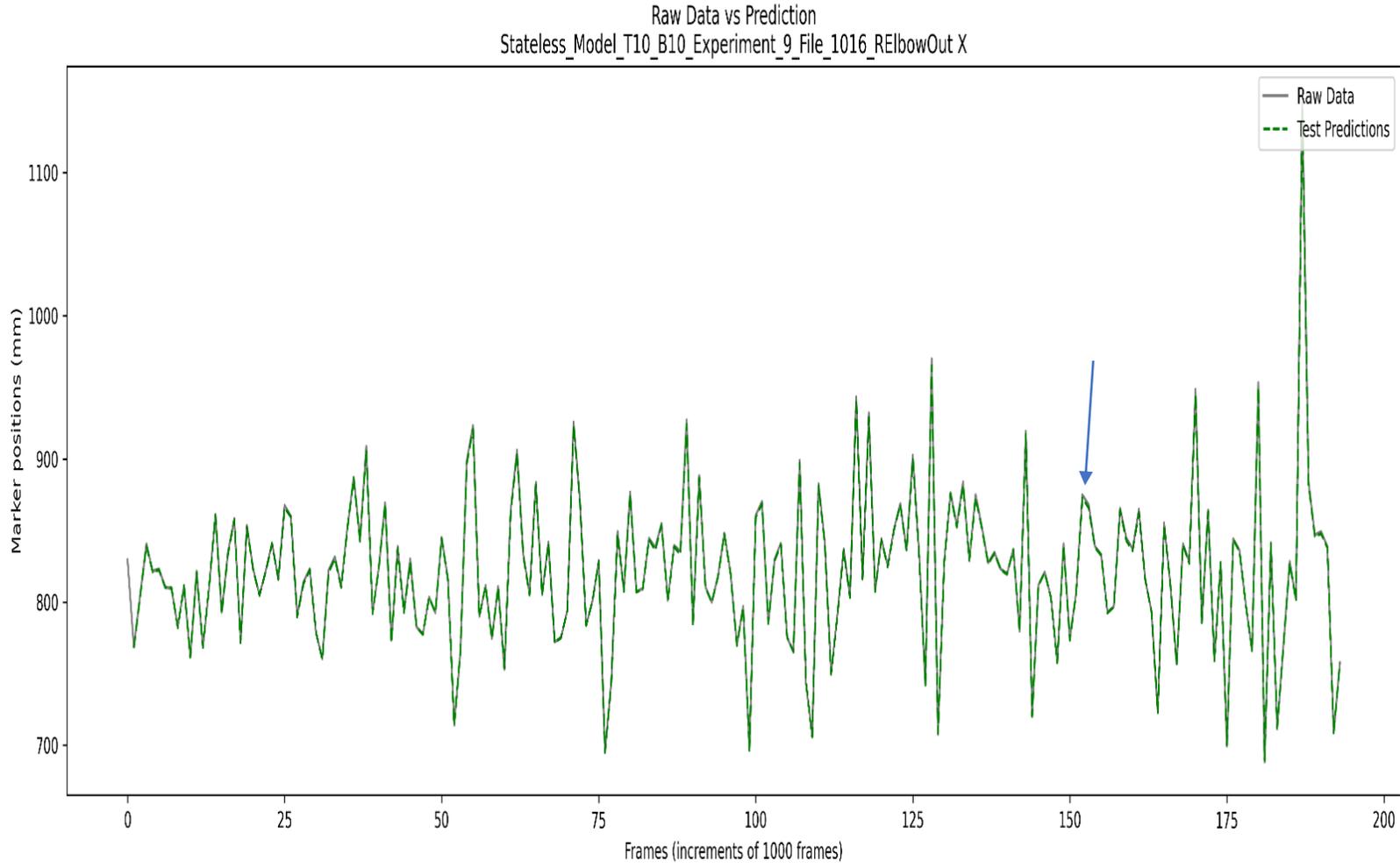


Figure 34 LSTM motion forecasting 100 msec of 1016 with batch size 10

In Fig. 30, the training and validation follow closely throughout the ten epochs, which proves that the model trained will fit well with unseen data. This means using test data that was not used in training the fitted model will perform well since the loss for training and validation converge close to zero error. Fig. 33 and 34 plots have fewer errors compared to the others which presents a case that this set of implementations will prove to work well with training the model.

LSTM Univariate Test Metrics Summary Tables

A full summary of each test conducted using the LSTM 1 layer model for univariate forecasting of MoCap RElbowOut X data is compiled into two tables below. These two tables correspond to the two different subjects used in the testing set: 007 and 1016. The best performing implementation for LSTM is using a batch size of 10 with a timestep prediction of 100msec.

Table 7 LSTM Univariate Forecasting Metrics for Subject 007

| Subject 007 Female Test Metrics LSTM | | | | | | | | | | |
|---|----------------------------------|-------|-----------------------------------|-------|-------------------------------------|------|-------------------------------------|------|--------------------------|------|
| | 10 milliseconds (1 Frame) | | 50 milliseconds (5 Frames) | | 100 milliseconds (10 frames) | | 200 milliseconds (20 frames) | | 1sec (100 frames) | |
| Batch Size | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 |
| Mean Square Error (MSE) | 58.93 | 15.37 | 2.45 | 21.86 | 17.21 | 1.11 | 26.92 | 6.10 | 100.38 | 4.97 |
| Root Mean Square Error (RMSE) | 6.60 | 3.53 | 1.53 | 4.41 | 3.81 | 1.05 | 4.53 | 2.25 | 8.09 | 2.06 |
| Mean Absolute Error (MAE) | 5.84 | 2.63 | 1.29 | 3.54 | 3.32 | 0.81 | 4.06 | 1.93 | 7.14 | 1.74 |

Table 8 LSTM Univariate Forecasting Metrics for Subject 1016

| Subject 1016 Male Test Metrics LSTM | | | | | | | | | | |
|--|----------------------------------|------|-----------------------------------|-------|-------------------------------------|------|-------------------------------------|------|--------------------------|------|
| Batch Size | 10 milliseconds (1 Frame) | | 50 milliseconds (5 Frames) | | 100 milliseconds (10 frames) | | 200 milliseconds (20 frames) | | 1sec (100 frames) | |
| | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 |
| Mean Square Error (MSE) | 38.16 | 7.77 | 2.83 | 14.36 | 11.11 | 1.19 | 17.71 | 3.68 | 97.72 | 2.95 |
| Root Mean Square Error (RMSE) | 5.53 | 2.72 | 1.58 | 3.69 | 3.15 | 1.07 | 3.80 | 1.80 | 7.99 | 1.63 |
| Mean Absolute Error (MAE) | 4.94 | 1.88 | 1.34 | 2.91 | 2.78 | 0.82 | 3.42 | 1.53 | 7.26 | 1.36 |

Looking at the metrics for both subjects, there is a solid indication that training with a batch size of 10 proves to increase the overall performance of the LSTM model as can also be seen by low error predictions and how well the validation fits well with training data in Fig 30.

GRU Worst Performance Loss Curve Plots: Batch Size 1 & 10

The loss curves for the worst performed experiments were plotted for the GRU implementation using batch sizes of 1 and 10. Fig. 35 is associated with forecasting 10 msec given 10 msec of previous data, using a batch size of 1. Fig 36, is associated with forecasting 50 msec given 50 msec of previous data, using a batch size of 10.

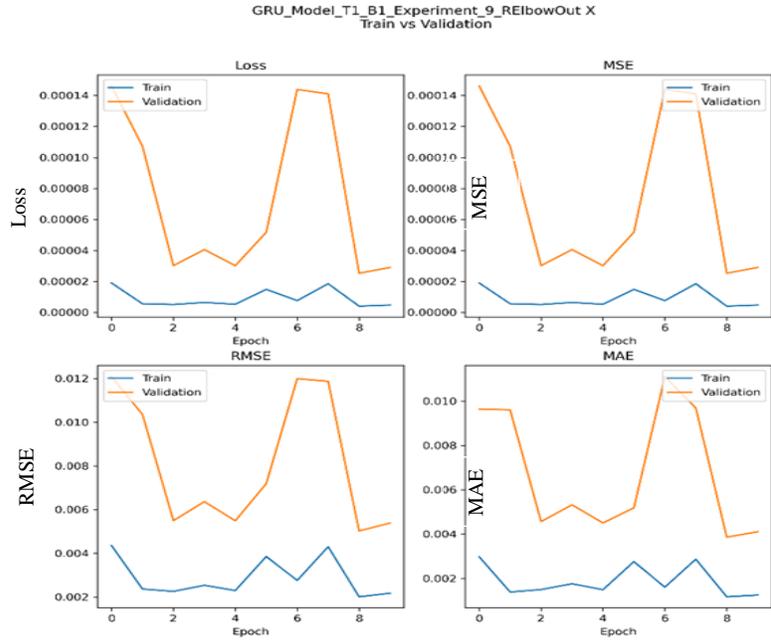


Figure 35 GRU 10 msec train and validation loss curves of using batch size 1.

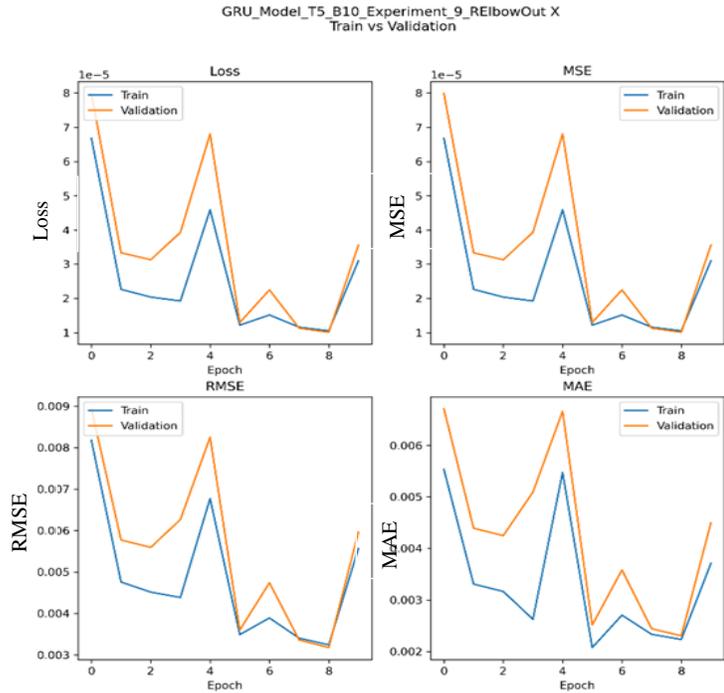


Figure 36 GRU 50 msec train and validation loss curves of using batch size 10.

GRU Worst Performance MoCap Forecasting with Batch Size of 1

In this section, the raw data against the predictive motion data of test data is plotted which performed the worst in all the experiments. These plots show the fitted model performance in predicting 10 msec, using GRU with a batch size of 1. Fig 37 is the plot for subject 007 and Fig. 38 is the plot for subject 1016.

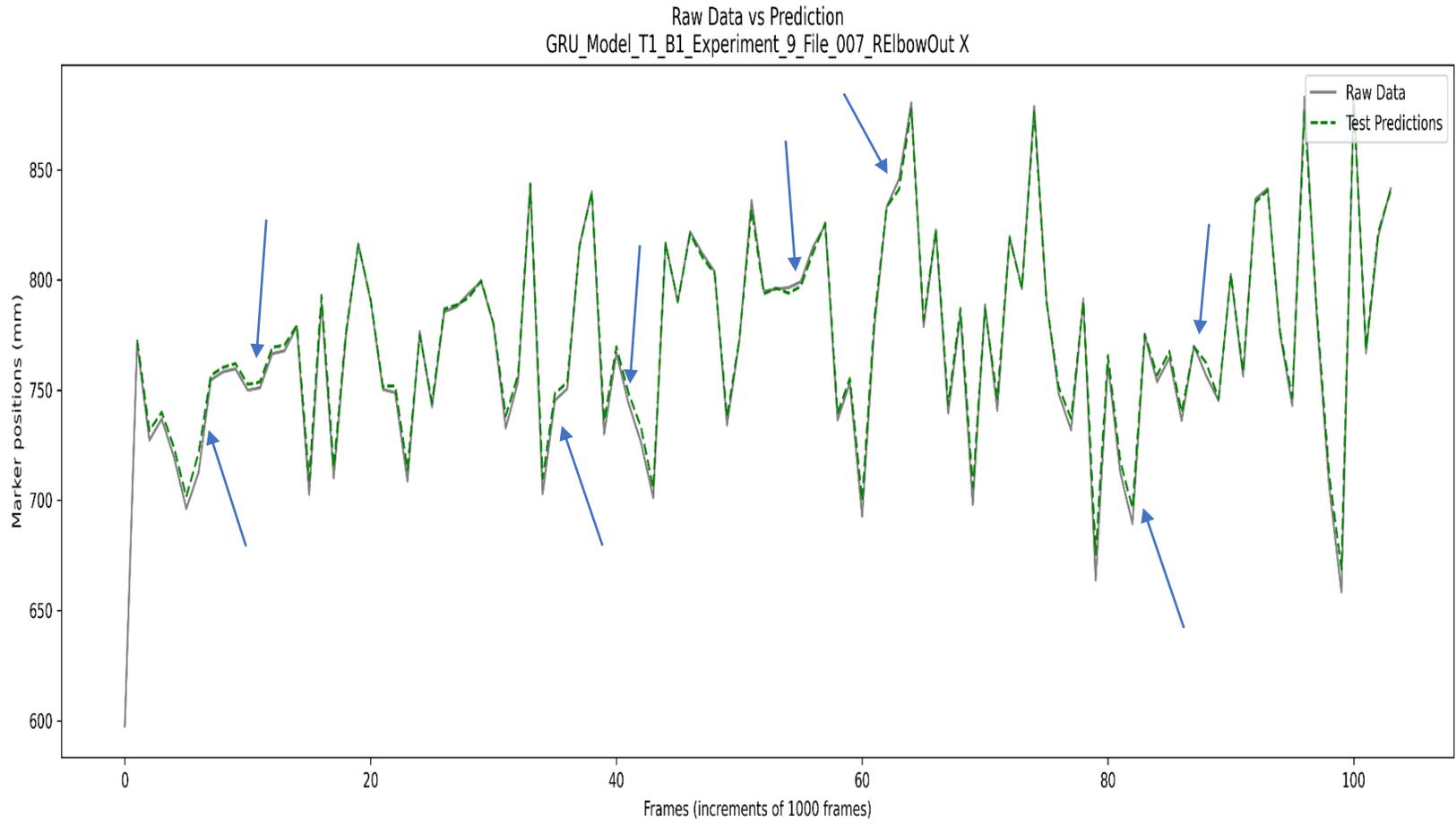


Figure 37 GRU motion forecasting 10 msec of 007 with batch size 1

68

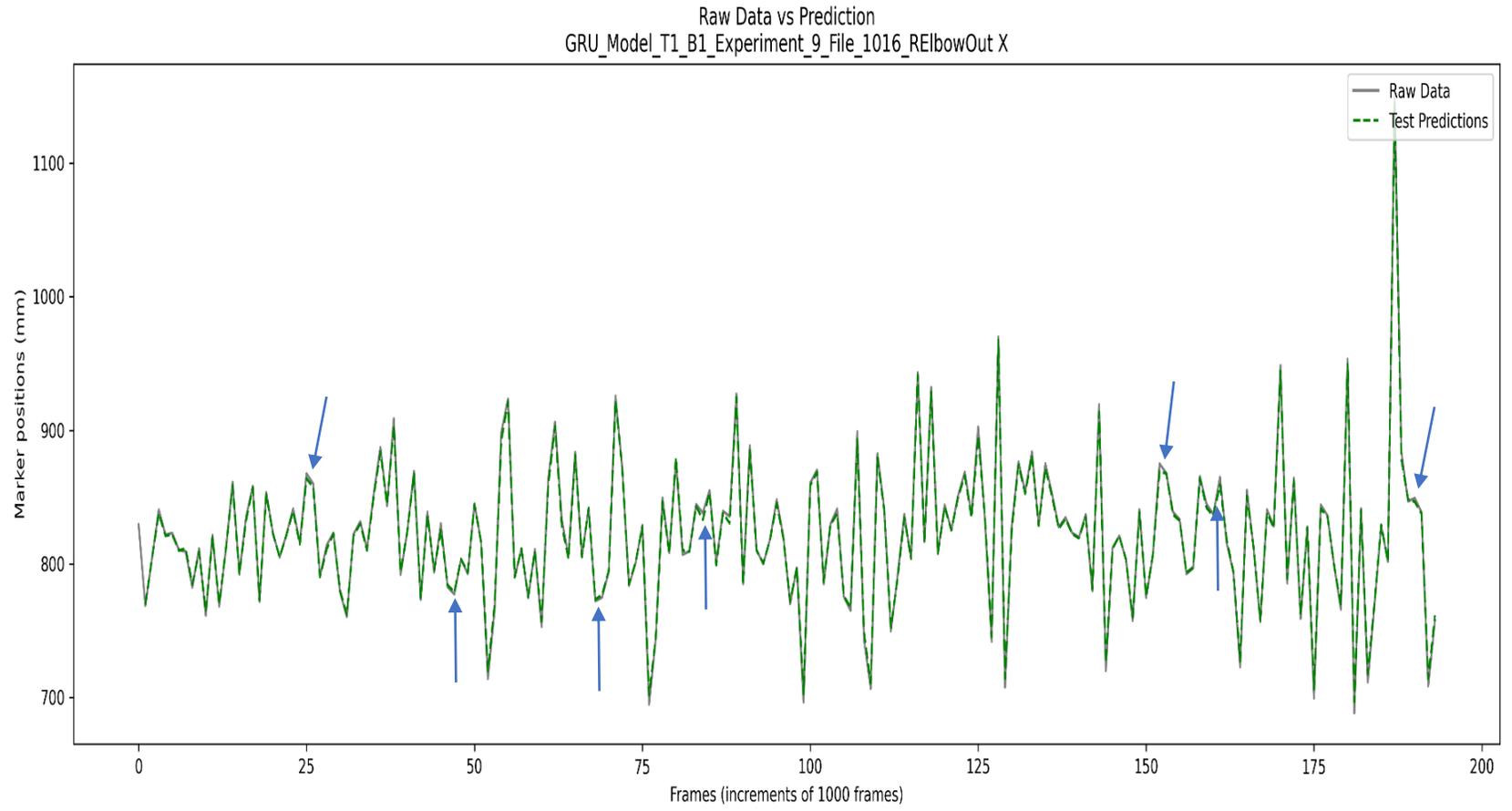


Figure 38 GRU motion forecasting 10 msec of 1016 with batch size 1

Looking at the loss curves for batch size 1 Fig. 35, there is a strong indication that some overfitting is occurring between the training and validation. There are big gaps between epoch four and eight, and the validation loss curves for MSE, RMSE and MAE do not converge to the training loss at the end of the 10th epoch. With 10 epochs, the model is not able to learn and fit well with the training data. Combining what is known from the loss curves and the forecasted plots, presents a correlation with a lot of missed predictions, in where the model does not follow the raw data This is indicated by all the blue arrows in Fig. 37 and 38. As with the GRU model, the predictions using 007 data presents more mispredictions of data since it does not fit well into the raw data.

GRU Worst Performance MoCap Forecasting with Batch Size of 10

In this section, the raw data against the predictive motion data of test data is plotted that performed the worst among all the experiments. These plots show the fitted model performance in predicting 50 msec, using GRU with a batch size of 10. Fig. 39 is the plot for subject 007 and Fig. 40 is the plot for subject 1016.

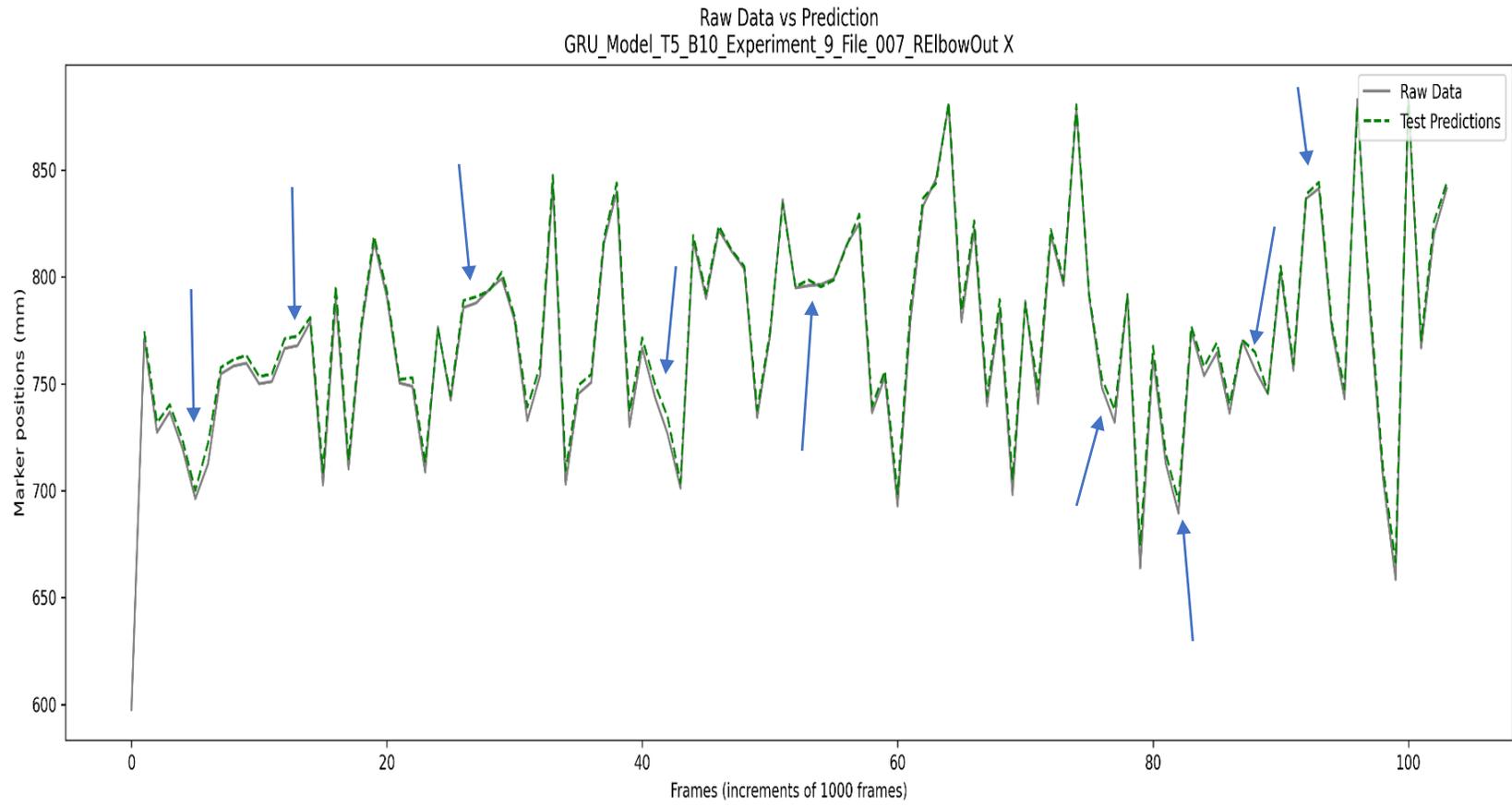


Figure 39 GRU motion forecasting 50 msec of 007 with batch size 10

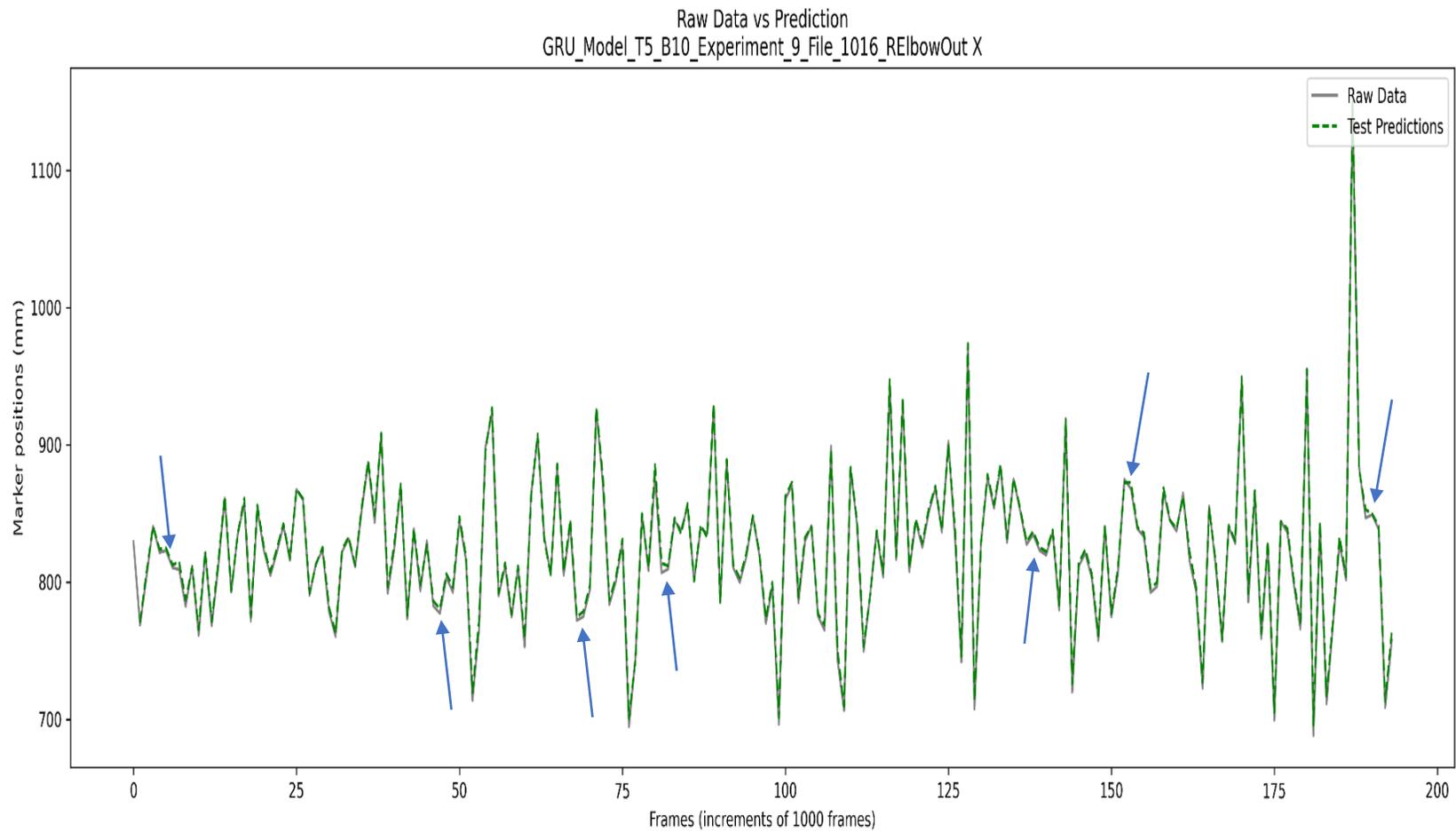


Figure 40 GRU motion forecasting 50 msec of 1016 with batch size 10

Looking at the loss curves Fig. 36 for a batch size of 10, the validation plot between epoch 0 and 4, there is a gap with the training loss. Afterward, the model seemed to fit well starting at epoch 4 since it began to converge to a loss of zero which will indicate a well-fitted model. This is not the case as the model begins to deviate from that at the eighth epoch meaning the model is not generalizing to the data. Combining what is known from the loss curves and the forecasted plots, presents a correlation with a lot of missed predictions, in where the model does not follow the raw data This is indicated by all the blue arrows in Fig. 39 and 40. subject 007 mispredictions are shown to be more apparent than with subject 1016.

GRU Best Performance Loss Curve Plots: Batch Size 1 & 10

The loss curves that proved to have better performance were plotted for the GRU implementation using batch sizes of 1 and 10. Fig. 41 is associated with forecasting 50 msec given 50 msec of previous data, using a batch size of 1. Fig 42, is associated with forecasting 10 msec given 10 msec of previous data, using a batch size of 10.

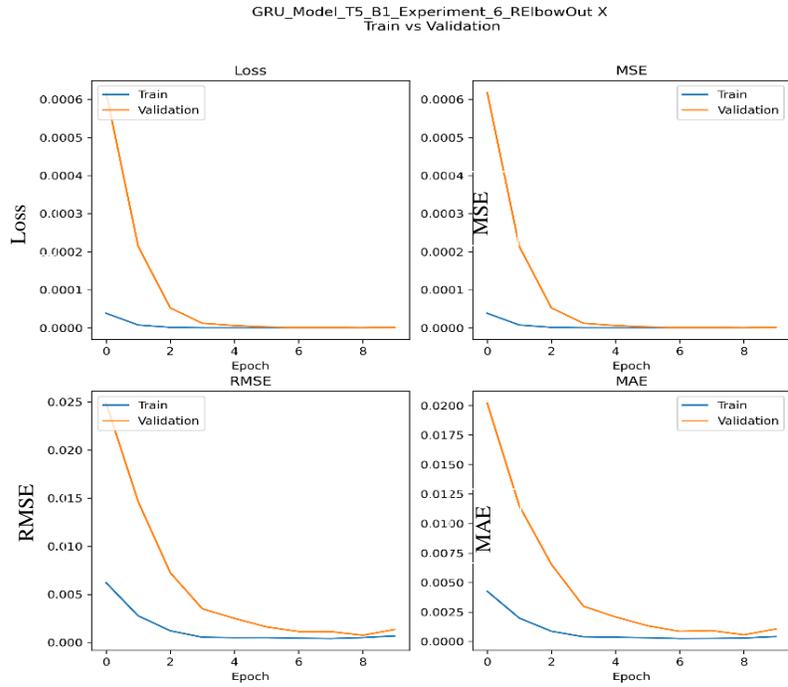


Figure 41 GRU 50 msec train and validation loss curves of using batch size 1.

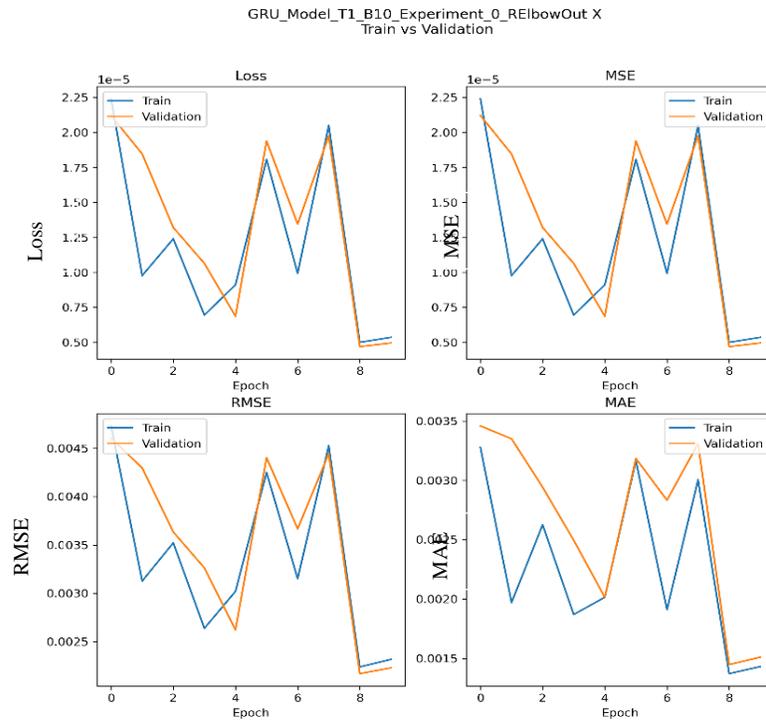


Figure 42 GRU 10 msec train and validation loss curves of using batch size 10.

GRU Best Performance MoCap Forecasting with Batch Size of 1

In this section, the raw data against the predictive motion data of test data is plotted which performed the best among all the experiments. These plots show the fitted model performance in predicting 50 msec, using GRU with a batch size of 1. Fig. 43 is the plot for subject 007 and Fig. 44 is the plot for subject 1016.

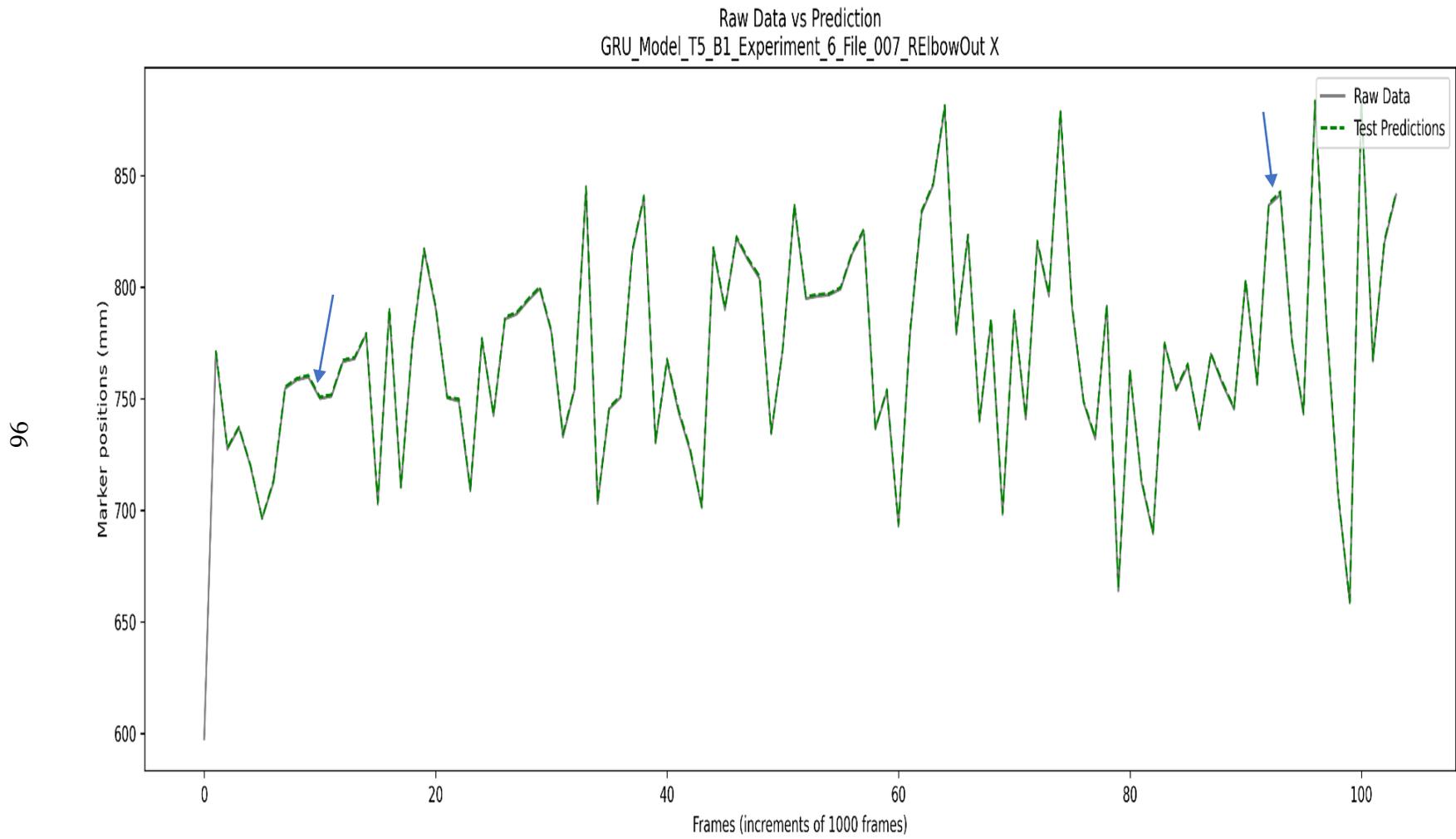


Figure 43 GRU motion forecasting 50 msec of 007 with batch size 1

Raw Data vs Prediction
GRU_Model_T5_B1_Experiment_6_File_1016_RElbowOut X

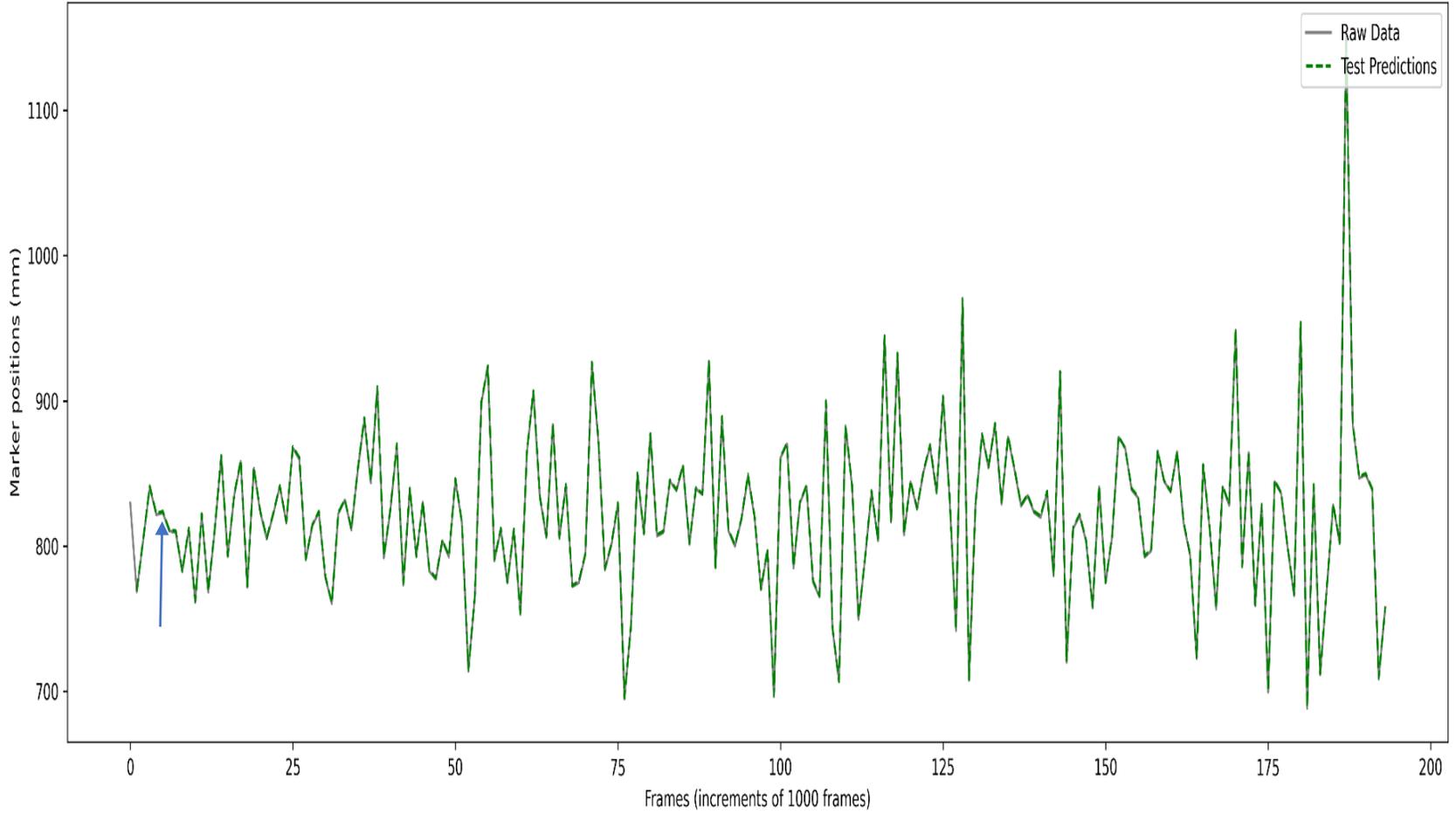


Figure 44 GRU motion forecasting 50 msec of 1016 with batch size 1

As seen in Fig. 41, the training vs validation loss curve plots indicates that the model is fitting well with unseen data. They both converge to zero at the end of epochs if looking at MSE, RMSE, and MAE, the error is close to zero. There is a small prediction error in the predictions of each subject. Fig. 43 and 44 show the visual plot of each subject when plotting the MoCap prediction. There are not as many wrong predictions visible to the naked eye for both subjects. subject 1016 test prediction is better this time around when using 50msec timestep prediction.

GRU Best Performance MoCap Forecasting with Batch Size of 10

In this section, the raw data against the predictive motion data of test data is plotted which performed the best among all the experiments. These plots show the fitted model performance in predicting 10 msec, using GRU with a batch size of 10. Fig. 45 is the plot for subject 007 and Fig. 46 is the plot for subject 1016.

66

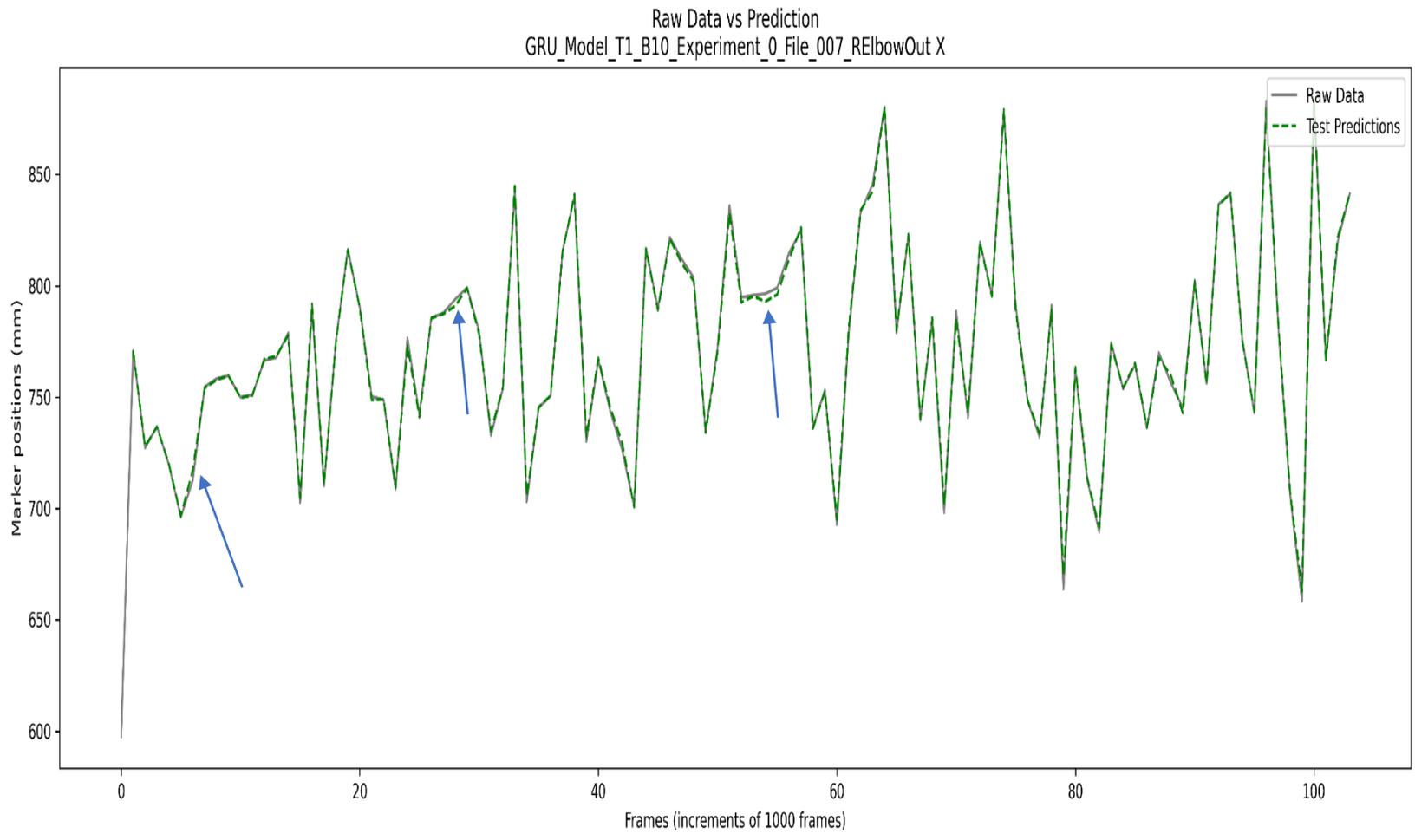


Figure 45 GRU motion forecasting 10 msec of 007 with batch size 10

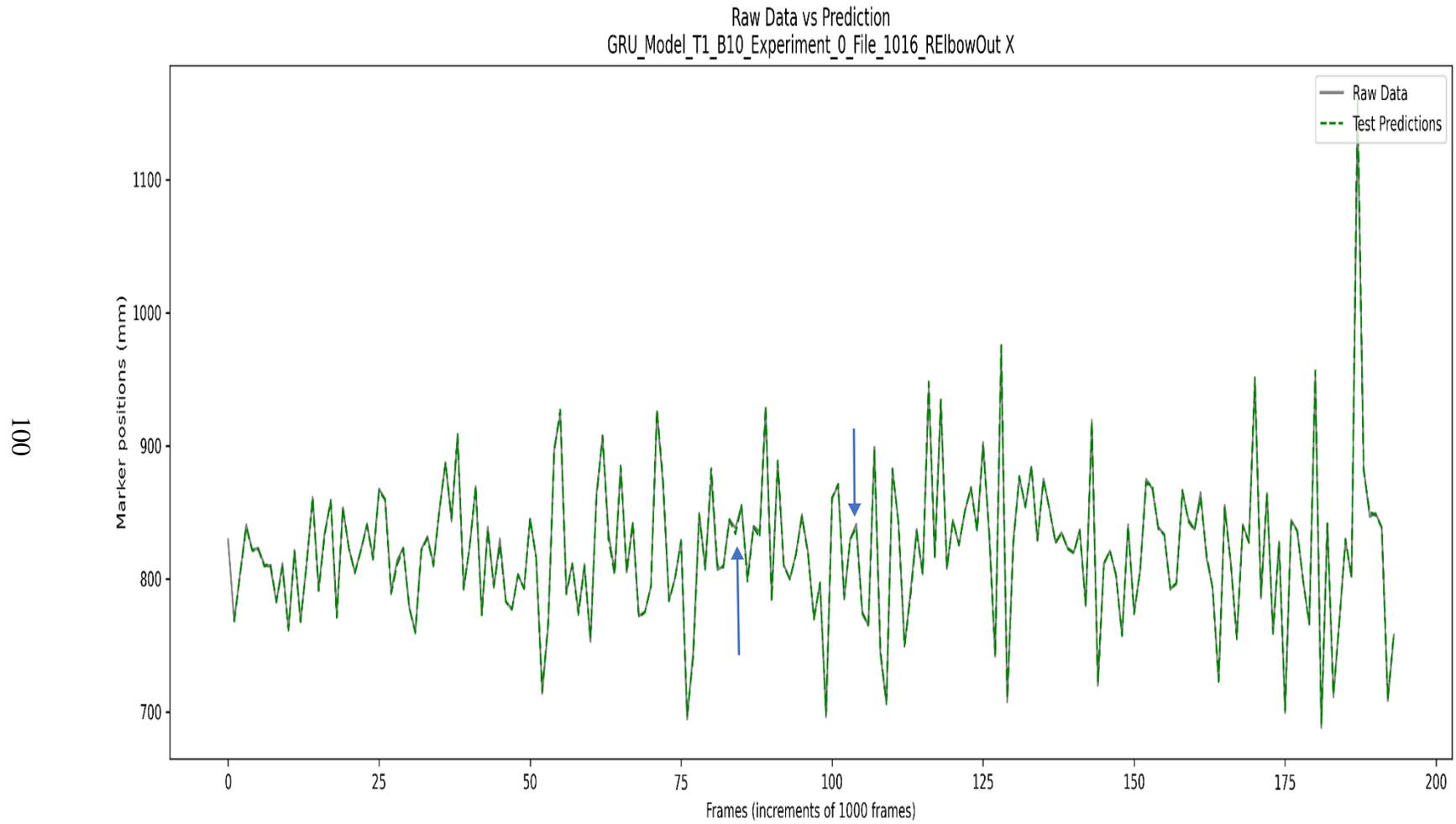


Figure 46 GRU motion forecasting 10 msec of 1016 with batch size 10

In Fig. 42, the training and validation follow closely throughout the ten epochs with validation in RMSE at epoch 4 having a lower prediction error but converges at the end. This indicates that the model trained will fit well with unseen data. This means using test data that was not used in training the model will perform well since the loss curves between training and validation converge close to zero. Fig. 45 presents several gaps between raw, and predictions indicated by the blue arrows. This means that for subject 007 the model performed relatively well. Compared to that, the subject 1016 plot in Fig. 46, does not have many mispredictions which means the GRU model performs better with this subject data. This could be due to the inconsistent amount of data for female subjects since 007 is a female subject, more female subject data could provide a better-fitted model.

GRU Univariate Test Metrics Summary Tables

A full summary of each test conducted using the GRU 1 layer model for univariate forecasting of MoCap RElbowOut X data is compiled into two tables below. These two tables correspond to the two different subjects used in the testing set: 007 and 1016. The best performing implementation for GRU is using a batch size of 1 with a timestep prediction of 50 msec.

Table 9 GRU Univariate Forecasting Metrics for Subject 007

| Subject 007 Female Test Metrics GRU | | | | | | | | | | |
|--|----------------------------------|------|-----------------------------------|-------|-------------------------------------|-------|-------------------------------------|------|--------------------------|------|
| | 10 milliseconds (1 Frame) | | 50 milliseconds (5 Frames) | | 100 milliseconds (10 frames) | | 200 milliseconds (20 frames) | | 1sec (100 frames) | |
| Batch Size | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 |
| Mean Square Error (MSE) | 78.57 | 7.85 | 0.78 | 10.74 | 4.02 | 11.14 | 6.13 | 8.84 | 5.45 | 9.18 |
| Root Mean Square Error (RMSE) | 7.53 | 2.59 | 0.85 | 3.04 | 1.87 | 2.85 | 2.25 | 2.52 | 2.06 | 2.54 |
| Mean Absolute Error (MAE) | 6.78 | 2.03 | 0.71 | 2.42 | 1.63 | 2.42 | 1.98 | 2.14 | 1.81 | 2.16 |

Table 10 GRU Univariate Forecasting Metrics for Subject 1016

| Subject 1016 Male Test Metrics GRU | | | | | | | | | | |
|---|----------------------------------|------|-----------------------------------|------|-------------------------------------|------|-------------------------------------|------|--------------------------|------|
| | 10 milliseconds (1 Frame) | | 50 milliseconds (5 Frames) | | 100 milliseconds (10 frames) | | 200 milliseconds (20 frames) | | 1sec (100 frames) | |
| Batch Size | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 10 |
| Mean Square Error (MSE) | 56.12 | 4.52 | 0.88 | 6.98 | 2.86 | 6.15 | 4.15 | 4.71 | 3.67 | 4.80 |
| Root Mean Square Error (RMSE) | 6.67 | 2.07 | 0.90 | 2.52 | 1.58 | 2.24 | 1.87 | 1.95 | 1.76 | 1.92 |
| Mean Absolute Error (MAE) | 6.05 | 1.57 | 0.74 | 1.97 | 1.38 | 1.85 | 1.64 | 1.60 | 1.56 | 1.57 |

Looking at the metrics for both subjects, there is a solid indication that training with a batch size of 10 does not increase the overall performance of the GRU model as can also be seen by higher error predictions compared to a batch size of 1. Comparing the metrics between LSTM and GRU metrics, GRU performed better with a batch size of 1. Therefore, training the GRU model with a batch size of 1 will provide better performance when using more than 10 msec of previous data.

Predicting Borg RPE value using MoCap Data

This section will discuss the experiment results on measuring fatigue based on the movement changes. For each time sequence, there will be 43 different inputs that will be used to predict the Borg value at that frame in time. In this case, the input at timestep t will look like $(X_t^0, X_t^1, \dots, X_t^{43}) = Y_{t+1}$ which contains the time at each frame and the data from each marker. The marker data is dependent on time so adding the time feature will help the machine learn the time in which the displacement of motions occurred corresponding to the Borg value of the participant. Each subject began at a Borg RPE value of 6 since they have not begun any activity indicating no exertion in the scale. For evaluating the performance, the metric plot curve will be looked at to see how the training data is actively reacting with learning the patterns. The LSTM and GRU are both ran using ten epochs and the file is randomly chosen in the beginning. Let's start with looking at their respective training metric curves that show how the errors were treated during each epoch.

LSTM_Model_T1_B1_1L_Borg_Prediction_Experiment_1_all
Train Metrics

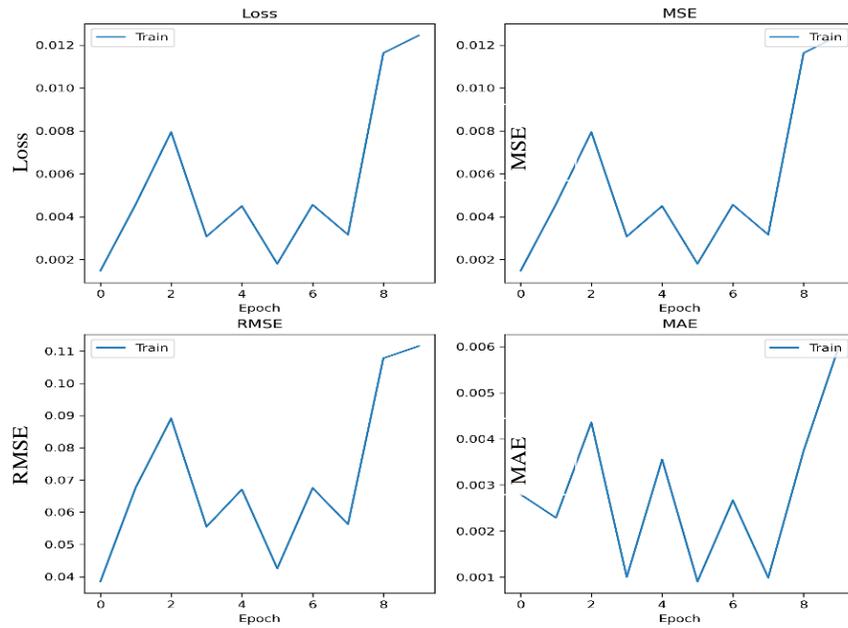


Figure 47 LSTM MoCap Borg Prediction Train Metrics Plot

GRU_Model_T1_B1_1L_Borg_Prediction_Experiment_0_all
Train Metrics

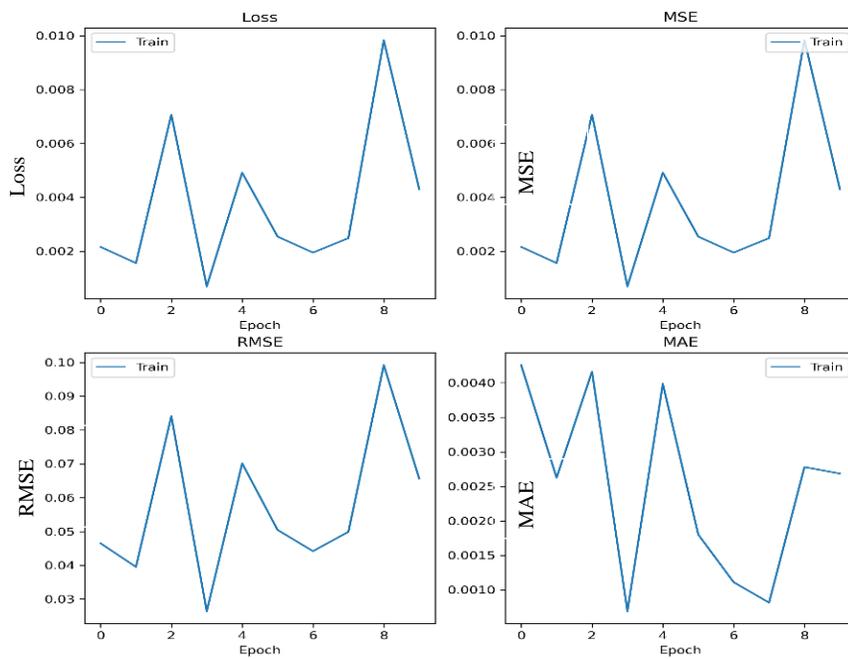


Figure 48 GRU MoCap Borg Prediction Train Metrics Plot

Looking at both metrics there are major differences in both implementations. Starting with the LSTM, it seems in the first epochs in the loss curve plot in Fig. 47 it was converging to 0 or close to it. Once it reached epoch 6 to 7 the loss will gain momentum and begin to experience bigger errors leading it to move upward and the algorithm is not able to learn within 10 epochs. Next looking at Fig 48, the GRU implementation looks like the opposite of the LSTM metrics in which it ends trying to converge the error to 0. Although each epoch is gradually increasing and decreasing between the predictive and target value calculations. This will also make it harder for GRU to learn the relationship between time and motion capture to predict Borg value.

Now after looking at the metrics plot, the next thing is to observe how the prediction from each model when using LSTM and GRU. If LSTM and GRU were able to create the relationship between the input data (the time per frame and MoCap data) and target data (Borg value) each prediction should begin at minimal at a Borg RPE of 6. Since each file contains the time of each frame the model should learn the association at the start of every file a value of 6 will be present.

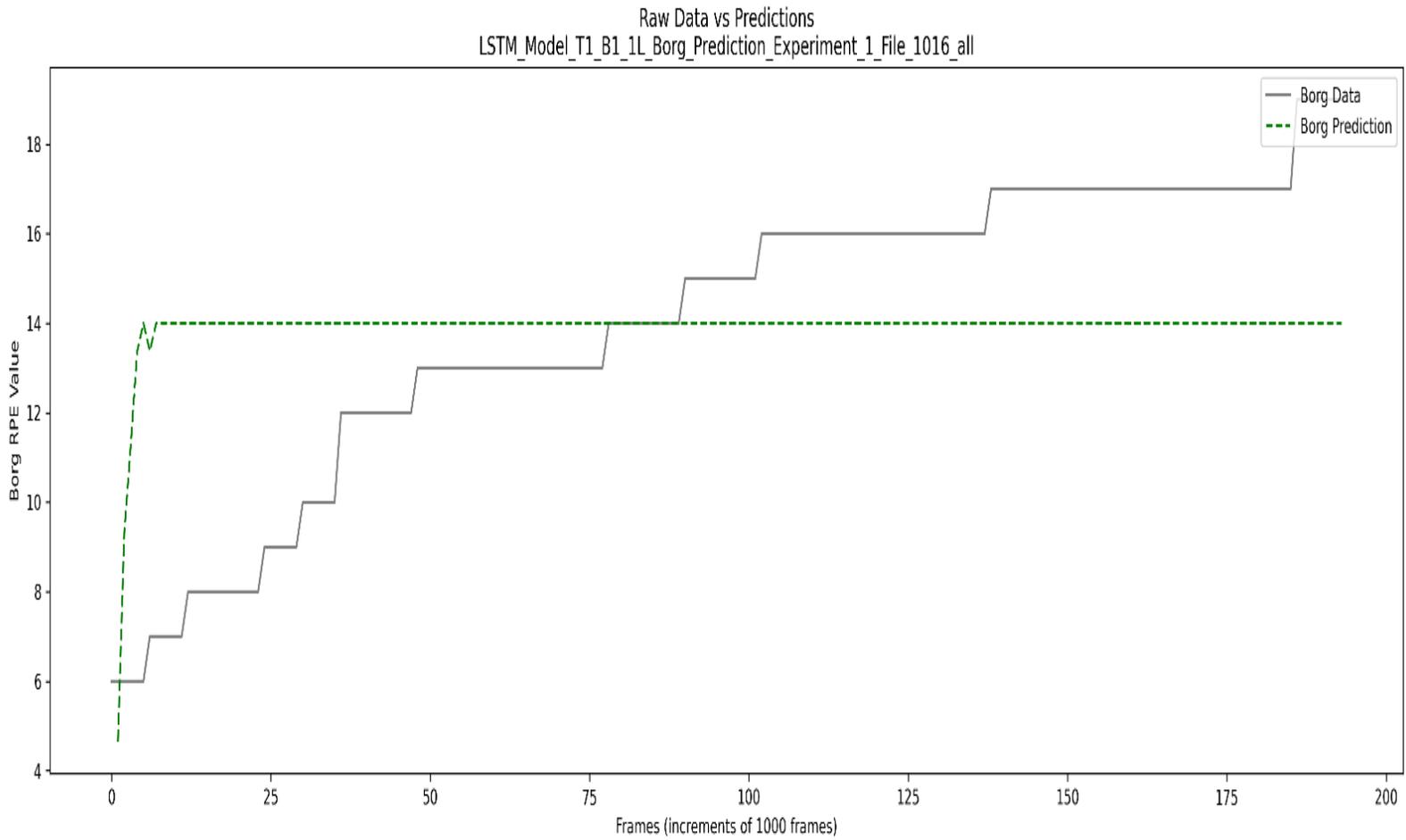


Figure 49 Borg Prediction of Subject 1016 using LSTM.

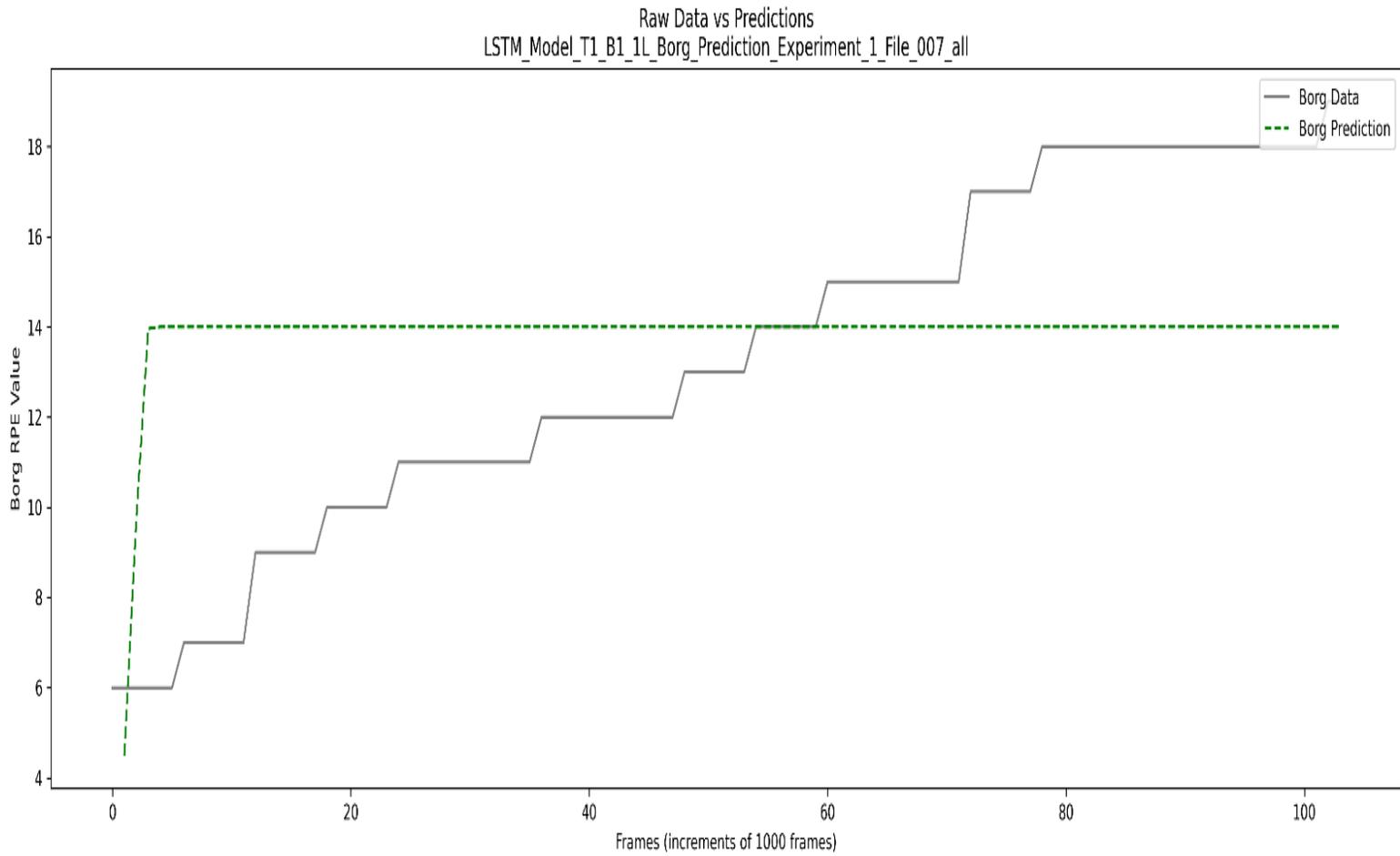


Figure 50 Borg Prediction of Subject 007 using LSTM.

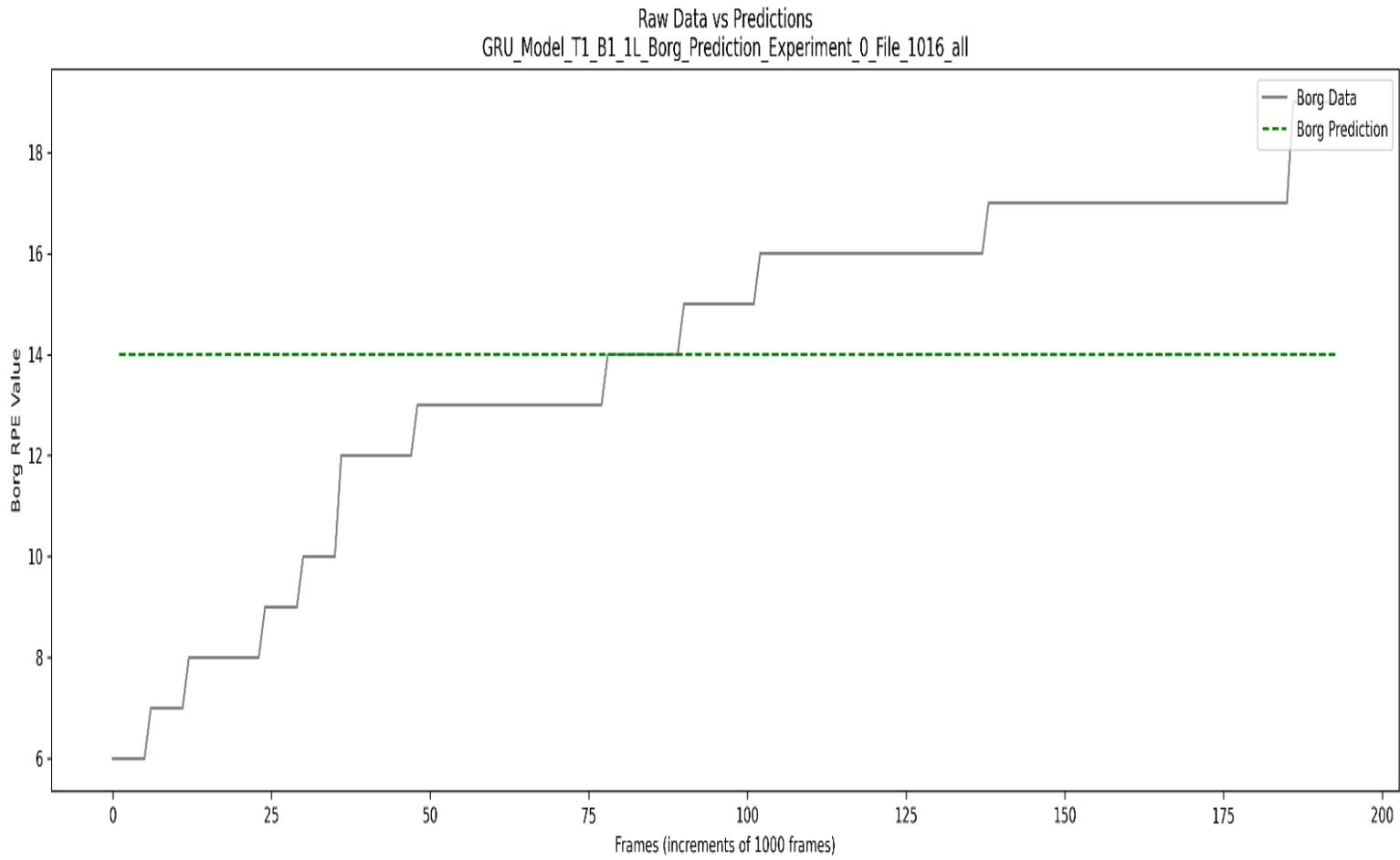


Figure 51 Borg Prediction for Subject 1016 using GRU.

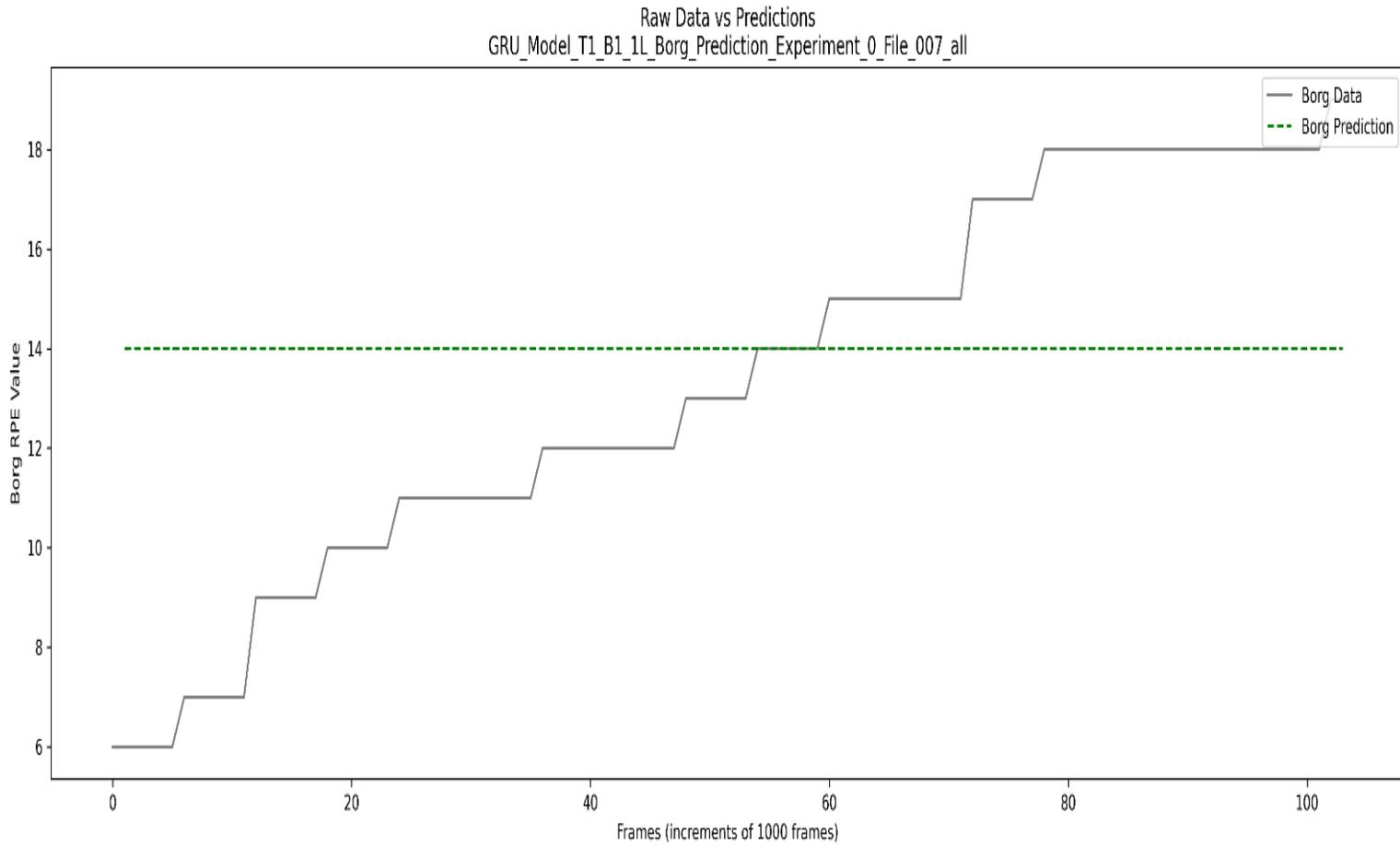


Figure 52 Borg Prediction for Subject 007 using GRU.

Comparing the LSTM and GRU figures, the LSTM produced a better prediction. Look at both Fig. 49 and Fig. 50 the prediction begins at a good start between 4 and 6 which is close to the initial RPE value of 6 that each subject report at the beginning of the experiment. Although it begins where it is expected, the prediction ends up consolidating up to an RPE value of 14 and does not continue to predict any further. Visually comparing the predicted line in green to the raw data in black, it is easy to see that the raw Borg data continues to linearly ascend upward but as stated before the RPE settles on 14. Similarly, the GRU performs a little differently than expected as seen in both Fig 51 and Fig 52. The same way as in LSTM the prediction is consolidated at 14 but the main difference is the beginning of the plot. The GRU does not predict anywhere near the initial value of 6 from starting point.

Using the MSE, RMSE, and MAE the errors between the predicted and true values are calculated to measure the performance of LSTM and GRU. The metrics are shown in Table 11 below and do not reflect the full picture which why having the metric plot and the prediction plot assist in analyzing the performance of the machine learning algorithm. Looking at MSE on average the error is greater than 10 for all four experiments. Since the loss function for these algorithms was the MSE, the updated values were treated as the average squared difference between the predicted and actual target data. In this case the larger the MSE the harder the machine will have to learn the actual data points. For this cause the higher the MSE the harder it is to interpret how well the model is performing since there is a certain range of values that the data contains for each RPE value.

Table 11 Borg Prediction Metrics for LSTM & GRU

| Testing Subject | 1016 (M) | | 007 (F) | |
|-----------------|----------|-------|---------|-------|
| ML Algorithm | LSTM | GRU | LSTM | GRU |
| MSE | 10.69 | 11.43 | 14.08 | 15.20 |
| RMSE | 3.27 | 3.38 | 3.75 | 3.90 |
| MAE | 2.70 | 2.77 | 3.22 | 3.32 |

Although in this case, there are issues with the balance in the data for each Borg value obtained from each subject. Below the RPE values for each subject used during the train can be found in Fig 53. Each process of collecting this data was first created in the mind of the subject picking and choosing when they felt exhausted and will indicate to the research in the lab that they would like to halt the experiment.

| Subject 005 (F) | | Subject 010 (M) | | Subject 012 (M) | | Subject 1001 (M) | | Subject 1002 (M) | | Subject 1007 (F) | | Subject 1013 (M) | | Subject 1017 (M) | |
|-----------------|--------------|-----------------|--------------|-----------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|
| Minute | Borg's Scale | Minute | Borg's Scale | Minute | Borg's Scale | Minute | Borg's Scale | Minute | Borg's Scale | Minute | Borg's Scale | Minute | Borg's Scale | Minute | Borg's Scale |
| 0 | 6 | 0 | 6 | 0 | 6 | 0 | 6 | 0 | 6 | 0 | 6 | 0 | 6 | 0 | 6 |
| 1 | 8 | 1 | 7 | 1 | 8 | 1 | 9 | 1 | 6 | 1 | 6 | 1 | 6 | 1 | 8 |
| 2 | 11 | 2 | 9 | 2 | 11 | 2 | 9 | 2 | 6 | 2 | 7 | 2 | 8 | 2 | 12 |
| 3 | 13 | 3 | 11 | 3 | 15 | 3 | 11 | 3 | 6 | 3 | 8 | 3 | 8 | 3 | 13 |
| 4 | 14 | 4 | 13 | 4 | 17 | 4 | 13 | 4 | 7 | 4 | 8 | 4 | 9 | 4 | 13 |
| 5 | 14 | 5 | 14 | | | 5 | 14 | 5 | 7 | 5 | 8 | 5 | 9 | 5 | 14 |
| | | 6 | 14 | | | 6 | 14 | 6 | 7 | 6 | 10 | 6 | 10 | 6 | 14 |
| | | 7 | 15 | | | 7 | 15 | 7 | 8 | 7 | 10 | 7 | 10 | 7 | 14 |
| | | 8 | 15 | | | 8 | 16 | 8 | 8 | 8 | 11 | 8 | 11 | 8 | 14 |
| | | 9 | 15 | | | 9 | 18 | 9 | 9 | 9 | 12 | 9 | 11 | 9 | 15 |
| | | 10 | 16 | | | | | 10 | 9 | 10 | 14 | 10 | 12 | 10 | 15 |
| | | | | | | | | 11 | 9 | 11 | 14 | 11 | 13 | 11 | 15 |
| | | | | | | | | 12 | 10 | 12 | 14 | 12 | 14 | 12 | 16 |
| | | | | | | | | 13 | 10 | 13 | 15 | 13 | 14 | 13 | 16 |
| | | | | | | | | 14 | 11 | 14 | 15 | 14 | 14 | 14 | 16 |
| | | | | | | | | 15 | 11 | 15 | 15 | 15 | 14 | 15 | 16 |
| | | | | | | | | 16 | 12 | 16 | 16 | 16 | 14 | 16 | 17 |
| | | | | | | | | 17 | 12 | | | 17 | 15 | 17 | 17 |
| | | | | | | | | 18 | 13 | | | 18 | 15 | 18 | 17 |
| | | | | | | | | 19 | 13 | | | 19 | 15 | 19 | 17 |
| | | | | | | | | 20 | 14 | | | 20 | 16 | 20 | 18 |
| | | | | | | | | 21 | 14 | | | 21 | 17 | 21 | 19 |
| | | | | | | | | 22 | 15 | | | 22 | 17 | | |
| | | | | | | | | 23 | 15 | | | 23 | 17 | | |
| | | | | | | | | 24 | 16 | | | 24 | 18 | | |
| | | | | | | | | 25 | 16 | | | 25 | 18 | | |
| | | | | | | | | 26 | 16 | | | 26 | 19 | | |
| | | | | | | | | 27 | 17 | | | | | | |
| | | | | | | | | 28 | 17 | | | | | | |
| | | | | | | | | 29 | 17 | | | | | | |
| | | | | | | | | 30 | 18 | | | | | | |
| | | | | | | | | 31 | 18 | | | | | | |
| | | | | | | | | 32 | 18 | | | | | | |

Figure 53 Self-Reported Borg Values from Each Subject used during training.

Since there was no consistency in when the subject will stop performing the lifting experiment this will cause the Borg value data to be inconsistent in that they would not all complete the task at the same level of fatigue. As can be seen in Fig. 53, each subject will end at a different fatigue value with the lowest value being 14. For this reason, the prediction consolidated to Borg value of 14 due to a disbalance when training the machine learning algorithm as there will be no consistent data to help the computer distinguish the patterns between the input and output values.

VIII. CONCLUSIONS

Limitations of Research

For this research there are limitations regarding the motion capture data collected was limited to the population of students and professors at Texas State University. This will be a different application compared to experts in the industry that perform MMH tasks regularly. Although the lab setting was to mimic an MMH task of lifting, the worker's conditions will be much different as they might change their lifting habits compared to students who just lift a box. This means participants cannot be generalized to MMH workers since it was only the population from the university. This limited the number of participants since every person had to be recruited to join the project, which can be unreliable as people's schedules might not fit into the experiment's time. Along with that is the population of subjects that were recruited to conduct the experiments. There were more males than females at the end of the data collection process, so that can create some bias in motion since there is a gender imbalance.

Also, during the experiments, each person had to self-report their level of fatigue based on the Borg scale which is subjective to themselves. Within the data used there were inconsistencies in when the individual felt exhausted so they would stop the experiment. This would cause a different range of Borg values between subjects which affects the machine learning process. With ML there is a need for consistent data to help teach the patterns and since each subject would complete their lifting experiment at different levels this would create problems and limit the patterns learned by the machine.

Another limitation to this study is the motion dataset itself. Since it is an OMCS with markers attached to participants using double adhesive tape different possibilities

could affect the motion capture data. One thing will be when placing the markers on the participants, no guarantee placing the markers will be in the exact spot-on subjects every time, this could cause some deviation in the data that could add some inconsistency between datasets. Another limitation is the double adhesive tape might not remain sticky enough throughout the experiment so it might slip off the subject and caused gaps in the data. This is a byproduct of subjects sweating that causes the adhesion on the tape to decrease and then the marker falls off.

Lastly, there were gaps in the set of experiments in which subjects still needed to perform some experiments. Each subject was to complete a set of six experiments using different parameters but within each parameter, there were missing gaps, in where the subject did not complete the task. This caused some incomplete data to occur between experiments.

Delimitations

Although there were other MoCap experiments conducted such as other lifting parameters with different heights as seen in Table II, only one parameter was considered due to the nature of how many subjects and datasets were collected. The parameter as stated in the previous chapters was the lifting height of 51 cm at an interval of 9 seconds between lifts.

Conclusions

In this paper, the dataset of human motion for lifting and the experiment procedure to capture such motion were introduced. Even though there were multiple subjects at different lifting tasks, only ten datasets containing lifting human motion were considered for the research. This dataset contained a fatigue level from each subject

based on the Borg RPE value which can be used as an indicator when the person might begin to feel some discomfort in their motions.

Although using LSTM and GRU proved to work to some extent on forecasting one individual marker data using the data provided there can be further work done. In this case, the data that was used was only normalized to range from [-1,1] to try and keep a simple solution that could fit into the LSTM without much preprocessing needed but other normalization techniques such as forward kinematics [40] or root normalization in which each segment is relative to a root segment [77].can be looked at. In general, this study provides evidence for one set of activities during MMM to forecast motion. Further studies should be conducted to understand more of the biomechanics since this implementation only regards the use of the relative marker positions attached to the subjects. For this system, it might work but for another MoCap system, it might not achieve the same results so using further knowledge in biomechanics can help focus on the mechanics of the WMSD risk from the lifting motions. Another way to boost the work would be to include multiple markers to predict the motion of the whole body [38, 47]. This could be something that could be used to predict the motion of a worker in the environment or be able to detect the posture of the worker and detect if they are performing a wrong lift or any task related to MMH activities.

In contrast, for this study, predicting the fatigue level based on the displacement of the motion over time proved to not achieve the expected results. When using LSTM, it proved to measure close to the beginning of the Borg scale of 6 and slightly perform better than GRU. In the end, both algorithms will consolidate to only predict around an RPE value of 14 and will not continue to predict any further. As described in the

limitation section, each subject had to self-report their subjective level of exhaustion using the Borg RPE scale. Their exhaustion level although subjectively reported was still dependent on the movement they were performing repetitively throughout the experiment. Since each person will react differently to the experiment, each subject was indicated to let the lab assistant when they wanted to stop the experiment. This will cause inconsistency in the data since each subject will end at a different level of fatigue. The model will tend to predict data that is more common and less rare in these cases which why it will make it difficult for each model to learn patterns with inconsistent data.

Contributions

In this thesis, some methods for understanding human motion using deep learning techniques for forecasting data.

My research has contributed to the field of Motion Capture and MMH by:

1. Design of Motion Capture experiments for lifting task as seen in MMH.
2. Collection of Motion data from different subjects in the university.
3. Confirm the idea of being able to forecast a single marker data based on historical data.
4. Study the performance of predicting Borg value using only the displacement of motions throughout each frame.

Future Work

In this research, the use of OMCS is used to collect human motion but requires equipment that must be placed beforehand and is usually permanently left there. High-frequency cameras are not as portable and are typically expensive. Multiple cameras will be required which also depends on the area of the environment in which the application

will be applied. For example, in a manufacturing factory, there is machinery, shelves, people regularly walking, etc. there will have to be cameras in places that will not obstruct the workplace and have a view of the workers. Additionally, reflective markers must be attached to multiple individuals which will require a longer setup before work even starts. Using an OMCS will not be ideal in a warehouse setting or any type of setting in where there is a lot of mobility. Therefore, for future work, using IMUs can provide a better implementation for data collection that will be less intrusive to the work environment [22, 51, 54, 78, 79].

Another idea for future work is to use a different type of machine learning algorithm to forecast either the motion data or the exhaustion level of individuals. A novel ML technique is also a deep learning architecture known as Transformer. A Transformer application is transforming an input sequence into an output sequence which can include speech recognition, trajectory forecasting, or any sequential data [80, 81]. This model is said to be faster than Long-Short Term Memory and facilitates more parallelization during training, it will reduce the time of training. The Transformer uses a set of encoder and decoder layers to process and generate encoding that contains information about which input value is relevant to each other [82].

REFERENCES

- [1] A. Rojko, "Industry 4.0 Concept: Background and Overview," *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 11, no. 5, 2017, doi: 10.3991/ijim.v11i5.7072
- [2] B. Vogel-Heuser and D. Hess, "Guest Editorial Industry 4.0–Prerequisites and Visions," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 411-413, 2016, doi: 10.1109/tase.2016.2523639
- [3] D. Vuksanović, J. Ugarak, and D. Korčok, "Industry 4.0: the Future Concepts and New Visions of Factory of the Future Development," presented at the Proceedings of the International Scientific Conference - Sinteza 2016, 2016.
- [4] S. Simoneau, M. St-Vincent, and D. Chicoine, "Work-Related Musculoskeletal Disorders (WMSDs) - A Better Understanding for More Effective Prevention," *Québec:IRSST*, 1996.
- [5] A. D. Banks and F. Aghazadeh, "Progressive fatigue effects on manual lifting factors," *Human Factors and Ergonomics in Manufacturing*, vol. 19, no. 5, pp. 361-377, 2009-09-01 2009, doi: 10.1002/hfm.20170[Accessed:2020-09-12T05:48:16].
- [6] L. Cavuoto and F. Megahed, "Understanding Fatigue and the Implications for Worker Safety," presented at the ASSE Professional Development Conference and Exposition, Atlanta, Georgia, USA, 2016/1/1/, 2016. [Online].
- [7] V. J. Gawron, J. French, and D. Funke, "An overview of fatigue," in *Stress, workload, and fatigue.*, (Human factors in transportation. Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers, 2001, pp. 581-595.

- [8] M. C. Sharpe *et al.*, "A report--chronic fatigue syndrome: guidelines for research," (in eng), *J R Soc Med*, vol. 84, no. 2, pp. 118-121, 1991. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/1999813>
- [9] M. Yung, "Fatigue at the Workplace: Measurement and Temporal Development," 2016.
- [10] V. C. Knott, A. Wiest, and K. Bengler, "Repetitive Lifting Tasks in Logistics – Effects on Humans at Different Lifting Task Durations," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 60, no. 1, pp. 2034-2038, 2016, doi: 10.1177/1541931213601462
- [11] M. Spallek, W. Kuhn, S. Uibel, A. Van Mark, and D. Quarcoo, "Work-related musculoskeletal disorders in the automotive industry due to repetitive work - implications for rehabilitation," *Journal of Occupational Medicine and Toxicology*, vol. 5, no. 1, p. 6, 2010, doi: 10.1186/1745-6673-5-6
- [12] P. Bonato, M. S. S. Heng, J. Gonzalez-Cueto, A. Leardini, J. O'Connor, and S. H. Roy, "EMG-based measures of fatigue during a repetitive squat exercise," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 6, pp. 133-143, 2001-01-01 2001, doi: 10.1109/51.982285[Accessed:2020-09-12T03:40:34].
- [13] N. Strimpakos, G. Georgios, K. Eleni, K. Vasilios, and O. Jacqueline, "Issues in relation to the repeatability of and correlation between EMG and Borg scale assessments of neck muscle fatigue," *Journal of Electromyography and Kinesiology*, vol. 15, no. 5, pp. 452-465, 2005-10-01 2005, doi: 10.1016/j.jelekin.2005.01.007[Accessed:2020-09-12T03:40:56].

- [14] Y. Yu, H. Li, X. Yang, L. Kong, X. Luo, and A. Y. L. Wong, "An automatic and non-invasive physical fatigue assessment method for construction workers," *Automation in Construction*, vol. 103, pp. 1-12, 2019-07-01 2019, doi: 10.1016/j.autcon.2019.02.020[Accessed:2020-09-12T02:51:17].
- [15] M. Karg, G. Venture, J. Hoey, and D. Kulic, "Human movement analysis as a measure for fatigue: a hidden Markov-based approach," *IEEE Trans Neural Syst Rehabil Eng*, vol. 22, no. 3, pp. 470-81, May 2014, doi: 10.1109/TNSRE.2013.2291327
- [16] V. Valentina, S. Fabio, C. Martina, and P. Alessandro, "Fatigue accumulation in the assignment of manual material handling activities to operators," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 826-831, 2018-01-01 2018, doi: 10.1016/j.ifacol.2018.08.441[Accessed:2020-09-11T22:10:39].
- [17] G. C. David, "Ergonomic methods for assessing exposure to risk factors for work-related musculoskeletal disorders," *Occupational Medicine*, vol. 55, no. 3, pp. 190-199, 2005-05-01 2005, doi: 10.1093/occmed/kqi082[Accessed:2020-09-12T05:53:36].
- [18] G. Borg, "Psychophysical scaling with applications in physical work and the perception of exertion," *Scand J Work Environ Health*, vol. 16 Suppl 1, pp. 55-8, 1990, doi: 10.5271/sjweh.1815
- [19] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent Network Models for Human Dynamics," *arXiv e-prints*, p. arXiv:1508.00271. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2015arXiv150800271F>

- [20] M. van Tulder, A. Malmivaara, and B. Koes, "Repetitive strain injury," *The Lancet*, vol. 369, no. 9575, pp. 1815-1822, 2007, doi: 10.1016/s0140-6736(07)60820-4
- [21] U. S. D. o. L. Bureau of Labor Statistics, "Number, incidence rate, and median days away from work for nonfatal occupational injuries and illnesses involving days away from work and musculoskeletal disorders by selected worker occupation and ownership," *The Economics Daily*, 2015.
- [22] Z. Sedighi Maman, M. A. Alamdar Yazdi, L. A. Cavuoto, and F. M. Megahed, "A data-driven approach to modeling physical fatigue in the workplace using wearable sensors," *Appl Ergon*, vol. 65, pp. 515-529, Nov 2017, doi: 10.1016/j.apergo.2017.02.001
- [23] P. Nogueira, "Motion Capture Fundamentals A Critical and Comparative Analysis on Real-World Applications," 2012.
- [24] M. Field, Z. Pan, D. Stirling, and F. Naghdy, "Human motion capture sensors and analysis in robotics," *Industrial Robot: An International Journal*, vol. 38, no. 2, pp. 163-171, 2011, doi: 10.1108/01439911111106372
- [25] S. Kim and M. A. Nussbaum, "An evaluation of classification algorithms for manual material handling tasks based on data obtained using wearable technologies," *Ergonomics*, vol. 57, no. 7, pp. 1040-51, 2014, doi: 10.1080/00140139.2014.907450

- [26] J. C. C. Guedes, R. Martins, and F. Bernardo, "Evaluation of physical fatigue based on motion analysis," *International Journal of Occupational and Environmental Safety*, vol. 3, no. 1, pp. 13-26, 2019, doi: 10.24840/2184-0954_003.001_0002
- [27] D. Battini, A. Persona, and F. Sgarbossa, "Innovative real-time system to integrate ergonomic evaluations into warehouse design and management," *Computers & Industrial Engineering*, vol. 77, pp. 1-10, 2014, doi: 10.1016/j.cie.2014.08.018
- [28] N. Vignais, M. Miezal, G. Bleser, K. Mura, D. Gorecky, and F. Marin, "Innovative system for real-time ergonomic feedback in industrial manufacturing," *Appl Ergon*, vol. 44, no. 4, pp. 566-74, Jul 2013, doi: 10.1016/j.apergo.2012.11.008
- [29] P. Plantard, H. P. H. Shum, A. S. Le Pierres, and F. Multon, "Validation of an ergonomic assessment method using Kinect data in real workplace conditions," *Appl Ergon*, vol. 65, pp. 562-569, Nov 2017, doi: 10.1016/j.apergo.2016.10.015
- [30] I. Akhter and M. J. Black, "Pose-conditioned joint angle limits for 3D human pose reconstruction," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015-06-01 2015: IEEE, doi: 10.1109/cvpr.2015.7298751
- [31] J. H. Fernando De la Torre, Javier Montano, Sergio Valcarcel, R Forcada, and J Macey. . Guide to the Carnegie Mellon University multimodal activity (CMU-MMAC) database.

- [32] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325-1339, 2014-07-01 2014, doi: 10.1109/tpami.2013.248[Accessed:2021-04-12T16:14:24].
- [33] T. R. M. Müller, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, *Documentation Mocap Database HDM05*. (HDM05. Technical Report CG-2007-2). June 2007.
- [34] N. F. Troje, "Decomposing biological motion: A framework for analysis and synthesis of human gait patterns," *Journal of Vision*, vol. 2, no. 5, p. 2, 2002-09-10 2002, doi: 10.1167/2.5.2[Accessed:2021-04-12T16:24:56].
- [35] L. Gui, Y.-X. Wang, X. Liang, and J. M. F. Moura, "Adversarial Geometry-Aware Human Motion Prediction," in *ECCV*, 2018.
- [36] L. Gui, Y.-X. Wang, D. Ramanan, and J. M. F. Moura, "Few-Shot Human Motion Prediction via Meta-learning," in *ECCV*, 2018.
- [37] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep Learning on Spatio-Temporal Graphs," *arXiv e-prints*, p. arXiv:1511.05298. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2015arXiv151105298J>
- [38] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," *arXiv e-prints*, p. arXiv:1705.02445. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2017arXiv170502445M>
- [39] D. Pavllo, D. Grangier, and M. J. A. Auli, "QuaterNet: A Quaternion-based Recurrent Model for Human Motion," vol. abs/1805.06485, 2018.

- [40] A. M. Lehrmann, P. V. Gehler, and S. Nowozin, "Efficient Nonlinear Markov Models for Human Motion," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 23-28 June 2014 2014, pp. 1314-1321.
- [41] K. Kipp, M. Giordanelli, and C. Geiser, "Predicting net joint moments during a weightlifting exercise with a neural network model," *J Biomech*, vol. 74, pp. 225-229, Jun 6 2018, doi: 10.1016/j.jbiomech.2018.04.021
- [42] G. W. Taylor, G. E. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," presented at the Proceedings of the 19th International Conference on Neural Information Processing Systems, Canada, 2006.
- [43] T. T. Um and D. Kulić, "An unsupervised approach to detecting and isolating athletic movements," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 16-20 Aug. 2016 2016, pp. 6268-6272, doi: 10.1109/EMBC.2016.7592161
- [44] M. F. Antwi-Afari, H. Li, D. J. Edwards, E. A. Pärn, J. Seo, and A. Y. L. Wong, "Biomechanical analysis of risk factors for work-related musculoskeletal disorders during repetitive lifting task in construction workers," *Automation in Construction*, vol. 83, pp. 41-47, 2017-11-01 2017, doi: 10.1016/j.autcon.2017.07.007[Accessed:2021-04-12T16:51:05].
- [45] W. Karwowski, *International Encyclopedia of Ergonomics and Human Factors, Second Edition - 3 Volume Set*. CRC Press, Inc., 2006.

- [46] L. Punnett and D. H. Wegman, "Work-related musculoskeletal disorders: the epidemiologic evidence and the debate," (in eng), *Journal of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology*, vol. 14, no. 1, pp. 13-23, Feb 2004, doi: 10.1016/j.jelekin.2003.09.015
- [47] M. A. Perez and M. A. Nussbaum, "A neural network model for predicting postures during non-repetitive manual materials handling tasks," *Ergonomics*, vol. 51, no. 10, pp. 1549-64, Oct 2008, doi: 10.1080/00140130802220570
- [48] B. Barazandeh, K. Bastani, M. Rafieisakhaei, S. Kim, Z. Kong, and M. A. Nussbaum, "Robust Sparse Representation-Based Classification Using Online Sensor Data for Monitoring Manual Material Handling Tasks," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1573-1584, 2018, doi: 10.1109/tase.2017.2729583
- [49] J. Chen, J. Qiu, and C. Ahn, "Construction worker's awkward posture recognition through supervised motion tensor decomposition," *Automation in Construction*, vol. 77, pp. 67-81, 2017, doi: 10.1016/j.autcon.2017.01.020
- [50] W. Lee, K. Lin, E. Seto, and G. C. J. A. i. C. Migliaccio, "Wearable sensors for monitoring on-duty and off-duty worker physiological status and activities in construction," vol. 83, pp. 341-353, 2017.
- [51] W. Lee, E. Seto, K. Y. Lin, and G. C. Migliaccio, "An evaluation of wearable sensors and their placements for analyzing construction worker's trunk posture in laboratory conditions," (in eng), *Appl Ergon*, vol. 65, pp. 424-436, Nov 2017, doi: 10.1016/j.apergo.2017.03.016

- [52] M. C. Schall, Jr., N. B. Fethke, H. Chen, S. Oyama, and D. I. Douphrate, "Accuracy and repeatability of an inertial measurement unit system for field-based occupational studies," (in eng), *Ergonomics*, vol. 59, no. 4, pp. 591-602, Apr 2016, doi: 10.1080/00140139.2015.1079335
- [53] E. Valero, A. Sivanathan, F. Bosché, and M. J. A. i. C. Abdel-Wahab, "Analysis of construction trade worker body motions using a wearable and wireless motion sensor network," vol. 83, pp. 48-55, 2017.
- [54] X. Yan, H. Li, A. Li, and H. J. A. i. C. Zhang, "Wearable IMU-based real-time motion warning system for construction workers' musculoskeletal disorders prevention," vol. 74, pp. 2-11, 2017.
- [55] C. Drury, C.-H. Law, C. S. J. H. F. T. J. o. H. F. Pawenski, and E. Society, "A Survey of Industrial Box Handling," vol. 24, pp. 553 - 565, 1982.
- [56] A. S. Ljungberg, A. Kilbom, and G. M. Hägg, "Occupational lifting by nursing aides and warehouse workers," (in eng), *Ergonomics*, vol. 32, no. 1, pp. 59-78, Jan 1989, doi: 10.1080/00140138908966068
- [57] T. Op De Beéck, W. Meert, K. Schütte, B. Vanwanseele, and J. Davis, "Fatigue Prediction in Outdoor Runners Via Machine Learning and Sensor Fusion," presented at the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.
- [58] S. Karvekar, M. Abdollahi, and E. Rashedi, "A Data-Driven Model to Identify Fatigue Level Based on the Motion Data from a Smartphone," in *2019 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, 2019-10-01 2019: IEEE, doi: 10.1109/wnyipw.2019.8923100

- [59] T. Carroll *et al.*, "Comparison of Inverse Kinematics Algorithms for Digital Twin Industry 4.0 Applications," 2020: IEEE, doi: 10.1109/smc42975.2020.9283253.[Online]. Available: <https://dx.doi.org/10.1109/smc42975.2020.9283253>
- [60] G. Hernandez *et al.*, "Machine Learning Techniques for Motion Analysis of Fatigue from Manual Material Handling Operations Using 3D Motion Capture Data," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 6-8 Jan. 2020 2020, pp. 0300-0305, doi: 10.1109/CCWC47524.2020.9031222
- [61] Qualisys. "Fill types." https://docs.qualisys.com/getting-started/content/37_trajectory_editor_series/37b_how_to_use_the_trajectory_editor_-_gap-filling/fill_types.htm [Accessed.
- [62] S. H. Snook and V. M. Ciriello, "The design of manual handling tasks: revised tables of maximum acceptable weights and forces," *Ergonomics*, vol. 34, no. 9, pp. 1197-1213, 1991/09/01 1991, doi: 10.1080/00140139108964855
- [63] Qualisys. <https://www.qualisys.com/> [Accessed.
- [64] Hexoskin. <https://www.hexoskin.com/> [Accessed.
- [65] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735-80, Nov 15 1997, doi: 10.1162/neco.1997.9.8.1735
- [66] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," vol. abs/1406.1078, 2014.

- [67] "Learning, Exploration, Analysis, and Processing (LEAP) - High Performance Computing Cluster." <https://doit.txstate.edu/rc/leap.html> [Accessed: Apr. 22, 2021)].
- [68] H. R. Group. "High-Performance Engineering (HiPE) Dell Rack Servers." <https://hipe.wp.txstate.edu/technology/hipe-dell-rack-servers/> [Accessed: Apr. 20, 2021].
- [69] F. Chollet. "Keras." <https://keras.io> [Accessed: Apr. 20, 2020].
- [70] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, 2007, doi: 10.1109/MCSE.2007.55
- [71] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357-362, 2020-09-17 2020, doi: 10.1038/s41586-020-2649-2[Accessed:August 8, 2021].
- [72] E. Gazoni and C. Clark. "openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files." <https://openpyxl.readthedocs.io/> [Accessed: August 8, 2020].
- [73] W. Mckinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, 2010-01-01 2010: SciPy, doi: 10.25080/majora-92bf1922-00a
- [74] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Accessed:Apr. 20, 2021].
- [75] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow, 2nd Edition*. Packt Publishing, 2017.

- [76] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247-1250, 2014, doi: 10.5194/gmd-7-1247-2014
- [77] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. J. A. T. o. G. Pons-Moll, "Deep inertial poser," vol. 37, pp. 1 - 15, 2018.
- [78] S. Pham *et al.*, "Wearable Sensor System to Monitor Physical Activity and the Physiological Effects of Heat Exposure," (in eng), *Sensors (Basel)*, vol. 20, no. 3, Feb 6 2020, doi: 10.3390/s20030855
- [79] M. Fera *et al.*, "Towards Digital Twin Implementation for Assessing Production Line Performance and Balancing," *Sensors (Basel)*, vol. 20, no. 1, Dec 23 2019, doi: 10.3390/s20010097
- [80] F. Juliari, I. Hasan, M. Cristani, and F. J. A. Galasso, "Transformer Networks for Trajectory Forecasting," vol. abs/2003.08111, 2020.
- [81] E. Aksan, P. Cao, M. Kaufmann, and O. J. A. Hilliges, "Attention, please: A Spatio-temporal Transformer for 3D Human Motion Prediction," vol. abs/2004.08692, 2020.
- [82] A. Vaswani *et al.*, "Attention is All you Need," vol. abs/1706.03762, 2017.