UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Automatic IMU-to-Segment Labelling Using Deep learning Approaches

Ruiyuan Li

M.Sc. Thesis
August 2022

Supervisors:
dr. M. Poel (Mannes)
dr. N. Strisciuglio (Nicola)

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Daily Supervisors:
dr.ir. F.J.Wouda
MSc. Xiaowen Song

Movella Inc.
Pantheon 6-A + 8A
7521 PR Enschede
The Netherlands

Summary

Human motion capture, the process of recording people's movements, contributes to kinematics research, medical rehabilitation, augmented reality, meanwhile commercially succeeds in video game development, the film-making industry, etc. The captured information is utilized to animate 2-D or 3-D character models. Our work focuses on the inertial motion tracking system composed of miniature inertial sensors, biomechanical models and sensor fusion algorithms. An inertial measurement unit(IMU) consists of an accelerometer and a gyroscope, and some include a magnetometer as well. It is highly appreciated in motion tracking for being affordable, trusty and energy-efficient, and has been developed since the early 1930s. The common wearable motion capture set contains several wireless IMUs allowing users to receive real-time data after installation.

However, the current installation process of the wearable full-body IMUs set is inefficient and troubled by human-made errors. Because the sensor-to-segment placement and alignment are crucial to the reliability and informativeness of the recording, the wearable motion trackers should be placed on a predefined location with a specific orientation concerning the segment. Assigning the numbered IMUs to their corresponding can be time-consuming and prone to error when 20 IMUs, for example, are involved. Referring to Xsens MTw Awinda used in this project, attaching all the 17 IMUs on corresponding segments takes 250 seconds, yet random assignment without restrictions on sensors pairing with specific body segments takes 180 seconds. On the one hand, incorrect placement due to unintentional human error will directly lead to the failure of the visualization (twisted and misplaced body parts, etc.) of sensor data in the supporting software. On the other hand, if the calibration is passed because the switched IMUs, for example, on the left and right shoulder, are kinematically similar, the following recording task is meaningless based on the wrong labelled data source. What's more, the mislabelled data are difficult to be detected without referring to the previously used hardware.

Therefore, to optimize the current installation process, automatic IMU-to-Segment (I2S) assignment methods based on recorded inertial data are proposed. The traditional methods take advantage of a large number of manually-selected features and shallow machine learning methods. However, shallow machine learning meth-

ods have some common shortcomings: a. Manually selecting features is time-consuming; b. The feature set is case-by-case and subjective, requiring prior knowledge; c. The commonly used features, like the magnitude of acceleration changing from individual to individual, are not discriminative enough which causes poor robustness of the shallow machine learning-based models. In this case, deep learning methods, which learn the features directly from the data, are considered to address the mentioned problems. Some previous research is done to minimize the manual labour in the installation stage of wearing IMUs. The previous researchers successfully applied deep learning methods to the I2S assignment task. But the CNN+GRU model for an arbitrary amount of IMUs is only applied to lower body configuration. Another method including PointNet and attention model extracting sensor-wise interdependencies surpasses the CNN+GRU model and works for full-body configuration as well. However, this model lacks flexibility in the number of IMUs.

In this project, we explore if convolution-based models can reduce the manual feature selection and keep the flexibility of the amount of test IMUs in the full-body I2S assignment tasks based on the acceleration, angular velocity, and rotation quaternion. The involved full-body motion capturing system from Xsens called MTw Awinda consists of 17 IMUs. Each IMU is marked by a sticker indicating its corresponding segment currently. The training dataset XsensMotion includes 69 trials on around 30 subjects, and the self-collected test set involves 30 testing trials collected on 5 subjects absent from XsensMotion. To increase the prediction accuracy, we also apply data processing methods including heading correction, walking motion filter on the dataset. The long trials are sliced into shorter sequences of 2 seconds using the sliding window method. The proposed model involves three convolutional layers, one GRU layer, and three linear layers. Dissimilar hyper-parameter settings in convolutional layers are designed to realize hierarchical feature merging. Besides the input (acceleration and angular velocity) usually involved in existing approaches, the effectiveness of rotation quaternion is explored in this project, which to the best of our knowledge has not been done by previous researchers. The biased trials and special segments are specifically studied in this project. It has been proved that hierachical feature merging model with the walking motion filter has the best performance among the mentioned models with all three configurations. The achieved performance is also comparable to the previous research without losing the system flexibility. Adding rotation quaternion in input or heading correction can neither contribute to the overall performance. To enhance the performance, we apply majority voting on predictions based on sliced windows to generate one final label for the whole trial. The trial-wise performance on lower-body configuration (left and right foot upper leg, lower leg, and pelvis) achieves 100% accuracy using the proposed model.

Contents

List of acronyms

IMU	Inertial Measurement Unit
DNN	Deep Neural Network
HAR	Human Activity Recognition
RNN	Recurrent Neural Network
I2S	IMUs-To-Segment
CNN	Convolutional Neural Network
TSC	Time Series Classification
FCN	Fully Convolutional Network
ResNet	Residual Network
LSTM	Long Short-Term Memory
GRU	Generalized Recurrent Unit
NN	Nearest Neighbour
DTW	Dynamic Time Warping
BOSS	Bag-of-SFA-Symbols
TSF	Time Series Forest
ROCKET	Random Convolutional Kernel Transform
MLP	Multilayer Perceptron
GVCNN	Group-View Convolutional Neural Network
WMF	Walking Motion Filter
HCFM	Hierarchical Convolutional Feature Merging

Chapter 1

Introduction

1.1 Background

Human motion capture, the process of recording people's movements, contributes to kinematics research [1], medical rehabilitation [2], augmented reality [3], and also commercially succeeds in video game development, film-making industry, etc. The captured information is usually used to animate 2-D or 3-D character models. The capturing systems can be classified as optical and non-optical systems. Optical systems consist of methods using makers and special camera sensors or marker-less algorithms based on computer vision. The non-optical system tracks the body motion by using inertial, magnetic, stretch sensors or directly measuring the joint angles between two segments.

This work focuses on the inertial motion tracking system which is composed of miniature inertial sensors, biomechanical models and sensor fusion algorithms [4]. Inertial Measurement Unit (IMU) which consists of an accelerometer, gyroscope and some also includes a magnetometer, is highly appreciated in motion tracking for its feature of being affordable, trusty and energy-efficient, which has been hugely developed since the early 1930s. A common wearable motion capture set contains several wireless IMUs allowing users to receive real-time data after the process of installation. The IMUs are fixed on the body using elastic straps, head straps and wrister depending on the segment, as shown in Figure 1.1.

However, the current installation process of the wearable full-body IMUs set is not the most efficient and is prone to human-made errors. To begin with, because the sensor-to-segment placement and alignment are crucial to the reliability and informativeness of the recording [5], [6], and the result of the sensor data could be influenced by the unexpected changes in the sensor placements among different subjects, the installation usually requires expert guidance throughout. Moreover, with a multi-IMU configuration, it is obligatory to know which segment the sensor is on to accurately record data and reflect them on the corresponding part in the

digital character model. Assigning the numbered IMUs to their corresponding can be time-consuming and prone to error [7], [8] when 20 IMUs [9], for example, are involved. Referring to Xsens MTw Awinda, attaching all the 17 IMUs on certain segments takes 250 seconds, while random assignment without restrictions on sensors pairing with specific body segments takes 180 seconds. Wrong placement due to unintentional human error will directly lead to the failure of the visualization(twisted and misplaced body parts, etc.) of sensor data in the supporting software. The following recording task can not continue, so the calibration has to be carried out again from the beginning until the segments are correctly assigned. Moreover, it is possible that the data from misplaced IMUs successfully generate a kinematic model if they do not have too different motions, for example, IMUs on the left and right shoulder. For Xsens MTw Awinda, as long as the data and visualization are archived, it will be difficult to detect the problem by checking the animation, unless comparing the log file with the unique number on the motion tracker. It will be harmful if those incorrect data are regarded as correct ones and used in research like medical rehabilitation.

Therefore, to optimize the current installation stage of the wearable IMUs set, automatic I2S assignment methods based on recorded inertial data are proposed [7], [8], [10]. In the field of Human Activity Recognition (HAR), studying the location of the sensor is also important because the movement of sensors provides information to HAR on one hand, but also interferes with the data collection, on the other hand, depending on the type of the movements. Possible movements of the sensors include three types: (1) on-body movement from hand to back pocket, (2) with-body

(a) A subject with 17 IMUs on different body segments.

(b) The wrister and strap used to attach the IMUs to the body.

(c) Each IMU has a sticker on it to indicate the respective segment.

Figure 1.1: Some plots of using Xsens MTw Awinda(one of the wearable IMU sets) in real life.

movement(unexpected shaking in the pocket, etc.), and (3) orientation changes according to previous research [11]. Researchers have to use location-independent features to get rid of the influence caused by unwanted movements or figure out the sensor placement to take advantage of the useful movements by using the acceleration signal alone [12], some with the help of proximity and light sensors [13]. The traditional methods, as mentioned above, take advantage of a large number of features of different segments, using shallow machine learning methods to recognize the IMUs placement based on manually selected features. However, the shallow machine learning methods have some common shortcomings: first, manually selecting features is time-consuming [7]; second, the selected set of features is case-by-case and subjective, requiring prior knowledge [11]; third the commonly used features, like the magnitude of acceleration changing from individual to individual, are not discriminative enough. According to research on gait recognition for human identification [14], the walking gaits can be so different from person to person which makes the shallow machine learning-based models less robust.

In this case, deep learning methods, which learn the features directly from the data, are considered to address the mentioned problems. Some previous research is done to minimize the manual labour in the installation stage of wearing IMUs. The previous researchers successfully applied deep learning methods to the I2S assignment task. The Deep Neural Network (DNN) performs so well in many fields, it also achieved 98.57% accuracy using Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) for the lower body I2S assignment task [7]. Acceleration and gyroscope data are used to form a 6-channel input matrix. Later in 2021, Kaichi et al. proposed a model which combines attention encoder [15] and IMU-wise global feature in PointNet model [16], with a root sensor, the model achieves 93.1% on full-body configuration involving 15 IMUs on Mo-cap dataset [17].

I2S placement classification, which is based on data with ordering notions, belongs to Time Series Classification (TSC) problems [18]–[20]. Many TSC algorithms have been presented in the last decade [21]. Pieces of research done within the field of TSC successfully achieve higher or equal performance using deep learning methods, compared to the ones using shallow machine learning methods. There are some deep learning models that have been proven to work well used in discriminative end-to-end approaches including Fully Convolutional Network (FCN) [22], Encoder [23], Residual Network (ResNet) [18], [24]. Methods aiming at automatic pattern learning without using DNNs are also proposed. These methods extract features from time-series samples and use the features to train a classifier. For instance, Random Convolutional Kernel Transform (ROCKET) uses thousands of variant kernels to learn different features and then gives the features to a linear classifier. Technically speaking, it is not a deep learning method because the convolutional ker-

nels are not being used in typical CNNs. The huge number of convolutional kernels, in combination, captures the informative patterns and the linear classifier keeps the computational complexity low, which together guarantees the state-of-art classification accuracy on a large dataset [25]. Though these models have not been applied to the I2S assignment problem, their performance on the related tasks shows the potential for this problem.

1.2 Motivation

To sum up, the current I2S labelling has inefficient installation steps and is prone to human-made errors which not only directly cause the failure of calibration on the other hand but also result in incorrect kinematic data with a high cost to be detected.

This work aims to automatically determine the I2S placement with the full-body configuration(17 IMUs) using proper convolution-based methods to minimize the manual operation(feature engineering, etc.) based on the acceleration, angular velocity, and generated orientation information with flexibility in the IMU number. In another word, the place of the IMU sensor is automatically labelled according to the inertial data collected by the sensors. For that reason, subjects' attention on which sensor to which segment is not needed anymore. The initialization stage will be less time-consuming and leave fewer human-made mistakes.

To realize this goal, there are several important issues. Firstly, we need to minimize the manual work. This is addressed by using end-to-end deep learning methods. Based on the high accuracy achieved by CNN+Generalized Recurrent Unit (GRU) model on I2S assignment tasks [7], [10] and other TSC tasks, we would evaluate its performance of it here. Additionally, we found that different hyperparameter settings of each CNN layer realizing feature merging step by step is possible to enhance the performance of the traditional CNN+GRU model, the comparison will be conducted between these two models.

Secondly, the calibration of the current wearable IMU set includes two parts: around 5 seconds' N-pose (standing still with the feet shoulder-width apart and hands on the sides of the body, looking straight ahead) and more than 10 seconds' walking motion. Inspired by previous work [26], we would like to evaluate the effectiveness of WMF, which extracts only the walking motion in a trial. Because the composition of calibration is simple and can be easily separated into N-pose and walking by detecting the frame with a sudden increase in the absolute value of the acceleration.

Thirdly, most studies have relied on accelerations and angular velocities, but orientation information–rotation quaternion has not been used to solve I2S assignment before. So we are going to explore their efficiency in this project.

Fourthly, ideally, we want the IMU assignment system to remain unaffected by a single IMU out of power, loss, or malfunction. In this case, the system should be flexible as well. Flexibility here refers to the ability of a model to work properly with different numbers of IMUs. However, according to Kaichi et al. [10] and Zimmermann et al. [7], keeping a balance between flexibility and accuracy of the model is not easy. Zimmermanna et al. ensure flexibility by using a one-by-one method, which means the model can recognize a single sensor's placement. The model proposed by Kaichi et al., taking advantage of the interdependency between IMUs, outperforms Zimmermanna et al.'s model. However, it brings a negative impact to the flexibility, because the model requires all the IMUs' inertial data to predict a single IMU's placement. So we proposed a model combining the root IMU with the one-by-one method. Root IMU is used as the reference for the other IMUs. After some proper data preprocessing, the model can learn the interdependencies between one root IMU and the other non-root IMUs, without adding more restrictions to the IMU configuration. Based on the test experiment and the previous work, we find that the long trials in I2S assignment are usually sliced into shorter sequences and the accuracy rate related to those sequences is commonly higher than 50%. Thus we would like to see if majority voting can improve the performance without hurting the flexibility in IMU number in the test phase.

Finally, previous work has also proved that IMUs on certain body segments are harder to recognize. For example, upper body recognition is harder than lower body [8], [10]. So the experiments will be implemented with three body configurations including lower body, upper body, and full body, which makes it possible to train different models for each body part, and specific analysis can be done on finer granularity.

1.3 Research question

Based on the motivation, here is the main research question:

Can convolution-based models reduce the manual feature selection and keep the flexibility of the amount of test Inertial Measurement Unit(IMU) in the I2S assignment tasks based on the acceleration, angular velocity, and rotation quaternion?

To be specific, it consists of several sub-questions as follows:

RQ1: What is the performance of HCFM deep learning model compared to the baseline model [7] Ttrained and tested on our dataset?

RQ2: What is the performance of the model trained and tested only with the walking motion (by using the WMF)?

RQ3: What is the performance of the model using rotation quaternion directly as a part of the input or as a necessary element to realize heading correction in deep

I2S assigning models?

RQ4: How much improvement can be brought to the performance quantitatively by majority voting?

1.4 Report organization

More related work will be discussed and compared in Chapter 2. In Chapter 3, the instrumentation, dataset information, data processing methods, neural network architecture, heading correction, and evaluation methods are introduced. Then, in Chapter 4, the results of the models with distinct settings and input are shown. Discussion will be stated in Chapter 5. Finally, in Chapter 6, conclusions and future work are given.

Chapter 2

Related work

This section will introduce the work related to the I2S assignment. The first part introduces the I2S assignment designed to optimize the calibration of the joint kinematic data recording(Section 2.1). Because there is not much previous work done exactly on the I2S assignment tasks, we studied other sensor placement recognition tasks concerning HAR as shown in Section 2.2. They are regarded as related work because a lot of inertial-data based HAR tasks have a similar process as the IMU placement recognition tasks. They are both the classification task based on time series input including but not limited to accelerations and angular velocities. Additionally, the sensor placement recognition methods mentioned in the prior two sections can be divided into shallow machine learning methods coupled with manually designed features, and DNNs used to extract discriminative methods in more recent works. This shift is similar to what happened in the TSC field, motivated by the prosperity of DNN methods and the increasing size of the available dataset [27]. Therefore, to keep this section concise, the related work of sensor placement recognition will be introduced from the perspective of usage, without repeating the differences resulting from the involved machine learning models. Section 2.3 introduces the existing TSC models using different machine learning methods from a broader perspective because the I2S assignment is regarded as one of the TSC tasks. Section 2.3.1 lists classic methods, and Section 2.3.2 discusses some models that use CNN and RNN related models. The characteristics of the task itself and the dataset used contribute to the selection of the related models. Figure 2.1 shows the relationship between the mentioned research topics. The most relevant sensor placement recognition tasks are summarized in Table 2.1. Finally, Section 2.4 summarizes the limitations of the previous research on sensor placement tasks.

It is worth noticing that the performance of the models depends on the involved subjects, the recorded motions, the dataset size, etc. Therefore the meaning of the accuracy of the listed model for our project is limited. The figures are regarded as referential information instead of specific baselines.

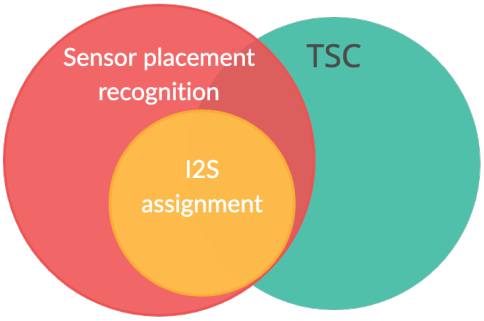


Figure 2.1: The relationship between I2S assignment, sensor placement recognition and TSC.

2.1 I2S assignment and the Calibration Stage of Inertial Motion Capture

I2S placement is the starting point and basis of the calibration stage for all tasks as long as they involve IMU-based motion tracking of the human body or specific segments [7], [8], [10], [28]. Nowadays, more researchers start to focus on optimizing the installation procedure of inertial motion capture. The range of sensor types is narrowed down to the IMU here. IMU placement recognition, or automatic I2S assignment, as a way to reduce human error and time spent on the calibration process, especially under a multi-sensor situation, is studied in a more precise context with many prerequisites. For example, the IMU positions on specific segments and the I2S orientations are usually pre-defined [29], and a guaranteed amount of effort is spent to ensure that the placement and orientation of the IMU in different trials are consistent.

Weenk et al. are the first to study automatic inertial sensor localization for the realistic scenario of equipping with the wearable IMUs set [8]. The researchers use the decision tree based on the c4.5 algorithm. 57 features based on acceleration and angular velocity are manually selected and functionally ranked. Hierarchical methods are proposed, which consist of segment identification, left and right upper arm and upper leg identification left and right identification for the rest parts. Strict prerequisites like subjects are asked to walk at a specific speed (around 5km/h), full-body configuration, and mandatory coordinate frame transformation are adopted, which results in weak robustness. For example, the system is not suitable for some groups of people with disability(wheelchair users, etc.), and it is not able to deal with the problem that some of the sensors are out of usage. For full-body configuration, 97.5% of the sensors are correctly classified and for the lower body, the accuracy is

100%. It is worth mentioning that a trial length of around 6 seconds results in the best performance while an increase makes no progress.

Later in 2018, Zimmermann et al. applied DNNs on this real-life I2S assignment task for the first time [7], inspired by DeepConvLSTM [30]. The researchers combine CNNs for Long Short-Term Memory (LSTM) recurrent networks, generalized recurrent units GRU for time dynamic features extraction as well to recognize the IMU placement out of 7 locations on the lower body as shown in Figure 2.2. 3-axis acceleration and angular velocity data are concatenated as the 6-channel input, without direct feature calculation on the dataset. They achieved 98.57% average accuracy and 100% if the side differences are ignored. Automatic I2S alignment is also studied in this work using regression models. Automatic I2S assignment on

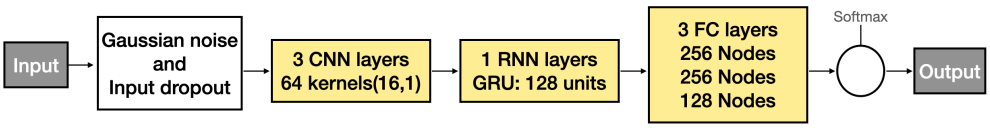


Figure 2.2: Overview of the network configurations proposed by Zimmermann et al. [7].

full-body scale approach is also implemented [10]. This paper proposes a combined deep learning model to automatically extract the sensor interdependencies. Firstly, the CNN-GRU module learns the local features of each IMU using shared weights. Then, inspired by a prior work Pointnet [16], a max-pooling layer is applied to extract the sensor-wise global feature which is concatenated to the local feature later as the input of the following model. Finally, the transformer encoder [15] module learns from the merged features and ends up with classification scores through a linear transformation with a softmax activation function. For IMU configuration, a root IMU(located on the lower back) is used in this work and it does not include IMUs on shoulders, which are very vulnerable in I2S assignments. The user can mount IMUs at a random angle, differing from the previous strict orienting rule. This piece of research also compares the performance with distinct deep learning module settings, and different sensor placements(lower, upper, or full body), on two independent datasets [17], [31]. It is said that their model over-performs the one-by-one method proposed by Zimmermann et al. [7] on the same datasets mentioned above.

2.2 Sensor Placement Recognition and HAR

More generally, locating sensors has been a way to assist HAR for a long time. HAR is an important task for both the academic world and industry due to its wide applicability to health monitoring, robotics, human-computer interaction(HCI) and sports science [32]. Sensor-based activity recognition can be divided into three types: ones using IMUs [33]–[35], camera [36] or hybrid configuration. The mentioned works use sensor signals directly training a model or extracting features to conclude the human activity without knowledge of the location of the sensors. However, human activity recognition and sensor position recognition are not always independent of each other. Conversely, the location of sensors serves as an aid to activity detection [12], [37]–[41]. More specifically speaking, locating the phones to head, pocket, chest or hand implies possible activities like calling, walking sitting and so on which allows automatic determination of suitable corresponding modes of the phone, as an example [13]. Tracking the location of the sensor also reduces the chance of misusing ambulatory monitors and increases system reliability [42], [43].

In most cases, inertial sensors are sufficient for sensor placement recognition, and particularly, acceleration is the primary data source. Research purely based on non-inertial information(ambient sounds [44], etc.) will not be discussed in this paper. In 2006, a group solely used accelerator signals to determine whether sensors were on the head, wrist, breast or trouser's pocket simulating activities in normal daily life [12]. This paper is based on the norm of acceleration vector to prevent the influence of sensor orientation. It focuses on walking for its distinct motion signature by recognizing data frames of 1s before sensor placement recognition. The best result is achieved by C 4.5 classifier: 100% recognition rate for event-based recognition and 94% for frame-by-frame recognition, after filtering out the non-walking windows and data smoothing. Later in 2007, a group used a similar model(C4.5 classifier), 6 best features out of 35 acceleration signatures to classify the sensor placement into 5 classes [37], but on an everyday activities dataset. They also compared two different sizes of the sliding window. The maximum accuracy is 80%, and if left and right trousers' pockets are combined into one class the accuracy increases to 92%. Amini et al. make use of acceleration of 10 sensors combined with an SVM (support vector machine) classifier to determine the on-body sensor locations and achieves an accuracy of 89% [41]. It is noteworthy that the subjects(25 in total) in this study are told no instruction regarding the exact placement and the orientation of the sensors. Lambrecht et al. use a simple classifier for recognizing 17 tasks while the subjects are patients suffering from essential tremor or Parkinson's disease [40]. It focuses on upper limb configuration and 18 candidate features are selected according to previous research on clinical rehabilitation involving poststroke

patients [45].

Some researchers consider other sensors to assist the placement recognition. Grokop et al. exploit accelerators and light sensors to classify the motion state into one of 6 categories and device position is classified into one of 7 categories, which achieves a macro-averaged F-score of 66.8% for device position recognition [13].

	year	**Recognized sensor sites (total number)**	**Dataset information (Activities; subjects; length)**	**Selected Features (source)**	**Classif-ication methods**	**validation method**	**Accuracy**
[13]	2006	Head, left trousers' pocket, left breast pocket, wrist. (4)	walking and other daily activities; 6 subjects; 216 to 270 min	6 features comfirmed by initial test (Acceleration)	C4.5	10-fold cross-validation	walking motion: 94% (89.81% without smoothed jumping window)
[37]	2007	Head, front trousers' pocket, back trousers' pocket, torso, wrist. (5)	everyday activities; 3 subjects; 9 hours	6 features out of 35 (Acceleration)	HMM	Not disclosed	83%
[41]	2011	forearm$\times 2$, upper arm$\times 2$, shin$\times 2$,head, thigh$\times 2$ waist. (10)	daily activities; 25 subjects; 750 min	5 features for walking, 1 features for non-walk, (Acceleration)	SVM	80% as validation set	89%
[8]	2013	shoulder$\times 2$, upper arm$\times 2$, forearm$\times 2$, hand$\times 2$, upper leg$\times 2$, lower leg$\times 2$, foot$\times 2$ pelvis, sternum, head. (17)	walking; 11 subjects; 35 trials (6 seconds each)	57 features (Acceleration, angular velocitie, angular acceleration)	C4.5	10-fold cross-validation	97.5% (full-body) 100% (lower-body)

[40]	2014	hand, distal forearm, proximal forearm, distal humerus. (4)	17 daily tasks; 13 subjects (affected by tremor and Parkinson disease); 40 seconds per trial	10 features (Acceleration)	simple classifier based on feature rankings	Not disclosed	91.77%
[38]	2015	arm, ankle thigh, hip wrist (5)	28 daily activities 33 subjects; around 2000 min	8 features (Acceleration)	SVM	LOSO	91.2%
[28]	2016	Thigh$\times 2$, shank$\times 2$, foot$\times 2$ (6)	17 daily tasks; healthy subjects and Parkinson disease patients (the number is not disclosed); 500 trial, each for 3 seconds	10 features (Acceleration)	case-depended method.	Not disclosed	99.7%
[7]	2018	Upper leg$\times 2$, lower leg$\times 2$, foot$\times 2$ pelvis. (7)	A: CMU-MoCap [17]; B: 4 subjects, walking; C: 28 subjects, walking.	No manually selected features (Acceleration gyroscope)	CNN + GRU	LOSO	98.57%
[10]	2021	lower body (7); upper body (9); full body (15). Lower back is the root sensor	43 subjects from the simulated dataset CMU-MoCap [17] 5 subjects from Total-Capture: [31]	No manually selected features (Acceleration, gyroscope)	deep learning method: PointCloud and Encoder	partial validation	full body: 93.1% on CMU-MoCap; 91.6% on Total-Capture

Table 2.1: The mentioned related works involve sensor placement recognition

2.3 Time Series Classification

Within the past years, time series classification became an important topic in data mining and researchers proposed hundreds of time series classification algorithms [21]. According to the previous definition, [18], the TSC tasks aim to map from the possible univariate time series input into a probability distribution vector indicating the prediction among N class labels.

2.3.1 Traditional Time Series Classification Methods

Apart from the mentioned shallow machine learning methods like SVM, and decision tree, which require manual empirical selection of features with non-negligible efforts, there are some more general methods in the field of TSC. Though no DNNs are involved, these methods achieve better results through an ensemble of different models and design special data processing algorithms and distance functions to reduce the need for insight into the data and inevitable manual labour.

The Nearest Neighbour (NN) classifier is one of the most popular classic TSC approaches. To use NN classifiers, the selection of distance function by the feature of the data is necessary. Dynamic Time Warping (DTW) is proved that outperforms or is equal to other distance functions [46]. Ensembling NN classifiers using different distance functions performs well, too.

The further study focuses on ensembling classifiers: dictionary-based Bag-of-SFA-Symbols (BOSS) [47] is trained on an ensemble of NNs classifiers, using a representation of patterns occurrence frequency; Time Series Forest (TSF) ensembles random forest [48]; COTE [49], is designed to address the problem that each single TS transformation algorithm(shapelets transform [21], etc.) is suitable for its specific data type, not standing over the others, consisting of 35 other existing classifiers like BOSS. Though COTE and the extended version HIVE-COTE are considered state-of-art algorithms, the intensity in computation is not a trivial thing that results in infeasibility under some circumstances [50].

It is worth noting that, unlike one-class time series classification, the data used for sensor-to-segment placement recognition are usually walking records. A complete walking cycle repeats an indefinite number of times, which means that the length of the entire time series is not important. Because each step of the subject can be viewed as a complete sub-series, whether the time series is properly segmented into each step may seriously affect the performance of these traditional methods. At the same time, considering the time complexity of these models and the diversity of raw data used (including clapping, turning, inconsistent length, etc.), traditional time series classification methods won't be our first choice.

2.3.2 CNN and RNN for Time Series Classification

Although there are hundreds of DNNs, only a part of them are proved efficient in TSC problems like Multilayer Perceptron (MLP), CNNs and Echo State Network [18]. This project empirically focuses on the convolutional-layer-related models due to their success in both TSC and sensor-to-segment assignment.

CNNs are famous for exploiting the locality in the data. The development in image recognition tasks, like image classification, has led to impressive performances [51]. Recent research has proved that it also plays an important role in TSC too [18], [19]. The idea is that 1-d time series data have a similar topology as images, only one dimensionless, which points to the effectiveness of applying CNNs in TSC tasks [52]. The layers of CNNs consist of three main classes: convolution layers, pooling layers and fully connected layers. Convolution layers use kernels to generate the feature maps, pooling layers decrease the spacial dimensions of the representation of the images and fully connected layers fulfil the function of the classification. Combining several groups of convolution layer and pooling layer sequentially, the higher-level features can be extracted and the trivial details would be ignored, being less extreme. Compared to the traditional TSC methods, CNNs make it possible to get rid of the hand-crafted domain-specific features that required expert knowledge.

ROCKET [25] integrates random convolutional kernels and linear classifiers, addressing the problem of the lavish computational expenses of the existing methods. ROCKET is not a type of deep learning model but is inspired by the effectiveness of CNN. It trains no neural networks. Instead, it uses the extracted features to train a simple machine learning model. It achieves state-of-art accuracy on the UCR archive [53]. Thousands of kernels with different lengths, weights, bias, dilation and padding are used to transform the time series into a considerable number of features to train the linear model.

The combination of CNNs and other deep learning models also obtains great success in this topic. DeepConvLSTM [30] combines CNNs to learn time series' stable features and the LSTM layer to learn sensor signals' temporal dynamics. Outperformance compared to baseline CNNs is observed in experiments on all datasets [30], [54]. Further research also uses GRUs to replace the LSTM layer showing better performance on sensor placement classification tasks than LSTM with comparably fewer parameters [7].

The mentioned gated method GRU and architecture LSTM are parts of the concept of RNN. The fact is that RNNs have drawbacks referring to TSC because of three main points: (1) RNNs architectures are designed to predict output for each timestamp, like instant translation as one of the applications; (2) Vanishing gradient is an essential problem, especially in long time series; (3) Computational complexity

of RNNs makes it a harder choice for researches [55], [56].

Considering that Kaichi et al. [10] successfully applied some deep learning models not being one of the main methods in TSC task [18], other groundbreaking models in computer vision and natural language processing like Group-View Convolutional Neural Network (GVCNN) [57] and PointNet [16] are potential to achieve good performance as well.

2.4 Limitations of the Previous Studies

However, we argue that previous literature suffers from certain weaknesses.

Firstly, all of the research mentioned using task-based classifiers or shallow machine learning methods have a common characteristic of the non-negligible amount of work spent on sorting out pertinent features, and it my be subjective and empirical. Leonard H. Grokop et al. select 6 features by initial test [13]; Kai Kunze and Paul Lukowicz propose a model based on 6 features out of 35 features [11]; 18 candidate features are selected in Stefan Lambrecht et al.'s sensor location identification task, and they have not explained why some of the features are chosen, like the sum of the maximum between acceleration among three axes [45]. Many pieces of research have discussed the lack of structural instruction and general standards of feature extraction [11]. There is an automatic feature extraction tool designed to speed up and simplify the process of feature extraction [58]. Genetic programming-based feature selection algorithms are proposed by Kilian Forste et al. [59]. Their objective is to enlarge the space of features and aim to increase the possibility of finding good features based on genetic programming, which should fulfil two basic goals: One can discriminate activity classes; The second is robust enough, especially for on-body sensor variation. However, the tools and algorithms designed to help feature engineering can not solve the current problems completely. The automated feature extraction tool can speed up the process, but prior and expert knowledge are still required to filter out the features. Kilian Forste et al.'s generated features indeed outperform the standard features. However, the generated features depend on the manually selected feature set and combination function set, which could be subjective.

Secondly, the proposed methods are case-dependent and lack robustness. For example, Weenk et al.'s method uses a decision tree, a shallow machine learning method, to determine the segment. The model achieves an accuracy of 100% with lower body configuration (8 segments in total. 7 are the same as those lower-body segments defined in this project in Section 3.3, the rest is on sternum), but it is based on a hierarchical decision process. The decision tree includes rules like "classifying the data point by the value of root mean square of the magnitude of

the accelerations($RMS(|a|)$)". We explored our dataset and find the $RMS(|a|)$ of the same segment among trials varies a lot which means the decision tree is less effective for new subjects.

Thirdly, the number of segments involved in the previous research is small. As shown in Table 2.1, most of the research use less than ten sensors on a part of the body. The full-body configuration is used in only two papers [8], [10]. The work uses a full-body configuration that can place the sensors on different body segments. The performances on different segments are dissimilar according to results on full-body configuration. Therefore, the differences in the configuration of each task add to the uncertainty of our task.

Fourthly, only a few works in literature study the flexibility in the IMU number of the model. If the model has flexibility in IMU number, it works properly when there are fewer IMUs in the testing stage than in the training stage. Other than the drawbacks of re-usability of the methods on another dataset, Weenk et al. have mentioned that the system they designed can only be used with full-body configuration because the decision tree includes rules concerning multiple IMUs, like correlation efficiency and the rank [7]. When the configuration is unknown, only 75.9% of the IMUs are identified correctly, because the mentioned multi-sensor features are not used. Though previous research has proved the effectiveness of the deep learning model on I2S assignment tasks, some key questions and notions are still not discussed in the literature. The usage of sensor-wise interdependencies as demonstrated by Kaichi et al. [10] yields better results than the single sensor method which uses no connection between IMUs [7], but it also harms the flexibility. Kaichi et al.'s model consists of the pooling aggregator extracting the global feature and the transformer learning the dependency between every two sensors, and a root sensor mechanism to eliminate the influence of walking directions. Therefore, the prediction depends on all the other IMUs, which are vulnerable to sporadic IMU dysfunction. New methods are required to keep the balance between the system flexibility in IMU number and the accuracy of segment recognition.

Chapter 3

Methods

3.1 Instrumentation

The available training dataset XsensMotion and newly collected test data are recorded using the full-body motion capture system called MVN Awinda from Xsens (Xsens Technologies B.V., Enschede, the Netherlands) [9], [60]. The system consists of 17 IMUs, and other supporting stuff like a charger, straps, and so on. The IMUs are fixed on respective body segments and the position with straps, as well as the orientation relative to the segments, are predefined. To use the MVN, the subject is asked to put on a customized T-shirt, a headband, and a pair of gloves with pockets to attach the IMUs on the sternum, left and right shoulder, head(on the right side near the ear), and the dorsal side of hands. 8 IMUs are strapped symmetrically on right and left limbs, one on the back located between the abdomen and the legs corresponding to the pelvis. The left two IMUs are placed on the foot. The mentioned installation can be found respectively in Figure 3.1.

To record motion data, a calibration phase is designed to figure out the sensor-to-segment alignments. A standard calibration phase in this project starts with N-pose (as shown in Figure 3.3b) for around 5 seconds and then walking at a normal pace for more than 10 seconds within the wireless range up to 20 meters indoor and 50 outdoors. Our trials replicate this calibration phase in the MVN installation. There are no specific restrictions on the walking trajectories and speed of the subjects in the training and test dataset, but running and jumping are forbidden in this project. The motions designed for the testing set will be introduced in Section 3.3.

3.2 Coordinate Frames

In this project, we work with two coordinate systems: the local coordinate system and the sensor coordinate system. The local coordinate system is the fixed frame.

Figure 3.1: From left to right, then up to down, the subplot is for IMU(s) on the sternum, left and right shoulder, head, hand, upper limb, lower limb, pelvis, and foot.

For example, the room the subject staying in. The sensor coordinate system is the frame of the IMUs. there is only one local coordinate system, but each of the 17 IMUs has its sensor coordinate system. There is another kind of coordinate system, the segment coordinate system as shown in Figure 3.2. In this case, the IMU is regarded as stationary relative to its segment. Since the I2S orientation is predefined, the relationship between a sensor coordinate system and its corresponding sensor coordinate system is unchanged during the recording and also among different trials. The deviation between the pre-defined and real I2S orientation as well as the position will not be considered in this project.

The local reference coordinate is defined by the magnetic information. The X-axis points to the local magnetic North; Y according to right-handed coordinates, which is the magnetic west; the z-axis is perpendicular to the x and y axes, pointing up to the sky. The definition of the sensor coordinate system is shown in Figure 3.3a. The x-axis points up from the face with a charging port to its opposite face, parallel with the longer side; the y and z axis is also defined according to the right-handed coordinate system.

The accelerator and gyroscope of the IMU measure the acceleration(including gravitational acceleration) and angular velocity respectively in the sensor frame, and the local orientation information is represented with unit quaternions.

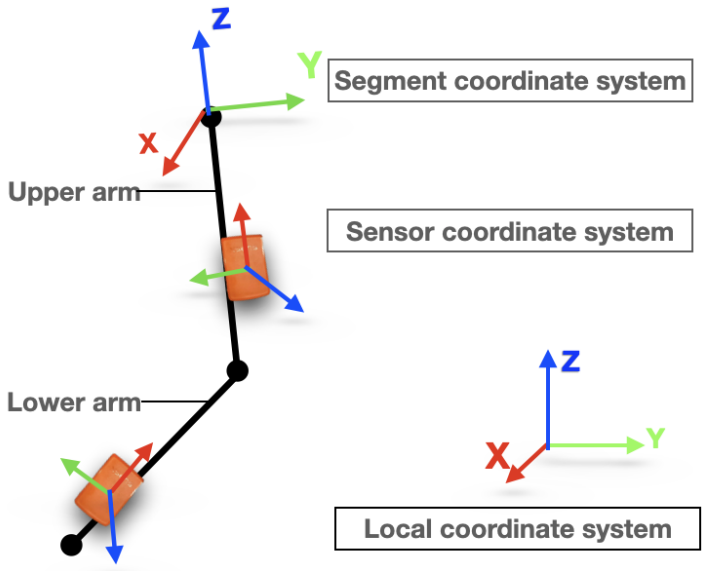
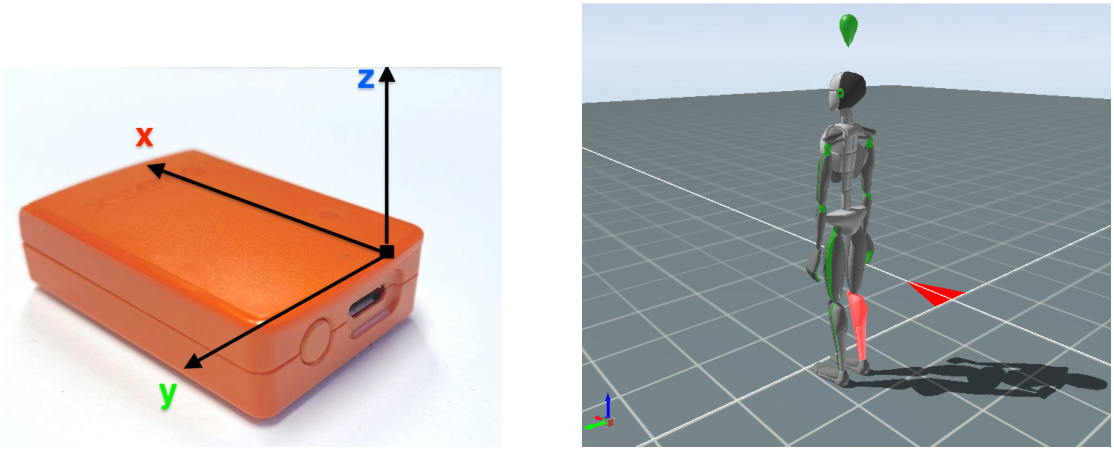


Figure 3.2: Exemplary relationship among local, segment, and sensor coordinate systems.



(a) *(b)*

Figure 3.3: (a): The sensor(IMU) reference coordinate. (b): The visualisation of a subject standing(N-pose) on the starting position concerning the local coordinate system.

3.3 Dataset

The dataset used in this project consists of two parts: the training set collected by previous researchers in Xsens and the test set collected in this project. The training set XsensMotion consists of 69 different trials with a total length of 1123.5 seconds (around 19 minutes) and an average length of 16.3 seconds. The longest trial is 55.75 seconds and the shortest is 11.2 seconds. The sampling rate of the trials is

distinct including 60Hz (23 trials) and 240Hz (46 trials).

Table 3.1: The information of the subjects in the test set.

NO	height	gender	trials info
1	177.5cm	male	3 standard trials; 1 infinity walking trial; 2 abnormal trials
2	182.5cm	male	
3	179.0cm	male	
4	169.0cm	female	
5	166.0cm	female	

The test set is recorded on participants different from those in the training set. We collect 30 trials on 5 subjects (6 trials per person) as shown in Table 3.1. The trials add up to 673.2 seconds with a size over 334MB. Each subject has 3 standard trials, 1 infinity walk trial and 2 abnormal trials. In all trials, the subject starts from 5 seconds' N-pose and then walks for 10 seconds or longer. In a standard trial, the trajectory is straight with U-turns linking the walk back and forth. In the infinity walk trial, the subject walks in a figure-eight pattern keeping on turnings to evaluate the influence of different trajectories with more turnings and fewer straight lines. In abnormal trials the subject is free to make slight movements occasionally (3-5 times in one trial) including clapping hands, picking up phones, waving at someone or scratching an itch to simulate the unexpected situation during calibration.

Each trial consists of an MVN format file with the information of 17 IMUs in MVN Awinda. The raw data without biomechanical assumptions are used as the input of the deep learning models. The orientation is represented by a four-dimensional rotation quaternion (check Appendix A for further information), three-dimensional acceleration and angular velocity, and the timestamp in seconds is taken into account. The acceleration and angular velocity in the sensor coordinate system are received from IMUs directly and the rotation quaternions within the local coordinate system are estimated using sensor fusion.

As previous research suggested, the performances of I2S assignment tasks vary a lot in different groups of involved segments. For example, during the walking motions, the upper body and lower body have diverse features which directly reflect on the prediction accuracy, which results in our idea of dividing the IMUs into upper body and lower body. Because the full body recognition is proved to be harder [8], [10], also considering the longer model training time, starting from the lower body or upper body configuration can shorten the process of determining appropriate models. This configuration setting also allows flexible model design targeting different parts of the body. The three different segment configurations involved in this project are as follows:

- Full body (17): pelvis, sternum (T8), head, right shoulder, right upper arm, right forearm, right hand, left shoulder, left upper arm, left forearm, left hand, right upper leg, right lower leg, right foot, left upper leg, left lower leg and left foot;

- Upper body (11): pelvis, sternum (T8), head, right shoulder, right upper arm, right forearm, right hand, left shoulder, left upper arm, left forearm, left hand.

- Lower body (7): pelvis, right upper leg, right lower leg, right foot, left upper leg, left lower leg and left foot.

The IMU-on-segment visualization of the mentioned body configuration can be found in Figure 3.4.

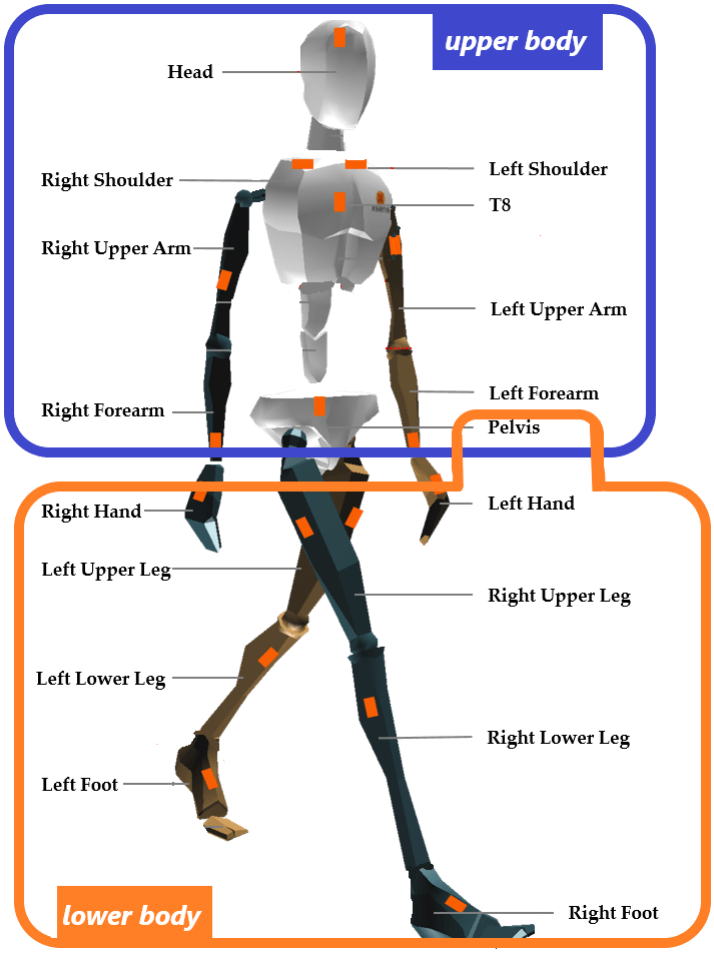


Figure 3.4: The IMUs placement on specific body segments

3.4 Experimental settings and workflow

Generally speaking, to answer the research questions, 4 types of experiments are designed in Table 3.2: 2018 model, HCFM model, HCFM model with WMF, and HCFM model with heading correction. The reasons for this design and detailed information will be introduced in the following paragraphs.

Table 3.2: The experimental design. There are two CNN+GRU models, one of which is also combined with two other data processing methods (walking motion filter and heading correction), three body configurations, and two input methods. To sum up, there are $4 \times 3 \times 2 = 24$ different models to be trained in total.

No.	Description	model	segment configuration	Input setting
Exp.1	The baseline model	2018 model	lower body upper body full body	a(6 channels): acceleration, angular velocity b(10 channels): add rotation quaternion to a
Exp.2	Distinct hyperparameters on 3 CNN layers implementing HCFM	HCFM model		
Exp.3	Apply WMF on Exp.2			
Exp.4	Apply heading correction on Exp.2			

First of all, we involve two types of CNN+GRU deep learning models to answer **RQ1**. The baseline of the task is from the model proposed by Zimmermann et al. [7] in 2018 (hereinafter referred to as the '2018 model'). To avoid the uncertainty brought by different characteristics gaits among individuals in the training dataset, we trained the 2018 model on our XsensMotion to get a comparable baseline. HCFM model is built to realize the hierarchical input merging by changing the CNN kernel inspired by Kaichi et al. [10].

Secondly, since HCFM model performs better than the 2018 model according to test experiments, we additionally applied two data processing methods namely WMF and heading correction to the former, to answer **RQ2** and **RQ3** respectively. WMF drops the N-pose at the beginning of the trials to reduce the noise in the dataset. We propose the method of heading correction using only one root sensor which keeps the flexibility of the model as well as exploits the sensor-wise relationships as a trade-off between flexibility and prediction accuracy.

Thirdly, each experiment has two kinds of input settings listed in the last column in Table 3.2 to explore the effectiveness of rotation quaternions in the input (**RQ3**).

Finally, all methods consist of three segment configurations: lower, upper and full body, since the upper body and lower body motions have significant differences

according to previous research. Separating the body into upper and lower parts allows for specific analysis matching the individual features. Additionally, it makes this work compared with the related work with similar configurations.

Based on the experimental settings, the workflow of the I2S assignment task is shown in Figure 3.5. To start with, the data processing phase is designed to down-sample part of inertial data to unify all sampling frequencies to 60Hz, and transform the CSV files into npy format. After that, the prepared time series inertial data is sliced into shorter sequences using the sliding window method. Then the "windows" are put into the CNN+GRU model in the training phase. Finally, in the evaluation phase, majority voting is applied to draw the trial-wise conclusion on the segment with more than half of the votes, based on predictions of all windows.

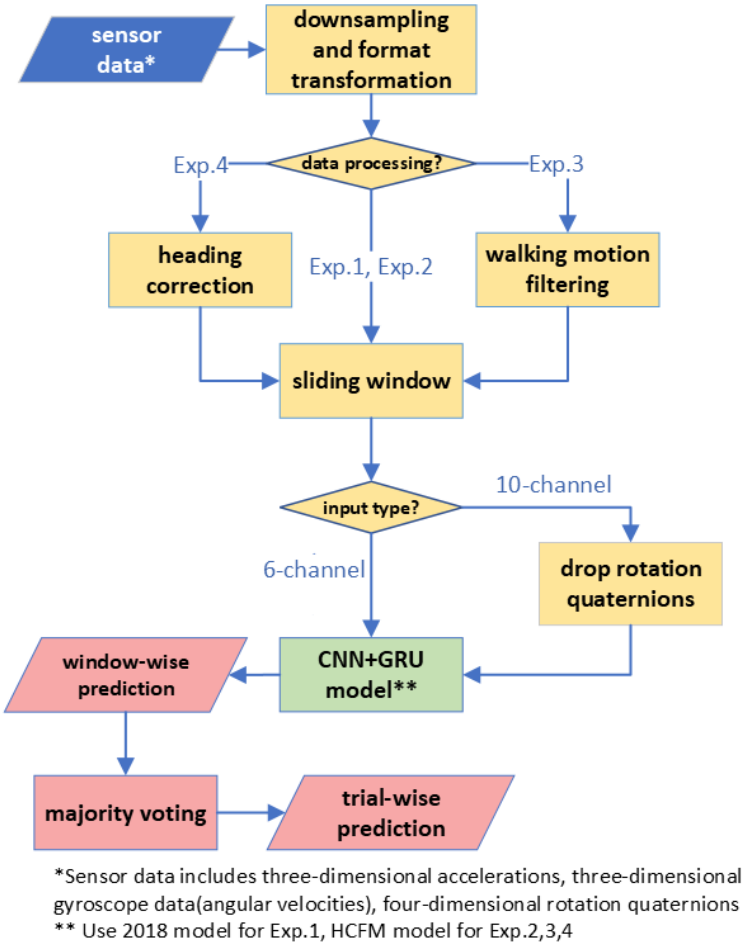


Figure 3.5: The workflow of the I2S assignment task consists of data processing phase(yellow), model training phase(green), and evaluation phase(red).

The dataflow starts from the process of the sliding window till the majority voting of the task is shown in Figure 3.6. Please note that window is a concept introduced here to stress only the number of serial frames (120 in this case). When we talk

about ”window” with the background of data processing, it represents a structure with inertial data from n stacked IMU which is just the way we organize the data; but when we discuss the window as an input unit in later sections, it includes only one sensor.

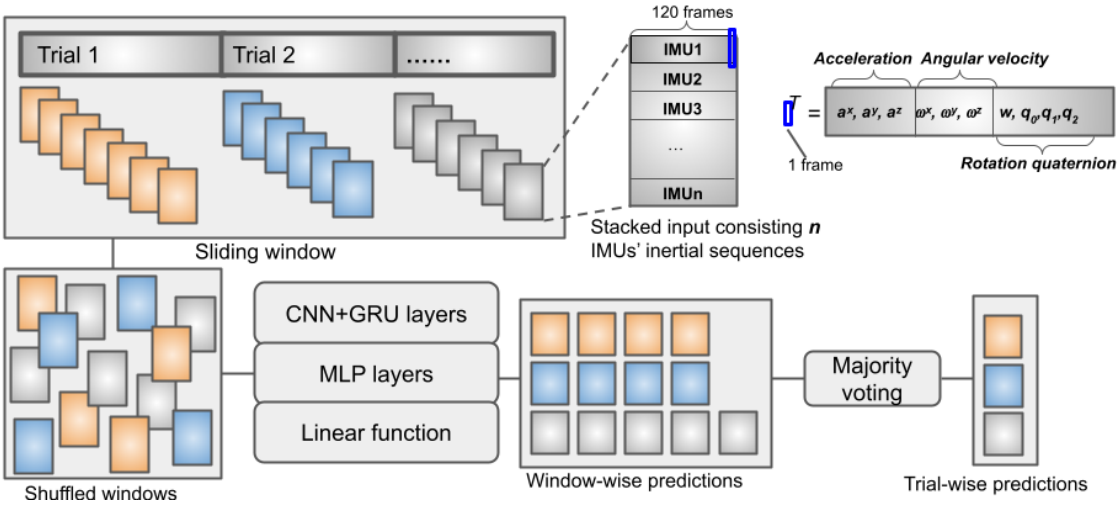


Figure 3.6: An overview of the data flow from the sliding window till majority voting. All trials include the stacked n IMUs' inertial data (acceleration and angular velocity, and also rotation quaternion if the input is 10-channel). Then the trials are sliced into shorter sequences by the sliding window with 120 frames and shuffled (mix up windows from all trials and also the order of staked sensors within the window) for the training stage before input into the deep learning models. The blue bold rectangle indicates the input composition for 1 frame: 3-D acceleration, 3-D angular velocity and 4-D rotation quaternion (if 10-channel input method id applied). For each stacked input unit, the model predicts the placement for all n IMUs. The window-wise predictions are then grouped by the trials they belong to. After majority voting, n final segments will be given to these n IMUs in one trial.

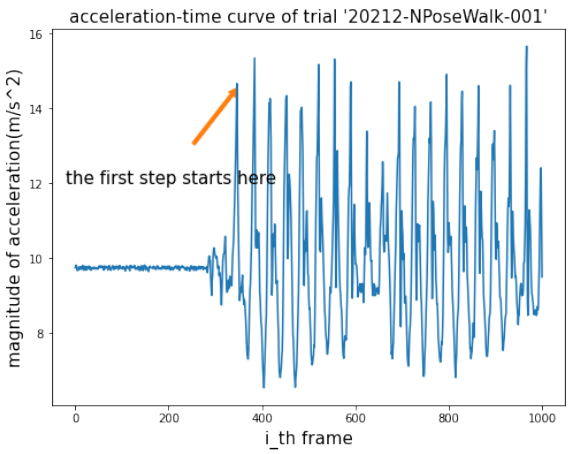
3.5 Data Processing

3.5.1 Resampling

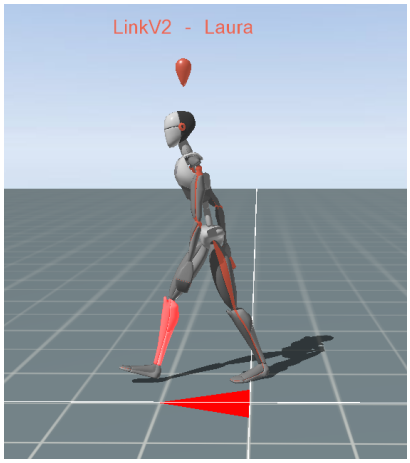
The sampling rate is the number of samples collected in one second. In the dataset, there are records of different sampling rates which requires resampling of the data. In most cases, the sampling rate is 60Hz, which is also commonly used by previous datasets [31] or research [7] related to walking motion capturing. For those data sampled using a smaller rate, upsampling or interpolation is needed; And those rates larger than 60Hz, need to be downsampled. In this project, the sampling rates of the trials will be unified to 60Hz by down-sampling the 240Hz trials.

3.5.2 Walking motion filter (WMF)

As introduced in Section 2, identifying time periods with walking motion have been discussed by many authors in literature [61]. Our dataset consists of trials including N-pose and walking motion. However, there are no strict rules on the length of N-pose or walking periods for the calibrations which leads to inconsistent n-pose lengths. Trials that do not contain n-pose phases at all are present in the dataset as well as trials that contain up to a ten-second upright stance. We are interested in the usefulness of the n-pose period to the performance of the I2S assignment, while the main data source is the walking motion. While studying the sensor data, we notice



(a)



(b)

Figure 3.7: (a): The acceleration-time curve of a certain trial. (b): The visualization of the subject's first step at the 341st frame is indicated by the orange arrow in Figure 3.7a.

that the magnitude of acceleration remains close to the gravitational acceleration value for the n-pose phase, while it fluctuates greatly with footsteps once walking began. As shown in the figure, after the 400th frame (6.7 seconds), the curve of the magnitude of the acceleration of the pelvis changes significantly due to the start of walking.

In this incident, with the considerations described above, we implemented the walking motion filter simply based on the magnitude of the three-axis acceleration. Once the absolute difference of the magnitude of the acceleration of the pelvis and the local gravitational acceleration (9.81060 m/s^2) lager than 2 m/s^2, the n-pose is regarded as terminated.

3.5.3 Heading Correction

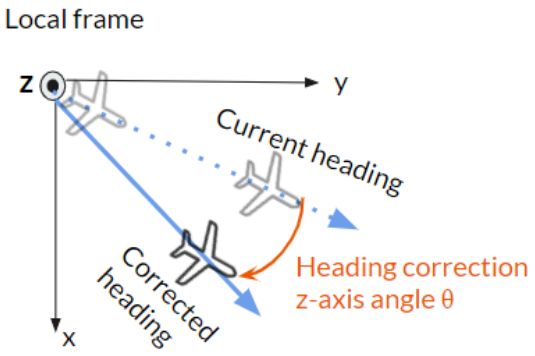


Figure 3.8: Definition of heading correction.

Heading correction is a method used in aircraft. It corrects the flight path as if the plane flies a linear course all the time as Figure 3.8 shows. In this project, the aim is to transform the trajectory on the x-y plane into a straight line. In real life, the motion tracker moves in different directions: the subject may walk circularly in a round room and straightly in the corridor, while smaller space triggers additional turns. Our idea is to eliminate the influence caused by irregular trajectory which is the thing heading correction does, thus narrowing the problem space.

However, heading correction is adopted not only because of the reason mentioned above but also because it meets our need to balance model flexibility with the utilization of IMU-wise interdependencies. The previous researchers have not intended to improve the flexibility of the I2S model as we know. The way to ensure flexibility is to sacrifice part of the IMU dependency information. After a root sensor is selected, the input is processed only according to the value of the root sensor and the matrix conversion method, and the data of other IMUs are converted into the root sensor's coordinate. Unlike Kaichi's method, the dependencies between

non-root IMUs are not considered here, only the root and non-root relationships are considered. Therefore, in this setting, only the root IMU needs to be present all the time, while Kaichi's method needs to ensure that all IMUs are equipped on the body. Here we are going to realize the root IMU system by applying the method of heading correction to all the data.

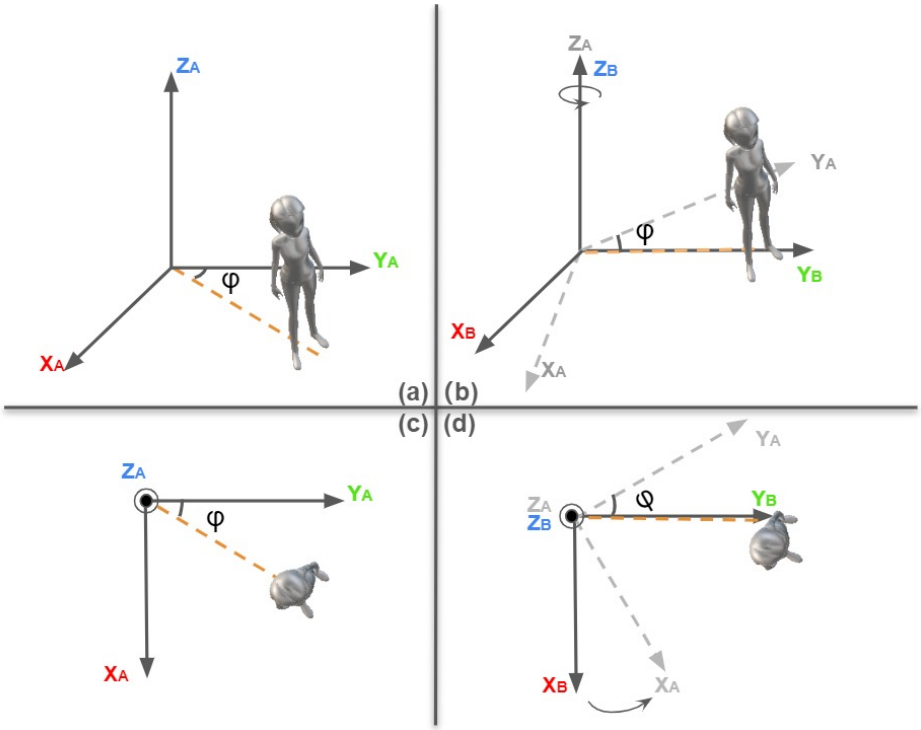


Figure 3.9: (a) shows the heading of the subject in frame Z_A. (b) shows the heading of the subject in Z_B which makes the subject's heading align to y-axis. Z_B rotating ϕ degree counter-wisely equals Z_A. (c) and (d) is the top view of (a) and (b) respectively.

To present the rotation of the subject heading, we can use Euler angles [62]. To eliminate the influence of different trajectories, we can assume the subject always walks along the y-axis(or any other direction) at the beginning. But along the walking, he or she can turn in another direction as Figure 3.9 shows. The angle between the current direction and y-axis is ϕ on the xy plane. To correct this, we can rotate the current frame A counterclockwise around z for ϕ degree which makes the subject still walking along y-axis relative to a new frame B. In this way, assume $\hat{v_A}$ is a vector (the heading of the subject here) in frame A, the coordinates for the vector in B are:

$$\hat{v}_B = R(z_\phi)\hat{v_A} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & -\cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{v_A} \tag{3.1}$$

Here, ϕ is already noted on the plot, while in this project, it equals the opposite number of the rotation angle α around z-axis. Because the rotation of the frame A to B is the inverse process of the walking deflection. α is determined by the rotation quaternion of IMU on the pelvis. To summarize, we first transform the rotation quaternion at moment T_i into the format of the Euler angle with the order of zxy (α, β, γ). Then for each IMU, we correct their accelerations $a^i \in \mathbb{R}^3$ and angular velocities $\omega^i \in \mathbb{R}^3$:

$$a^i_{correct} = R(z_{(-\alpha)})a^i \tag{3.2}$$

$$\omega^i_{correct} = R(z_{(-\alpha)})\omega^i \tag{3.3}$$

where

$$R(z_{-\alpha}) = \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) & 0 \\ \sin(-\alpha) & -\cos(-\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

We repeat the same operation for each time frame till the end of the trial. The root sensor will be dropped at the training stage as its location is already known to us.

3.5.4 Sliding window

In the TSC research, the length of the time series sequence is one of the most important features of the dataset. However, in our dataset, the lengths of the trials are unequal, which is also inevitable in real life. Assuring the same calibration time for every subject is very difficult. Additionally, RNNs have the shortcoming that they forget the long-term information. So the sliding window is adopted to slice the trial into shorter pieces as shown in Figure 3.10. Considering the sampling frequency is 60Hz, a 120-frame-sized sliding window is designed which equals 2 seconds time span referring to previous work [7], [10]. The window slides every 15 frames, which means 1.75 seconds' overlapping between neighbouring windows (2-15/60), to avoid splitting potentially important shapes of the input. The window slides from the first frame because the sensor data is considered informative from the first line. The processing methods involving useless data dropping (walking motion filter, etc) are placed before this process as shown in Figure 3.5.

3.6 Network Architecture

This section presents the network architectures and the involved hyper-parameters to address the research problems mentioned above. The architecture of the 2018

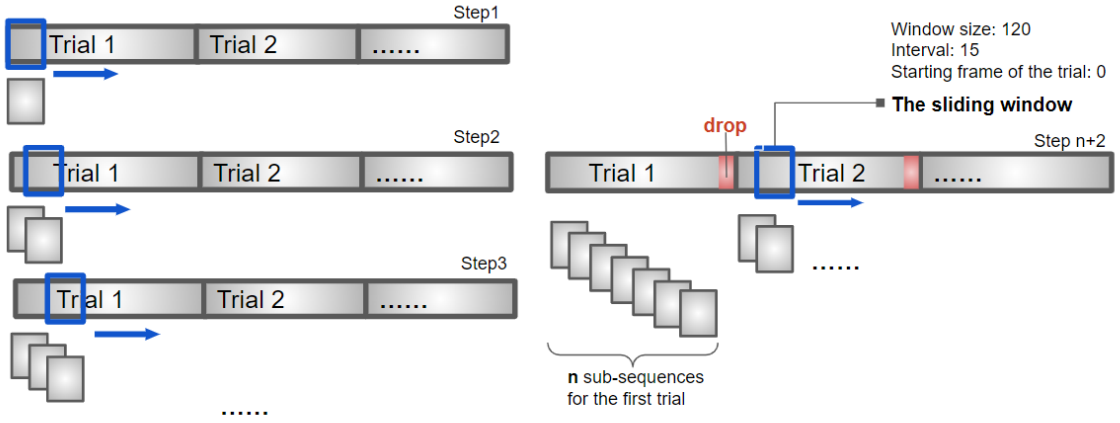


Figure 3.10: The process of sliding window mechanism. The data from all trials are concatenated sequentially. When the sliding window moves along the same trial if the rest data is smaller than the window size, it will be dropped. And the window starts from the first frame of the next trial again.

model has been shown in Figure 2.2, therefore this section mainly introduces the HCFM model. Figure 3.11 illustrates the architecture consisting of three CNN layers, one GRU layer, and three MLP layers connect to a linear classifier in the end. The main difference between HCFM model and the 2018 model is the dissimilar convolutional kernel settings which will be explained later. The model is trained for 200 epochs and early stopping is applied to avoid overfitting. If the performance of the model on the validation set stops improving for 100 epochs, the training will be terminated to save time.

Convolutional Neural Network (CNN) layer

The architecture starts from three CNN layers which respectively consist of a convolution layer, a batch normalization layer and a ReLU activation layer. The self-optimised CNN is efficient in image-related tasks. The convolution kernels each serve as distinct filters generating different feature maps. By customizing the shape and stride of the kernel according to the input, one-dimensional CNN is powerful in extracting features automatically in fields like HAR [63]. The batch normalization layer is used to accelerate and stabilize the training process.

Experiments 2 and 3 involve dissimilar kernel parameter settings realizing the hierarchical convolutional feature merging. The logic is hierarchically merging the extracted features till the 6-channel/10-channel input is 1-channel. The size of the kernel changes among different layers, which indicates the different scales of feature extracting. To illustrate, the first convolutional kernel has the height of 3 as shown in Figure 3.12, learning the features from acceleration and gyroscope respectively. Then the second kernel's height is 2, fusing the learned features of acceleration

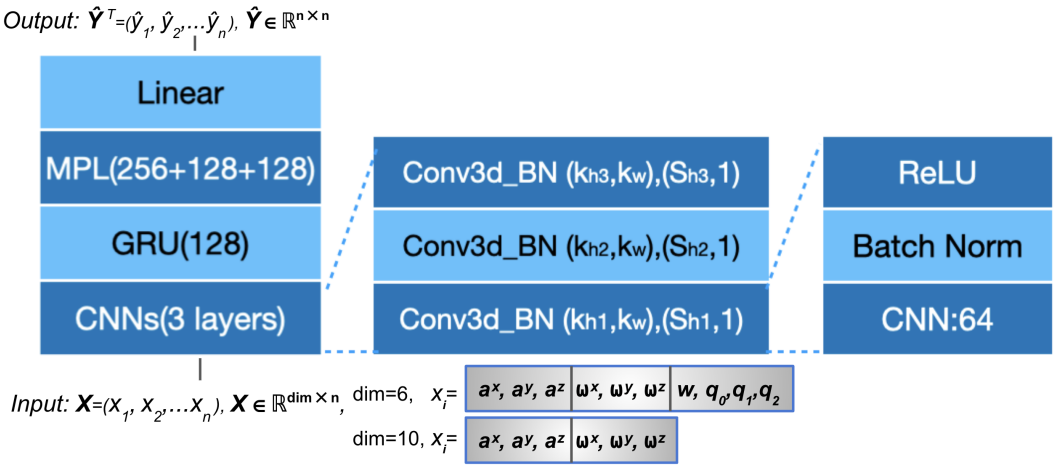


Figure 3.11: The architecture of the HCFM model. The hyper-parameters in the first bracket indicate the 3-d kernel size, including the kernel height K_h, and kernel width K_w which always equals 16. The second round bracket (S_h,S_w) represents the stride height and stride width when sliding through the input. S_w is 1 all the time. The kernel depth and the stride depth not shown on the plot are both equal to 1 for all layers. Different kernel settings allow distinct experiments as introduced. dim has too values for 6-channel and 10-channel input methods.

and gyroscope. Finally, the 2d features are merged using a kernel with a height of 1. Because of the unequal dimensions of the sensor data(acceleration is 3d and rotation quaternion is 4d), methods of 0 padding acceleration and gyroscope into 4-channel input will be implemented in experiment 3 to realize the convolutional feature merging logically with dissimilar dimensions. The kernel parameters are listed in Table 3.3.

Table 3.3: Convolution kernel setting in each experiment.

Experiment	K_{h1}	K_{h2}	K_{h3}	S_{h1}	S_{h2}	S_{h3}
1	1	1	1	1	1	1
2	3	2	1	3	1	1
3	4	3	1	4	1	1

Recurrent Neural Network (RNN) layer As a recurrent mechanism, GRU can expose the temporal dynamic behaviour of the input using the memory unit to keep the history information to the next time step. The traditional RNN structure suffers from a vanishing gradient problem causing difficulty preserving the information many steps ahead. GRU and LSTM solve this problem by using gates to control if

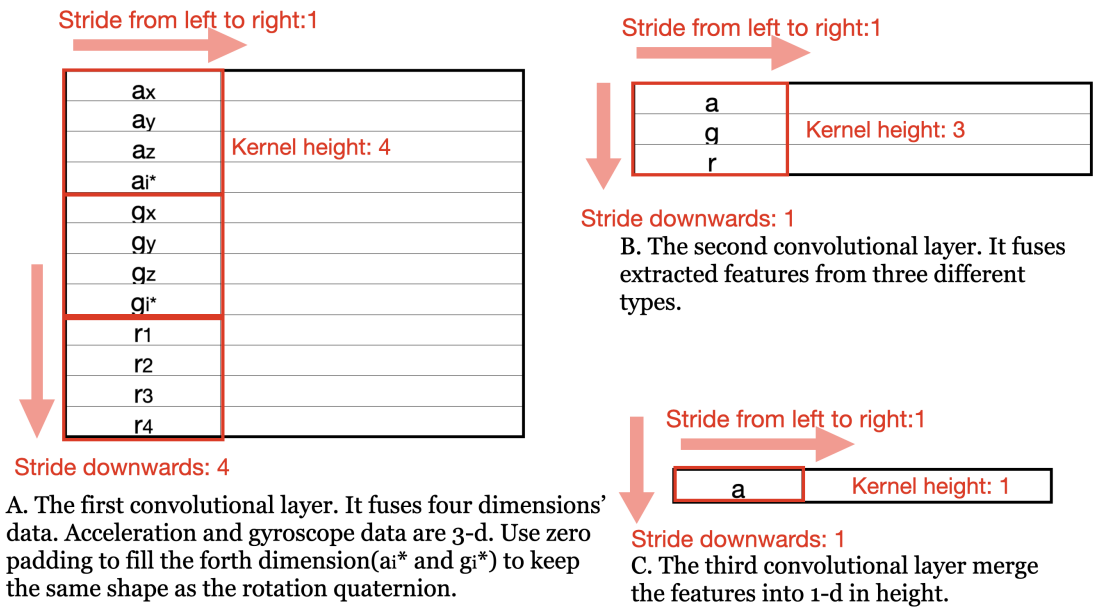


Figure 3.12: The process of hierarchical convolutional feature merging.

the history information is being sent to the next step based on its importance to the whole sequence. GRU outperforms LSTM in I2S classification according to earlier research [7]. This layer involves 128 GRUs. The output of the RNN layer will be flattened before input to the next coming MLP layers.

The dataset in this project is limited in size which makes it challenging to avoid the problem of overfitting. Firstly, inspired by the previous work [7], [10], [30], we add zero-mean Gaussian noise to the accelerations. Secondly, we use the 5-fold cross-validation approach as shown in Figure 3.13. The subjects in the testing set are guaranteed to be absent in the XsensMotion in case the results are biased.

3.7 Performance Evaluation

First of all, to prevent assigning to IMU to the same segment during the test stage, we adapt a function minimizing the cost of wrong labelling inspired by previous researchers [10]. Here, we assume the number of test IMUs is equal to the training IMUs. Otherwise, columns for the absent IMUs can be removed according to the given test IMU configuration in an extra process, to get a square matrix as well. Given n as the number of stacked input (unlabelled IMUs) in the same window (Figure 3.6), let $\hat{\boldsymbol{Y}} \in \mathbb{R}^{n \times n}$ denotes the output of the neural network, boolean matrix $\boldsymbol{Y}_{assigned} \in \mathbb{R}^{n \times n}$ where the sum of each row or column is 1. If $\boldsymbol{Y}_{assigned}(i, j) = 1$,

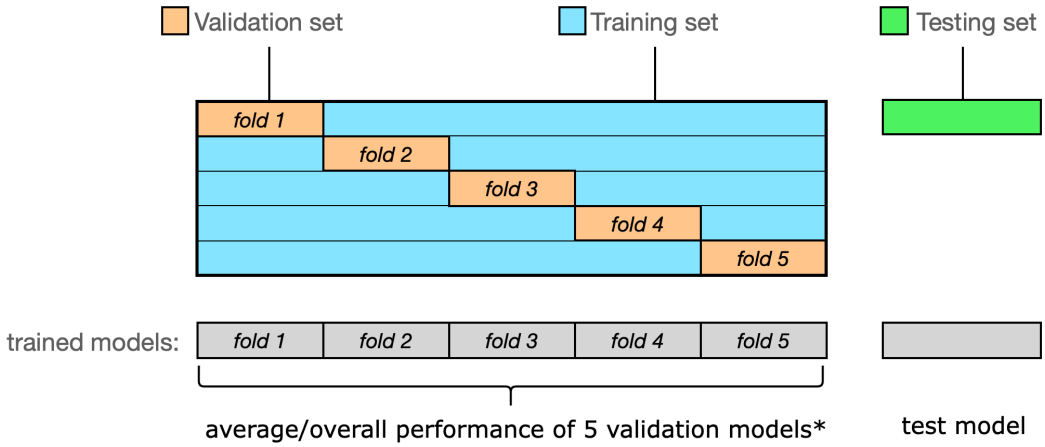


Figure 3.13: The 5-fold cross-validation method. Each trial in XsensMotion will be used four times as the training data and only once as the validation data. After determining the architecture and every detail involved, the model will be trained, taking the whole dataset XsensMotion as the training set and testing it on a new test set with no overlapping trials in between.

the i-th IMU is assigned to the j-th segment. The optimization is realized by function:

$$\hat{\boldsymbol{Y}}_{assigned} = \arg\max_{\boldsymbol{Y}} \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{Y}_{assigned}(i,j) \cdot \hat{\boldsymbol{Y}}(i,j) \tag{3.5}$$

Thus $\hat{\boldsymbol{Y}}_{assigned}$ is our window-wise prediction.

To evaluate the performance quantitatively, we firstly think about the commonly-used metric–prediction accuracy. The normal prediction accuracy(hereinafter referred to as window-wise accuracy) used in previous work [7], [10] is based on all single windows (the sliced short sequences) which are shuffled before training to keep the model unbiased. It is worth mentioning that stacking all IMUs in a window is just out of structural convenience for model design and optimization function application. In the following text, when talking about the word ”window” with prediction performance, we indicate the minimum unit with the 120-frame inertial data of only one sensor. The window-wise accuracy is calculated as:

$$\boldsymbol{A}_{window-wise} = \frac{\boldsymbol{N}_w^{correct}}{\boldsymbol{N}_w} \tag{3.6}$$

where $\boldsymbol{A}_{window-wise}$ denotes the overall accuracy rate of the 120-frame one-sensor input, $\boldsymbol{N}_w$ denotes the total number of minimum input units, $\boldsymbol{N}_w^{correct}$ denotes the corrected classified sensors.

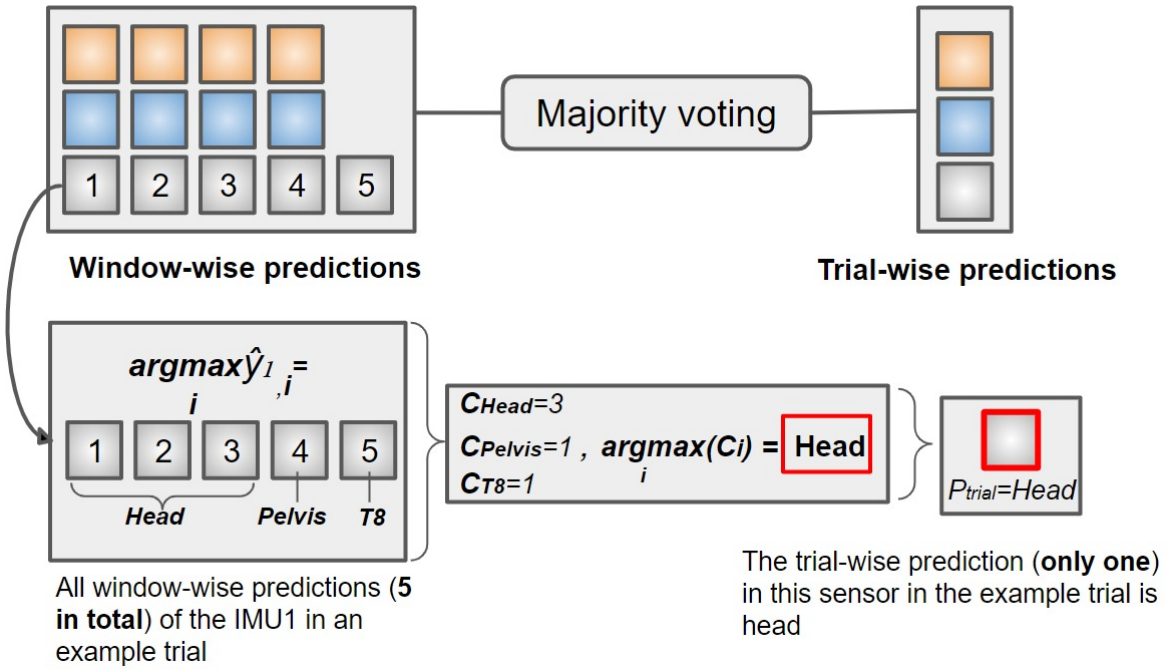


Figure 3.14: The plot shows how majority voting works here. For all the 2-second small units generated by the sliding window method, one prediction, the so-called window-wise prediction, with the highest score is given based on the output of the linear layer. Those predictions are related to the same true label (IMU location) and the same trial. Majority voting is then applied and determines only one result of the whole time sequence of one IMU's data in a trial, which is called trial-wise prediction. In an extreme situation, if there are two most frequently voted, prediction is determined according to the order in which the labels are entered.

When we look into the trials, we found that some segments are more prone to error with a lower accuracy which influence the overall performance. However, the accuracy is still high than, for example, 50%. Therefore, we propose the concept of trial-wise accuracy to enhance the performance since we aim to assign IMU based on the whole calibration instead of one window. In another word, all windows belonging to the same trial can be used together to draw the conclusion. We use majority voting to determine one most reliable final decision for the trial. In one trial t, for a random segment s, let n denote the number of involved segments,$c_i(1 \leq i < n+1)$ the amount of windows predicted to be the i-th segments, so $C = [c_1, c_2 ... c_n] \in \mathbb{R}^n$ consists of the votes for all the n possible segments. The most frequent votes will be the winner:

$$\hat{\boldsymbol{I}} = \arg\max_{\boldsymbol{i}} \boldsymbol{c_i} (1 < \boldsymbol{i} < \boldsymbol{dim}(\boldsymbol{C}) = \boldsymbol{n} + 1) \tag{3.7}$$

$\hat{\boldsymbol{I}}$ is the predicted segment. So instead, the overall trial-wise accuracy is:

$$\boldsymbol{A_{trial-wise}} = \frac{\boldsymbol{N_{\hat{I}}^{correct}}}{\boldsymbol{N_{\hat{I}}}} \tag{3.8}$$

where $\boldsymbol{A_{trial-wise}}$ is the overall accuracy after majority voting for one trial(hereinafter referred to as trial-wise accuracy). In this case, the smallest constituent unit of input is all windows belonging to the same segment in a single trial as shown in Figure 3.14. If there are $\boldsymbol{n}$ segments, then $\boldsymbol{N_P} = \boldsymbol{n}$. $\boldsymbol{N_P^c}$ denotes the sum of the corrected classified segments. For the whole testing set, we use the average of $\boldsymbol{A_{trial-wise}}$:

$$\overline{\boldsymbol{A}}_{\boldsymbol{trial-wise}} = \frac{\boldsymbol{1}}{\boldsymbol{m}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{A_{trial-wise}^{i}} \tag{3.9}$$

Besides the accuracy mentioned above, we also use prediction certainty in the result section. Because when the true label of the input is unknown, we still want this model to tell something other than the prediction itself. For example, majority voting gives one prediction per trial for an unknown sensor, so the certainty of the prediction can help the user decide if the result is taken. A trial-wise prediction with a certainty less than 50%, for example, is probably rejected. What's more, prediction certainty is different from accuracy. It tells quantitatively how confident the model is. For example, the model also gives a "switched" prediction with 0% accuracy but 100% certainty, while it normally performs accurately. This could be the switch between the true labels of the two sensors. In this case, the value of certainty can be used to detect abnormal input. The certainty of the prediction on a sensor is:

$$\boldsymbol{P_{certainty}} = \frac{\boldsymbol{c_{\hat{I}}}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{c_i}} \tag{3.10}$$

$\boldsymbol{c_i}$ is the number of votes for a random possible segment $\boldsymbol{i}$. $\hat{\boldsymbol{I}}$ is the segment with the highest votes in a trial, so $\boldsymbol{c_{\hat{I}}}$ is the number of votes for $\hat{\boldsymbol{I}}$.

Chapter 4

Results

In this section, we evaluate the performance among all the introduced models window-wisely and trial-wisely, then includ some typical mistakes made by different models and show the confusion matrices of the selected model–HCFM model with WMF (Section 4.1). Then the performance is shown with respect to different segments (Section 4.2). The impact of the types of trial motion is also explored in Section 4.3. Eventually, the robustness of the models is investigated from an entry point of individual performance of the involved 5 subjects, then left and right shoulder recognition and finally the performance of the proposed model trained on TotalCapture [31] dataset (Section 4.4).

4.1 Model comparison

4.1.1 Window-wise performance and trial-wise performance

The average $\boldsymbol{A}_{window\_wise}$ of 30 trials are shown in Table 4.1. Lower-body The best performance is achieved by HCFM architecture with WMF for each body configuration. Compared to the baseline model–previous proposed 2018 model, the prediction accuracy is 6.5% higher on lower-body configuration (8.8% better on the upper body and 9.3% full body). Without WMF, HCFM performs worse, but still better than the 2018 model when the input is 6-channel. For lower-body configuration, except for WMF model, the other models perform similarly with the prediction accuracy around 90%, and the lowest and the second lowest accuracy are only 1% per cent apart. However, the poorest performance is both found when using the heading correction model, which is at least 8.4% and 4.9% less accurate than the other models with upper-body and full-body configurations respectively. It has been demonstrated that when the model and data processing methods are unchanged, directly adding rotation quaternion to the input doesn't enhance the accuracy except for the heading correction model. A table comparing the performance before and

after adding rotation quaternion to the input is available in Appendix B.1. The 10-channel heading correction model has better performance on lower and upper body configurations.

Table 4.1: The window-wise accuracies ($A_{window-wise}$) of all models. Each method consists of 2 input methods and 3 body configurations as introduced in Section 3.4. The listed prediction accuracy is the average of 30 trials (including three trial types introduced in Section 3.3). 6-channel input includes acceleration and angular velocity, while 10-channel input has the additional four-dimensional rotation quaternion. The best combination of model and input data for each body configuration is highlighted in the table.

No.	Description	Input channels	Lower body	Upper body	Full body
Exp.1	**2018 model**	6	0.924	0.781	0.803
		10	0.888	0.745	0.779
Exp.2	**HCFM**	6	0.937	0.800	0.823
		10	0.878	0.744	0.777
Exp.3	**HCFM+WMF**	6	**0.989**	**0.869**	**0.896**
		10	0.982	0.869	0.885
Exp.4	**HCFM+heading correction**	6	0.900	0.660	0.738
		10	0.935	0.691	0.728

Table 4.2: The majority voted trial-wise accuracies ($A_{trial-wise}$) of all models. Models achieving the highest prediction accuracy are highlighted for each segment. Different from window-wise performance, the predictions are more similar after majority voting, especially on lower-body configuration.

No.	Description	Input channels	Lower body	Upper body	Full body
Exp.1	**2018 model**	6	**1.000**	**0.927**	**0.953**
		10	**1.000**	0.909	0.949
Exp.2	**HCFM**	6	**1.000**	0.918	**0.953**
		10	0.990	0.900	**0.953**
Exp.3	**HCFM+WMF**	6	**1.000**	0.924	0.947
		10	**1.000**	0.924	0.949
Exp.4	**HCFM+heading correction**	6	**1.000**	0.880	0.910
		10	**1.000**	0.827	0.854

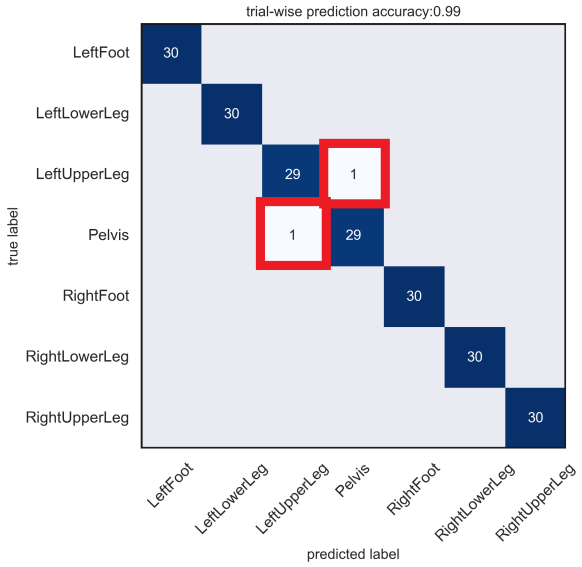


Figure 4.1: Confusion matrix of 10-channel lower-body HCFM model (Exp.2).

After applying majority voting to the window-wise predictions, the accuracy is increased for all models. The differences between best and poorest accuracy are reduced by 10% for all configurations. As shown in Table 4.2, the accuracy is 1 for all models on lower-body configuration except for 10-channel HCFM model. Because the majority voting only generates one prediction of a segment per trial, for each segment there are finally 30 (equals to the number of trials) predictions. The values of accuracy are more sensitive to a few different predictions. For example, as shown in Figure 4.1 and Figure 4.4b (the lower-body configuration), 2 failed predictions between left upper leg and pelvis from the same person (subject-1) result in the different accuracy between lower-body 10-channel HCFM model and other model.

The situation is similar between window-wise best model–6-channel WMF model, and trial-wise best model – 6-channel 2018 model when comparing their majority vote prediction: the poorer performances on upper and full body models are caused by 1 and 3 failed predictions (indicated in Figure 4.4 using black boxes), which are related to the same subjects (subject-1), if look into the results grouped by subjects in Figure 4.3a and Figure 4.3b.

In this case, 6-channel HCFM+WMF model is still selected as the best model in this project. The majority voting has been proved to be efficient in enhancing prediction accuracy. However, the majority vote accuracy between the 2018 model and HCFM+WMF is close to each other and the gap was caused by a few biased test trials of the same subject, thus we end up opting for the best model by evaluating window-wise accuracy.

4.1.2 Typical mislabelling of distinct approaches

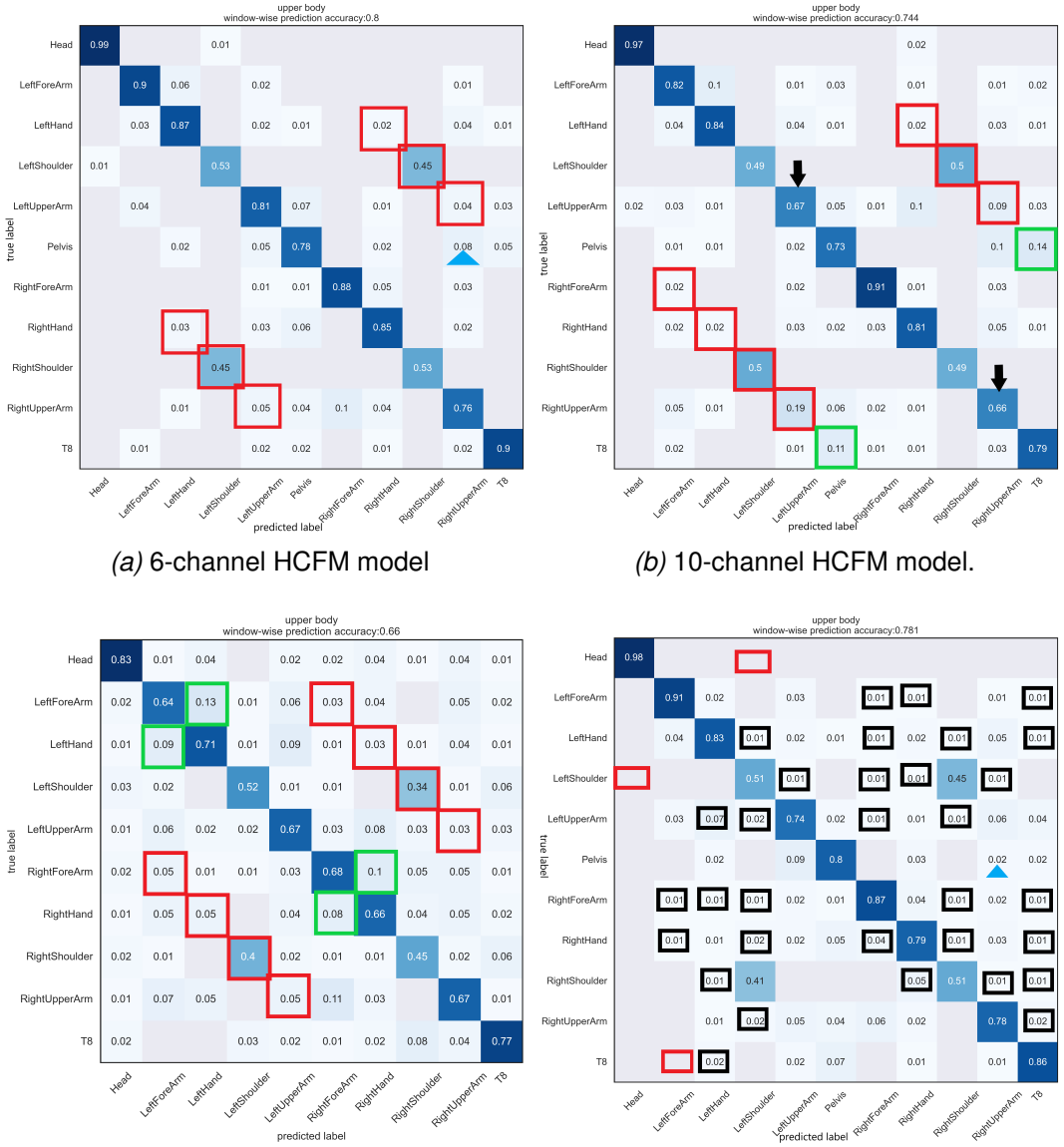
Analysing the overall $\boldsymbol{A}_{window-wise}$ and $\boldsymbol{A}_{trial-wise}$ gives limited information on specific situation of IMU assignments. Thus to discuss the typical mistakes of each method, we select upper-body configuration models as examples. The models with other configurations will not be included here as they are showing similar issues.

Four confusion matrices are included (see Figure 4.2). The comparison will be drawn between 6-channel HCFM model (Figure 4.2a) and 4 other models: first, baseline (2018) model (Figure 4.2d) discussing the superior performance of HCFM model and the different distributions of these predictions; second, 10-channel HCFM (Figure 4.2b) model, to introduce the results of using rotation quaternions as input; third, 6-channel HCFM model with heading correction method (Figure 4.2c); finally, 6-channel HCFM model with the walking motion filter (Figure 4.4a).

As Figure 4.2a and Figure 4.2d show, 2018 model has many types of mislabelling (the black rectangle) not in HCFM model and most of them are around 1%. The small blue triangles in both Figures indicate the pelvis-to-right-upper-arm mislabelling. Besides the 6-channel upper-body model, all HCFM models have higher pelvis-to-right-upper-arm accuracy compared to their corresponding 2018 model. Figure 4.2b gives an example of the consequences of inputting rotation quaternions in the neural network. 10-channel models perform poorer on most of the l/r recognition: for example, 19% of the IMUs on the right upper arm is assigned to the left side based on 10-channel HCFM model, while the number is 5% in 10-channel model. Additionally, it has pelvis-sternum (T8) switching problems and low accuracy related to the left and right upper arms indicated by the black arrows.

Applying heading correction to HCFM model decreases the overall accuracy significantly from 80% to 66%. Comparing Figure 4.2c and Figure 4.2a, also bring about notable errors on segments like forearms and hands which are physically close to each other (Figure 1.1b) and have even more types of mislabelling compared with the 2018 model (Figure 4.2d). Prediction on all segments is less accurate comparing the accuracy rates on the diagonal of the confusion matrices. However, the l/r recognition mislabelling rates indicated by red boxes are not remarkably increased, and some of them are even lower (l/r shoulder switches, left-to-right upper arm mislabelling rate) after applying heading correction.

Using WMF on HCFM model enhances the recognition accuracy for nearly all segments. There are no big differences between the coloured blocks in the two Figures. In Figure 4.4a (subplot in the middle) the $\boldsymbol{A}_{window\_wise}$ of each segment (numbers on the diagonal) increases, and most of the mislabelling rates decrease. The only poorer performance on WMF and HCFM model is l/r upper arm recognition.



(a) 6-channel HCFM model

(b) 10-channel HCFM model.

(c) 6-channel HCFM model + heading correction

(d) 6-channel 2018 model.

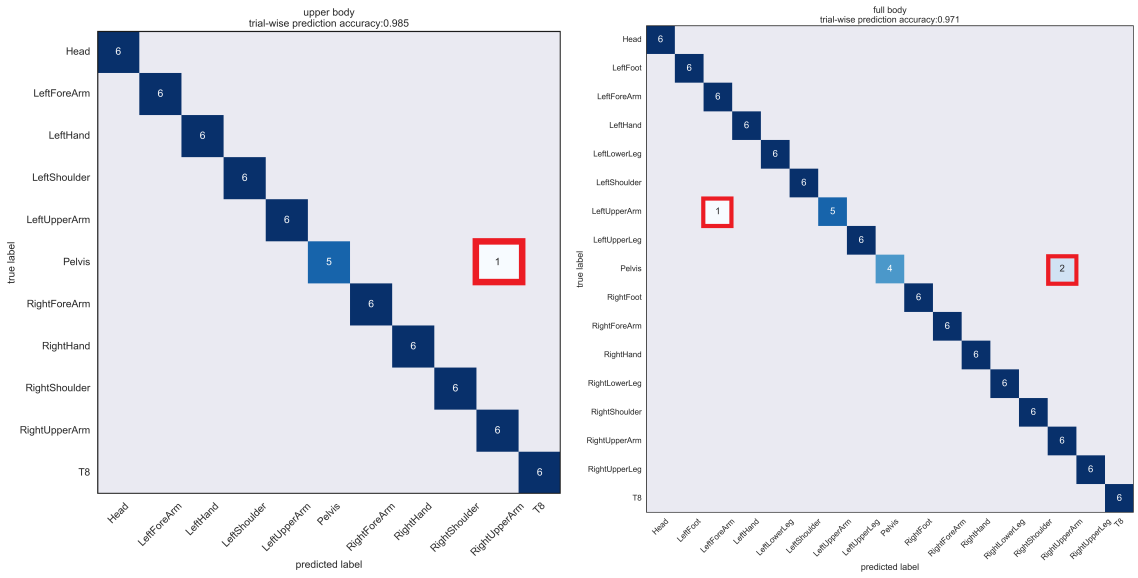
Figure 4.2: Confusion matrices of the models with upper-body configuration. Red boxes in Figure 4.2aFigure 4.2b and Figure 4.2c indicate the mistakes on l/r side recognition. The black arrow indicates the poor performance on upper arms and blue triangles for the pelvis-to-right-upper-arm error. Green boxes indicate some notable segment switches. In Figure 4.2d, black rectangles show the errors absent in Figure 4.2a and red rectangles are errors in Figure 4.2a but not shown here.

4.1.3 Look into the selected model

Representative confusion matrices of the other methods are already given before. Therefore only the selected method– 6-channel HCFM model + WMF are investigated in this section by showing 6 confusion matrices.

Figure 4.4 shows the assignment results of 6-channel HCFM+WMF model. From the confusion matrices, we find the proposed method is reliable on lower limbs (upper leg, lower leg and foot) prediction. IMUs mounted on lower limbs (legs and feet) all achieve accuracy over 96% with both lower-body and full-body configurations. The matrix of full-body configuration also shows the correctness of the upper or lower body assignment (u/l assignment). In another word, few IMUs mounted on the upper body are assigned to the lower body and vice versa.

Incorrect u/l assignments take up less than 0.005 of all predictions, thus not shown in the matrices after rounded to the hundredths place. More specific results and analysis for segments can be found in Section 4.2. The trial-wise assignment

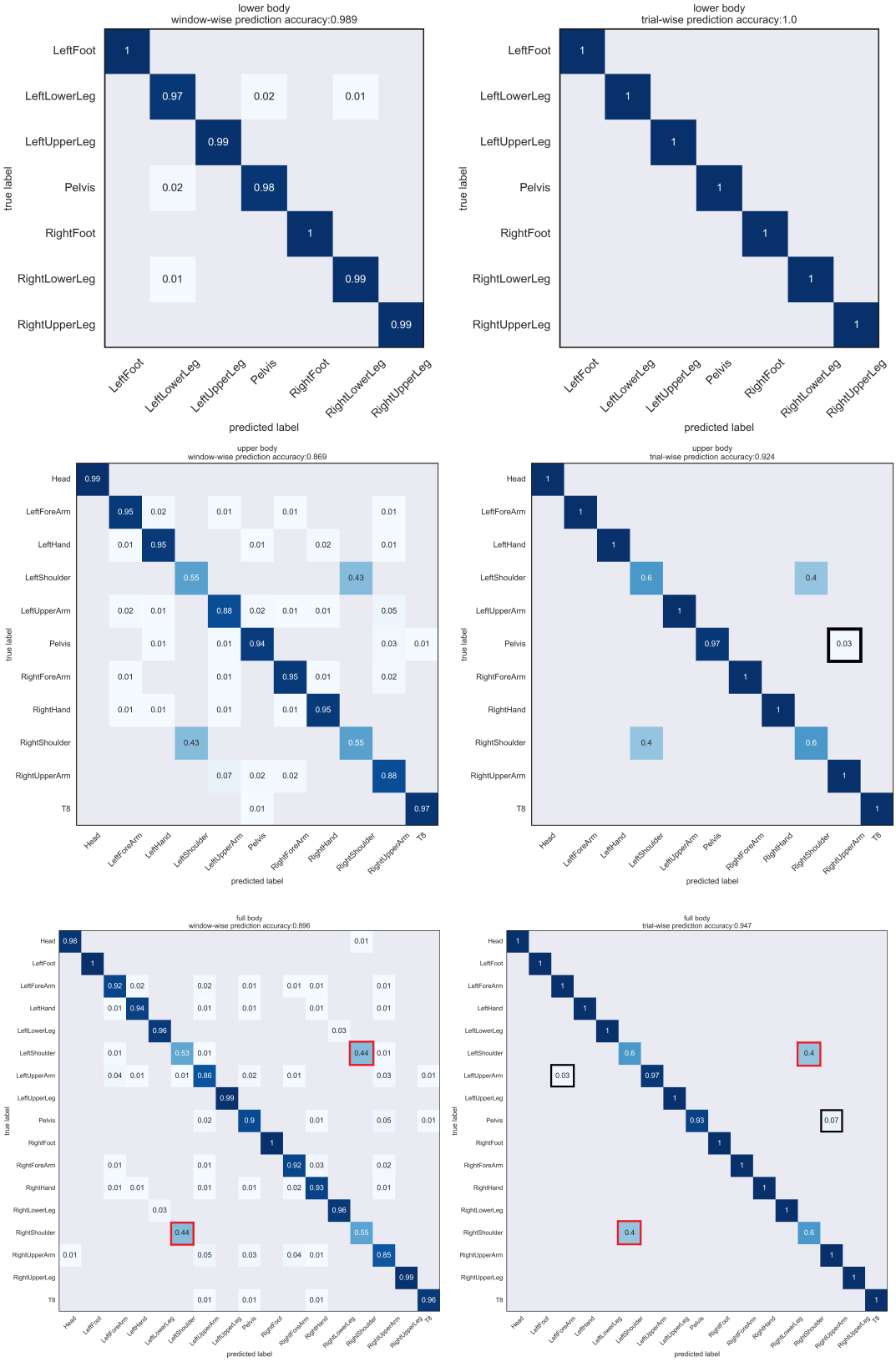


(a) Upper body *(b)* Full body

Figure 4.3: The absolute trial-wise prediction of Subjet-1 (Exp.3). There are 6 trials related to Subject-1. The 4 failed predictions (highlighted use black boxes) shown in Figure 4.4 all from two trials belong to the same subject. Two Pelvis-RightUpperArm mislabelling Figure 4.3a and Figure 4.3b, and one LeftUpperArm-to-LeftForeArm mislabelling come from one ”abnormal” trial; the rest Pelvis-to-RightUpperArm mislabelling in Figure 4.3b is from a ”standard” trial.

of the proposed model is illustrated in Figure 4.4 on the right side. Among all the trials, the majority of the segments generate the expected correct assignment based

on their respective window-wise predictions above 50%. The failed predictions all fall into the same type of mistake: l/r shoulder switch. Figure 4.4 indicates the particular low prediction accuracy on the left and right shoulder assignment which will be further investigated in Section 4.4.



(a) Window-wise accuracy.

(b) Trial-wise accuracy (window-wise accuracy after majority voting).

Figure 4.4: Confusion matrices of the model with best window-wise prediction accuracy: 6-channel HCFM+WMF model (Exp.3). Elements smaller than 0.5% of the number of windows in a trial are masked in the plots to guarantee the visibility after rounded to the hundredths place as 0.

4.2 Segment comparison

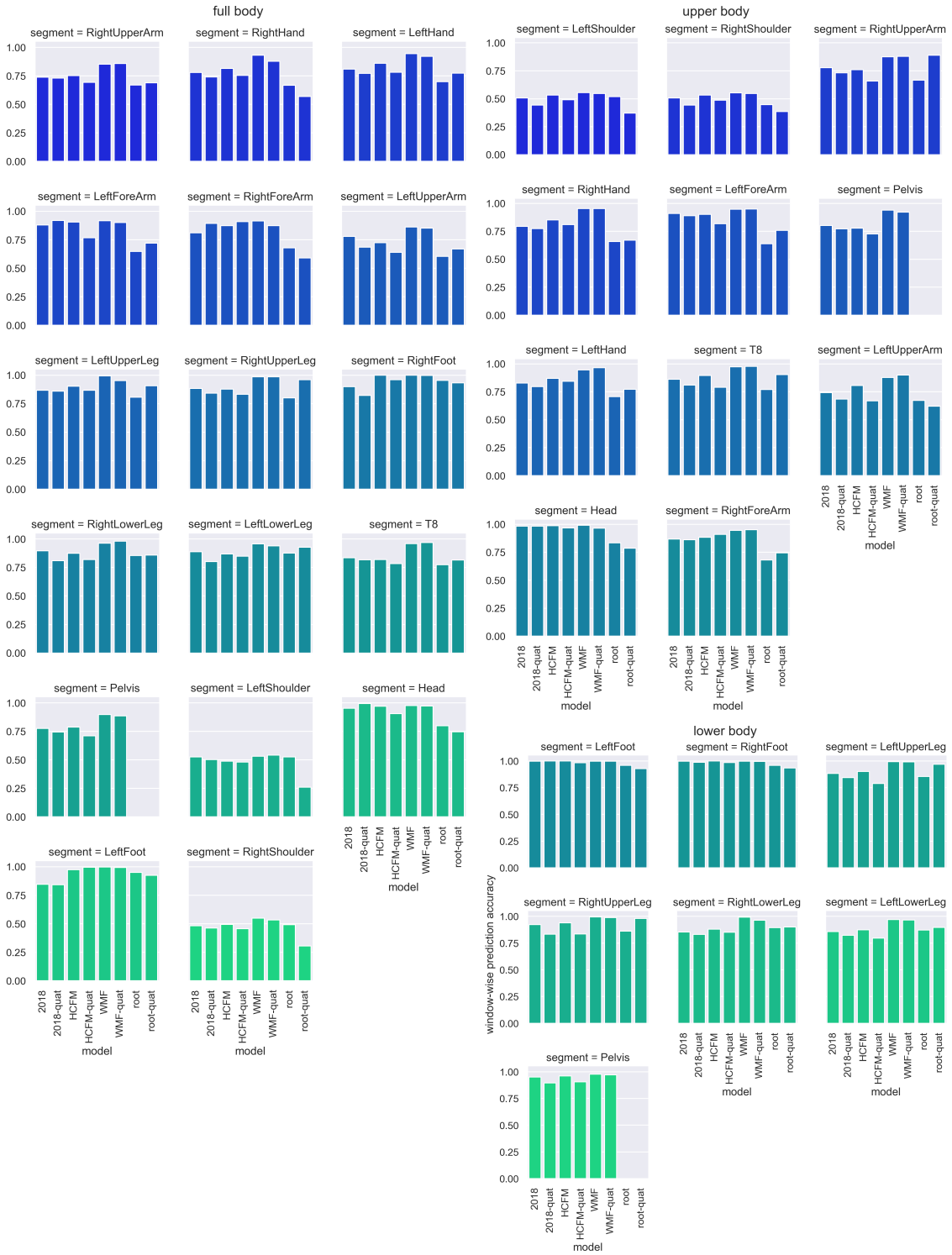


Figure 4.5: Bar plots of accuracy(y-axis)-model(x-axis) relationship. Model names can be found at the bottom under x-axes. The names include ”quat” represent 10-channel input, ”WMF” for HCFM with WMF (Exp.3) and ”root” for HCFM with heading correction (Exp.4).

According to Table 4.1 and Table 4.2, for all 8 methods, both window-wise and trial-wise performance on lower-body configuration is the best, followed by full-body configuration, and upper-body configuration is the poorest.

Different models have the dissimilar ability to recognise the segments respectively as illustrated in Figure 4.5. As the model has the best performance evaluated by window-wise average prediction accuracy, 6-channel HCFM+WMF is optimal for most of the segments but not all of them. For example, with a full-body configuration, 10-channel HCFM+WMF is better on assignment right upper arm, right lower leg and sternum (T8), and the 10-channel 2018 model achieves the highest accuracy on head. For some models, the high accuracy on upper/lower body configuration is not maintained with full-body configuration (e.g., on l/r foot, 2018 model's performance is poorer with full-body configuration than lower-body, however, HCFM and HCFM+WMF can keep their prime performance). The inability to accurately assign l/r shoulders is not limited to the two models mentioned in Section 4.1.3, but in all experiments.

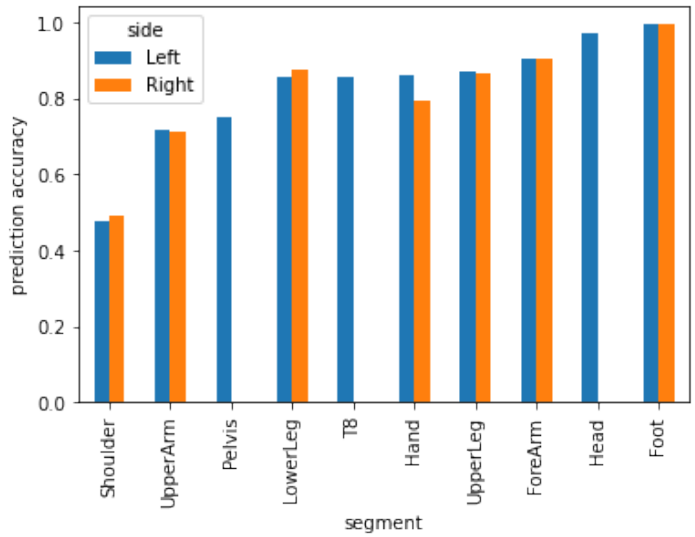


Figure 4.6: The window-wise prediction accuracy of the proposed model on the whole test set (30 trials). Symmetrical body segments are grouped. Single segments are put on the spot of the "Left" segment. The order of x tick label is sorted in ascending order by the corresponding left segments' accuracy rates.

Though all 17 segment-wise prediction accuracies with full-body configuration are surpassed by those with upper or lower-body configuration, the gap between two (or three for pelvis) types of performance is not significant as the model is trained without dependency on multiple sensors. Therefore, the following segment comparison is concentrating on full-body configuration. Figure 4.6 shows the aver-

age window-wise accuracy of each segment of the proposed model. From the plot, there is no evidence to say that the model is more accurate on the left or right side. Except for the shoulder, the other segments' accuracy rates are higher than 70%. Foot recognition has the highest accuracy rounded up to 100%, while the shoulder is lower than 50%.

To study the performance of the best model with respect to a different segment in each trial to provide a more in-depth view, the distribution of window-wise prediction accuracy is visualized in Figure 4.7. It can be observed that the segments that belong to the same category, that is, regardless of the left and right sides, have similar performance. All IMUs mounted on lower limbs outperform the ones on upper limbs, which is consistent with common sense as the movement of lower limbs is more recognizable in walking motions. The distributions of the prediction accuracy of upper-body segments, including hands, forearms and upper arms, are broader compared to lower-body segments. The spread-out distribution suggests the particularity of upper limb movements in different subjects. Additionally, the distribution of shoulder prediction accuracy is special as its accuracy scale ranges from about 0 to 1. In around half of the trials, the shoulder recognition failed completely (accuracy close to 0%), but in the other half, the model achieves accuracy approaching nearly 100%. The shoulders achieve an average accuracy of less than 55%. Combined with Figure 4.4, if we pay attention to the red box in the full-body confusion matrix, the proposed model is prone to l/r shoulder switch mistakes. Ignoring the incorrectness of side recognition, the prediction accuracy rate has increased to 98%.

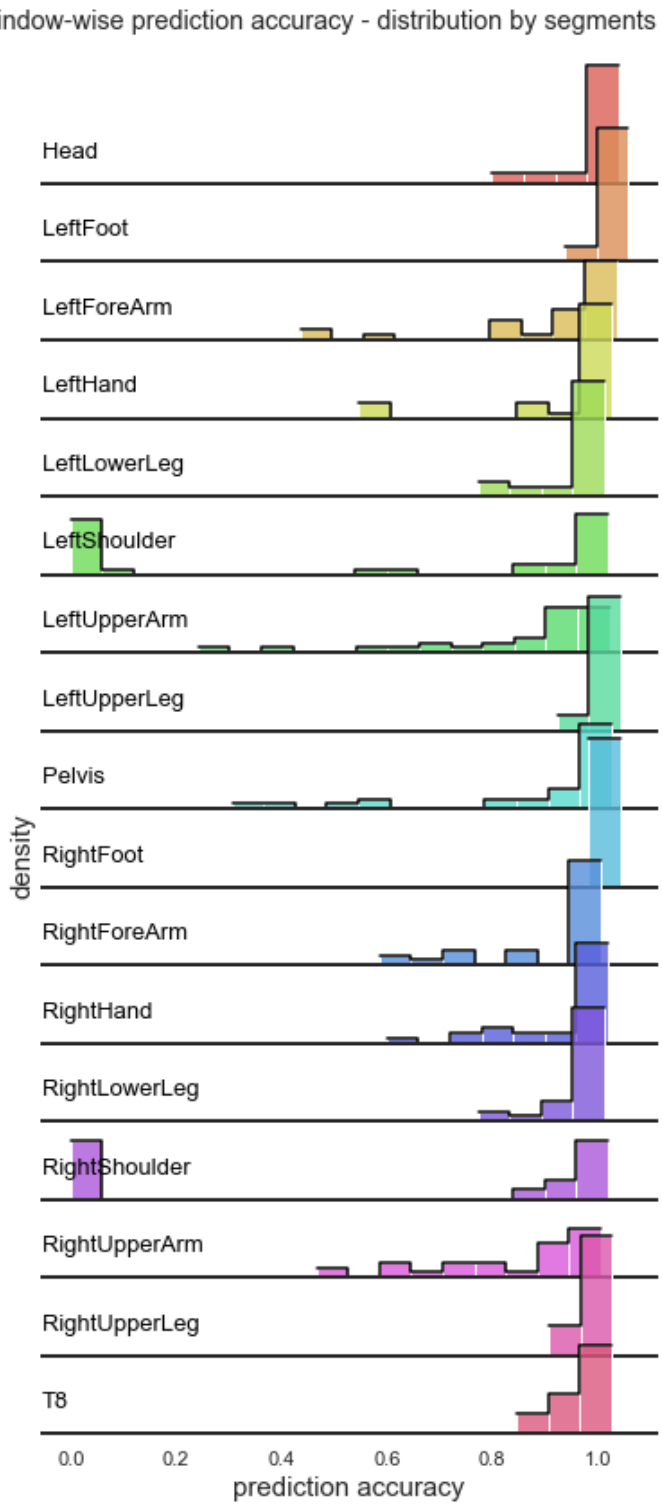


Figure 4.7: The histogram shows frequencies of window-wise prediction accuracy among 30 test trials within full-body HCFM +WMF model for each segment. Histogram without left side border line indicates the minimum accuracy rate among 30 trials and right for the maximum.

4.3 Motion comparison

As mentioned in Section 3.3, there are three trial types: standard, infinity and abnormal. The average window-wise and trial-wise prediction accuracy grouped by motions are shown in Figure 4.8. Except for 10-channel HCFM+headning correction (labelled as ”root-quat” on the plot), the model performs best on ”infinity” trials and worst on abnormal trials. The majority voting bridges the gap between different walking motions which indicates tolerance for unexpected motions as the trial-wise bars are similarly tall.

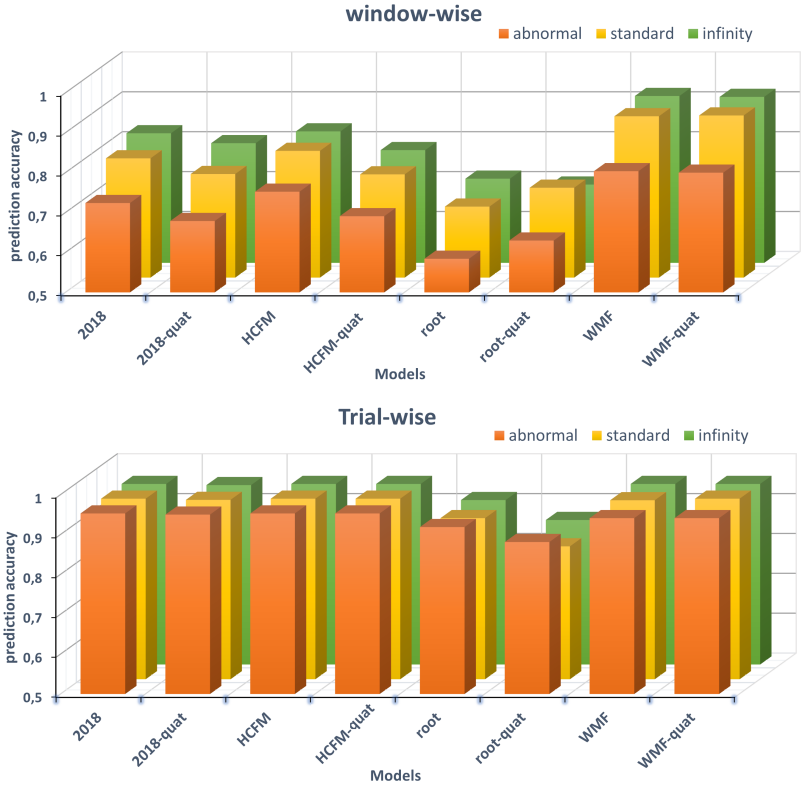


Figure 4.8: Bar plots of the prediction accuracy grouped by trial types. The y-axis starts from 0.5 for both plots. The average window-wise accuracy rates of abnormal, standard and infinity trials are 0.756, 0.817, 0.834, respectively; after majority voting the rates end up with 0.936, 0.931, 0.937 correspondingly. ”root” is for heading correction methods, and ”quat” is for the 10-channel input setting.

4.4 Robustness

4.4.1 Performance on Distinct Subjects

As formerly discussed, Figure 4.3 indicates the individual features of the subjects which are reflected in the prediction accuracy. For this reason, to further study the performance of the proposed model (full body), the window-wise and trial-wise accuracy rates are grouped by subjects as shown in Figure 4.9. The gap between the best and poorest subjects is comparably huge: window-wise 0.11 and trial-wise 0.12. The best subject achieves the window-wise accuracy of 96% with full-body configuration which is 6% higher than the average. What's more, its trial-wise accuracy is 100%, which suggests a high probability of commercial application in the future. Nevertheless, the poorest two subjects' accuracy rates are less than the average window-wise accuracy after majority voting.

To explore the reason, confusion matrices of the best and poorest subjects are generated (Figure 4.10). Respective confusion matrices of all subjects with three configurations of the proposed model can be found in Appendix B.3. According to 4.10a, the prediction accuracy of l/r shoulders is rounded up to 0, which means the proposed model fails to recognize l/r for all the trials. The l/r shoulder switch issue will be further discussed in Section 4.4.2.

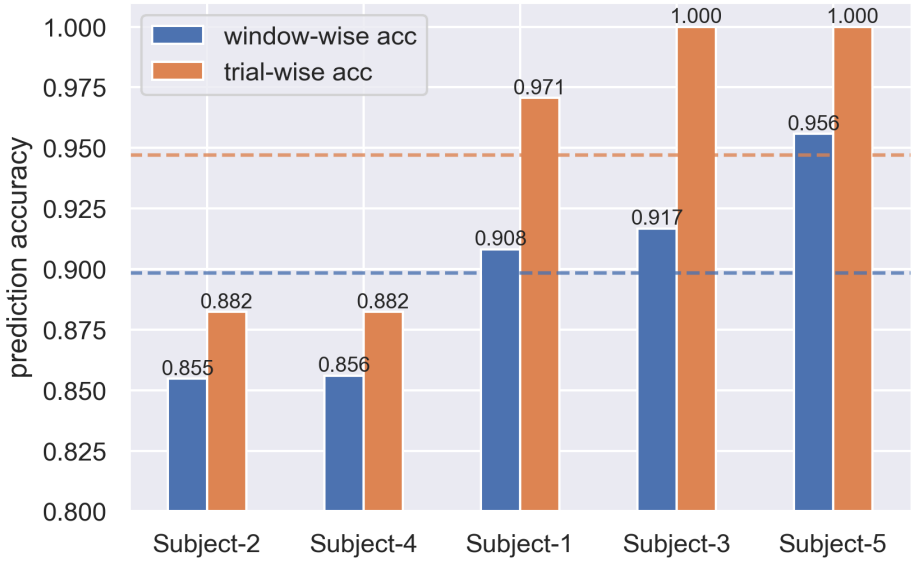
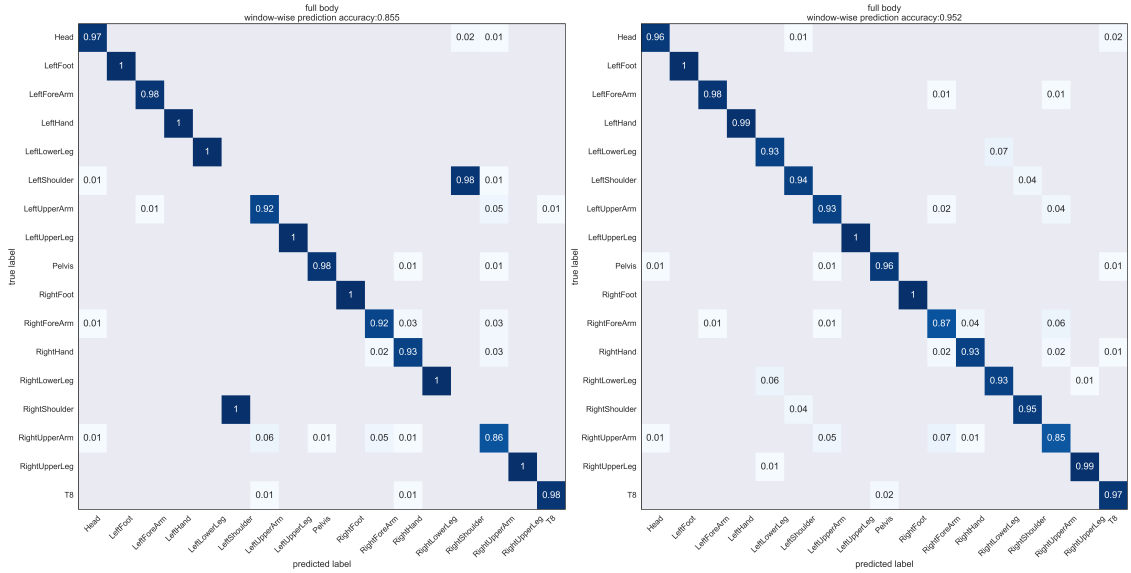


Figure 4.9: Accuracy-Subject bar plot comparing the window-wise (blue bar, left) and trial-wise (orange bar, right) performance of the full-body 6-channel HCFM+WMF model. The dashed lines represent the average accuracy. The lower one is for window-wise accuracy, and the higher line is for trial-wise accuracy.



(a) Subject-2 *(b)* Subject-5

Figure 4.10: Full-body HCFM+WMF model performance on two subjects. Considering all motion types, Subject-2 has the poorest average accuracy rate, and subject-5 has the highest among all the subjects. Except for l/r shoulders, the prediction accuracy between two subjects among the rest segments is relatively close: the gap in between is 0.018 on average and 0.07 at most. The distinction between shoulder recognition performances of two subjects causes a noticeable difference in the overall accuracy.

4.4.2 Left and Right Shoulder Recognition

By analysing the overall confusion matrices (Figure 4.4) and segment accuracy (Section 4.2), the notable low recognition accuracy on the shoulders of the proposed model has been displayed. Section 4.4.1 indicated the connection between low shoulder accuracy and certain subjects. To provide targeted information, the window-wise accuracy of full-body HCFM+WMF model on l/r shoulder is listed in Figure 4.11. For Subject-1,5,3 the performance of shoulder recognition is close to or higher than the overall accuracy rate. For Subject-2,4 the l/r shoulder recognition seems like a complete ”failure” as the average accuracy of 0% and 0.8%.

	LeftShoulder			RightShoulder			average
subject	abnormal	infinity	standard	abnormal	infinity	standard	
1 (Male)	1.000	1.000	1.000	0.978	1.000	0.994	0.995
5 (Female)	0.914	1.000	0.934	0.947	1.000	0.934	0.955
3 (Male)	0.596	1.000	1.000	0.904	1.000	1.000	0.917
4 (Female)	0.035	0.000	0.000	0.012	0.000	0.000	0.008
2 (Male)	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 4.11: 5 test subjects' window-wise prediction accuracy on the left and right shoulder of full-body HCFM+WMF model to the motion type. The subjects are sorted by their average shoulder accuracy rates in descending order. The average window-wise accuracy rates on all segments are 0.908, 0.956, 0.917, 0.856, and 0.855 corresponding to the subjects in the first column from top to bottom.

However, 0% is an unusual accuracy rate in a classifier. First of all, the failure on shoulders is caused mostly by l/r switch as shown in Figure B.3 and Figure B.5. Additionally, as illustrated in Figure 4.12, the prediction certainty of the shoulders is higher than 95%, meaning that guessing the segment is not the case. On the contrary, it shows the confidence of the model in its choices. Finally, a similar situation (l/r shoulder switch on Subject-2,4) exists among all of the models demonstrated in Table 3.2.

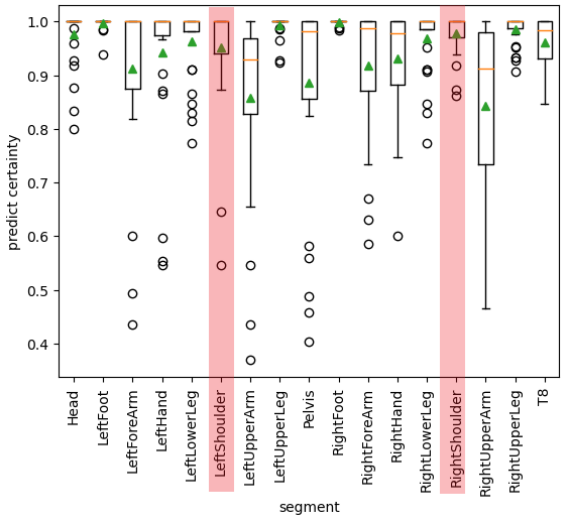
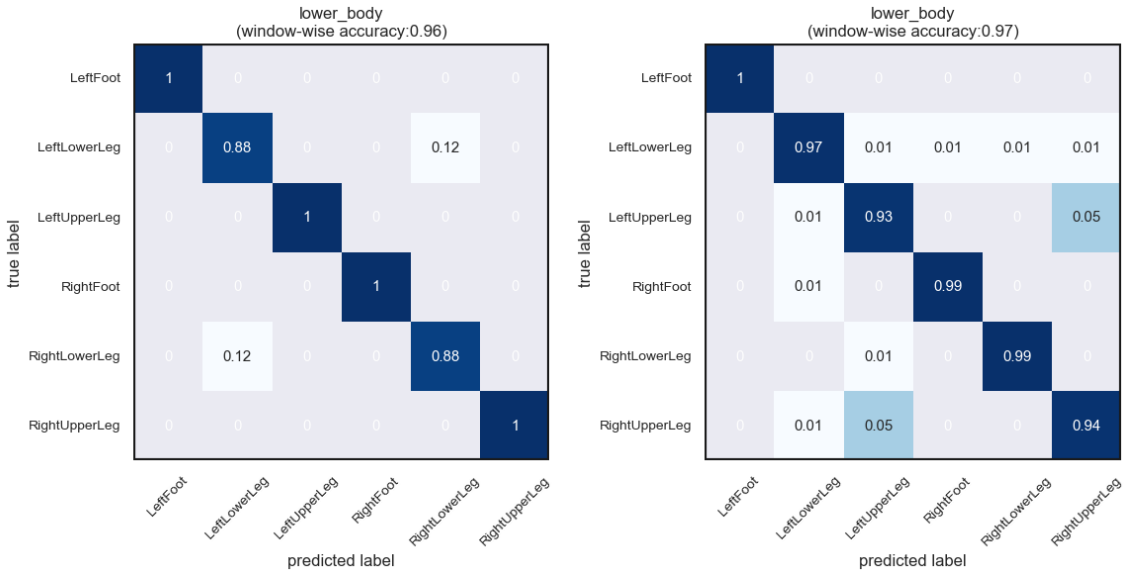


Figure 4.12: Box plot of the prediction certainty of each segment HCFM+WMF model on full body configuration. The certainty of the left and right shoulder is 0.951 and 0.979 respectively, which are both higher than the average certainty value of 0.945 based on all 17 IMUs. The Left and Right shoulders are marked by semi-transparent red strips in the plot.

4.4.3 Results on the TotalCapture dataset

To compare the performance of our selected model with the previous work [7], [10], we trained the model on the dataset TotalCapture [31]. Though Zimmermann et al. do not train their model on the TotalCapture dataset [7], Kaichi et al. [10] provide the results trained on the mentioned dataset using the model proposed by the former research group. We adopt the same dataset division to train (subject 1, 2, 3) and test (subject 5) the model use only trials of the walking motion as Kaichi et al. do. Additionally, since the root sensor is used in the previous model, it is in neither three body configurations, and they also include no IMUs on shoulders. In this case, we only trained our model with a lower-body configuration without the pelvis to ensure the comparability of the results. The confusion matrix of HCFM+WMF model trained on dataset TotalCapture can be found in Figure 4.13a. Confusion matrix in terms of model trained on XsensMotion introduced in Section 3.3 is illustrated in Figure 4.13b. The model trained on TotalCapture has inferior general window-wise accuracy. Despite that, it has more certain predictions on the left and right upper legs and right foot than the model trained on the XsensMotion dataset. The incorrect labels only show up in l/r lower leg recognition in the former model while the latter has a relatively scattered distribution of the errors.



(a) Training set: TotalCapture [31]

(b) Training set: XsensMotion

Figure 4.13: Confusion matrices of 6-channel lower-body HCFM+WMF model without Pelvis. The previously proposed model achieved an accuracy of 96.7% and the confusion matrix can be found in the relevant paper [10] (Figure 9a).

Chapter 5

Discussion

This work aims to evaluate CNN+GRU models combined with data processing methods for I2S assignment using 6-channel (3-axis acceleration and 3-axis angular velocity) or 10-channel (extra rotation quaternion) input data, to find a flexible and accurate way realizing automatic I2S assignment. The proposed approach combines WMF and 6-channel deep learning model HCFM achieved the goal with flexibility in IMU number, which answers the research question defined in Section 1.3.

The baseline model [7] with the same kernel setting for all convolutional layers is outperformed by HCFM model (with and without WMF) according to the window-wise accuracy in Tabel 4.1. The method of step-by-step change in kernel size for each CNN layer proposed by Kaichi et al. [10] is thus proved to be efficient in I2S assignment tasks based on inertial sensor data. Though their overall performance is close, the prediction of the 2018 model includes more types of errors with low rates (1%), while HCFM models' errors are concentrated in fewer places with higher rates. However, the baseline model has better performance after the majority voting. According to Figure 4.2a and Figure 4.2d, the HCFM model is more prone to pelvis-to-right-upper-arm which results in that contributes to the lower ranking based on trial-wise performance. After studying the predictions by trials, it has been observed that the inferior trial-wise performance is because of one certain subject. It has to be further researched what kind of gaits causes the pelvis-to-right-upper-arm error.

To further improve the performance of the deep learning model, we compared two data processing methods: WMF and heading correction with a root sensor. The superior performance results in WMF are in line with previous findings [12], [64] which stressed the distinct motion signature of walking. The reason for the remarkable decrease in the heading correction model's accuracy can be information lost during reference transformation of the non-root sensors, or the absence of the IMU data on the pelvis, which could be crucial to the whole model.

Compared to the previous work, the proposed model is comparatively accurate and flexible at the same time, meanwhile involving a larger scope of body segments.

The proposed model trained and tested on XsensMotion achieves the accuracy of 98.6%, 86.9%, and 89.6% on the lower, upper and full body respectively which are in line with the latest research (Tabel 2.1). In comparison with the shallow machine learning approaches [8], [13], the proposed model is not restrained by complex manual feature selection and has comparable performance after majority voting. Considering the related deep learning approaches, the lower-body performance of the proposed model (96%) is close to Kaichi et al.(96.7%) and Zimmermann et al.(93.6%) [10] trained and tested on the same dataset TotalCapture [31]. It is worth mentioning that if the model is trained on XsensMotion, the performance is better (97%). XsensMotion has over 1000 seconds of data with around 30 subjects and the training set of TotalCapture is 500 seconds on 3 subjects which could explain the inferior performance. Besides the high accuracy, our model is flexible in IMU number. Weenk et al.'s work achieves the full-body accuracy of 97.5% but depends on multi-sensor features (correlation coefficients of a sensor with all other sensors, etc.) so "the sensor configuration needs to be known" as they said in their paper. Otherwise, without features like ranking and correlation coefficients, only 75.9% of the sensors' classification results are correct. [8]. Kaichi et al. depend on the root sensor and a global feature learned from all IMUs, which makes it impossible to use only part of the full set in the test stage (calibration for the users).

Generally speaking, using rotation quaternion does not efficiently enhance performance. The usage of the rotation quaternion can be divided into two types: one attaches it directly to the 6-channel input, and the other uses the orientation information to realize root sensor-based heading correction. Putting together the validation and test performance, the inferior accuracy rates of the 10-channel 2018 model, HCFM model with and without WMF can be the result of overfitting on the validation dataset. At the same time, the 10-channel input setting is better than 6-channel referring to the heading correction method. Heading correction method rotates all IMUs to let the subject (based on the pelvis) face the direction towards the y-axis and pollutes the non-root information unintentionally while using rotation quaternions under these circumstances compensates for the loss thus increasing the accuracy rate.

On different segments, the proposed model (HCFM+WMF) achieves unequal prediction accuracy respectively. Generally speaking, the lower-body configuration has the best performance, followed by the full body and finally the upper body according to the prediction accuracy. Because the upper body is adjusted to remain stable by the central nervous system during the bipedal gait of humans [65]. The lack of motions damages the feature salience of the inertial data and causes inferior upper-body performance compared with lower-body performance as in Table4.1. Each segment has been involved in two configurations (pelvis is in all three con-

figurations). The full body is the combination of the lower and upper body thus it has intermediate performance. From the point of view of the segment types, IMUs mounted on feet are the least likely to be mislabeled, as expected. These two IMUs are fixed on the instep horizontally, while the rests are vertically attached to the body segments.

Meanwhile, the shoulders achieve an accuracy of less than 55% according to Figure 4.4 and l/r shoulder switch mistakes account for the vast majority of mislabelling related to shoulders for all the models. It seems that the proposed model is purely guessing between the left and right sides. However, as shown in Figure 4.11, the accuracy of close to 50% is caused by the average of three subjects with high accuracy of over 90% and two with very low accuracy of nearly 0%. This can be explained by the particularity of the shoulder configuration. The IMUs are attached close to the coracoid process of the scapula on the upper body, and they are the only paired sensors on the trunk. Additionally, the differences between the unilateral limbs' periodical activities are crucial for side recognition. However, the trunk has no noticeable pendular activity as upper limbs do and no periodical stance and swing motions as lower limbs do [66]. Thus, the lack of distinguishable movements between left and right shoulders reduces the accuracy greatly, so probably only on subjects with certain gaits the l/r shoulders can be well recognized. However, it is not clear what kind of common features the subjects with high shoulder recognition accuracy have. To determine the reason for this, further exploration would be necessary.

Three types of testing trials were recorded. Trials containing walking in infinity trajectory outperform the others. Infinity walking contains more turnings than standard and abnormal trials. The latter ones include fewer U-turns per trial which implies the idea that turnings are informative as investigated in previous research [67]. Abnormal trials, as expected, include many motions absent from the training dataset. The training set mainly contains motion types of walking, turning+walking, slower down or accelerating and N-pose (standing still).

Different prediction performance among subjects indicates identical human gaits. Except for the nearly contrary performance related to the l/r shoulder recognition, dissimilar features are observed in other segments as well. As an example, in Figure 4.10a and Figure 4.10b it could be observed that most of the corresponding segments have a close accuracy rate. However, their mislabelled predictions are different. For example, the left upper arm of Subject-2 is mistakenly assigned to the right upper arm (5%), left forearm(1%) and T8 (1%), however, the incorrectly predicted labels of Subject-5 are the right upper arm (4%), right forearm(2%). What's more, the proposed model has problems recognizing lower legs on Subject-5 (accuracy of 93% for both sides), but it achieves 100% lower leg accuracy on Subject-2.

The dissimilarity of the human gaits thus proves the necessity of involving various subjects in the training set.

In this research, majority voting has greatly improved the performance of the model. It raises the window-wise accuracy of the non-shoulder segments to 100% for most of the test subjects (except for Subject-1 with 3 failed predictions out of 450). It also bridges the gap between different models making it more tolerant of abnormal trials as shown in Figure 4.8. Since the limited scenario of this study is the calibration phase using the wearable IMU suit, the model only needs to answer the whole process. Thus it improves the fault tolerance of the model: as long as the window-wise prediction accuracy is above 50%, the accuracy rate on this segment will be improved to 100% in the end. This allows models with slightly inferior performance to be trained using lenient early stopping criteria. For instance, quick model retraining within 10 epochs is possible, as long as the accuracy rates are higher than 50% (majority voting can give correct prediction). In this case, the users are provided with real-time feedback, which can be commercially important to the company, as the current speed for testing is already quite fast (Appendix B.4).

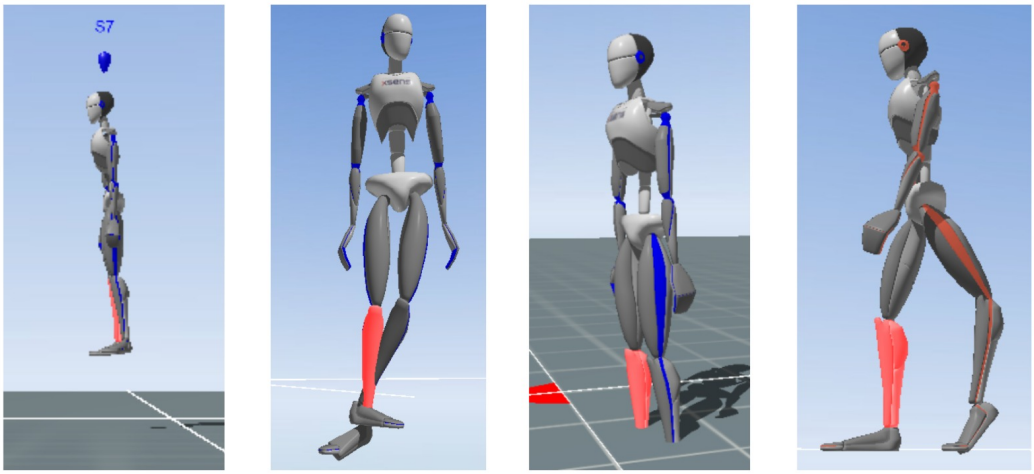


Figure 5.1: Some examples of strange visualization of the training set.

Additionally, the training dataset is imperfect as Figure 5.1 shows. The visualization flaws imply possible misplacement of the IMUs, though to maintain the variety and robustness of the dataset, we kept some of the trials (first three trials in Figure 5.1, etc.). However, it is difficult to detect the faults that do intervene with the model training (the last subplot in Figure 5.1). Among those human-made mistakes, switching around IMUs is easier to recognize by checking the property file and unique hardware numbers for the involved IMUs. But it becomes harder when the Awinda set used for data recording is unavailable to the researcher. The orientation and slight displacement can not be recognized currently. Ensuring that data

is 100% compliant without human errors is hard. It results for three reasons: firstly, the I2S relative position and orientation have a crucial impact on the prediction, but the minor deflection and displacement are hard to notice; secondly, users who are unfamiliar with the system can ignore obvious switches of two or more IMUs. With the misplacement, the calibration and visualization programs stillhave a chance of success. Currently, it is almost impossible to go back and check the compliance of the sensor data after the recording stage. For the obvious misplacement, reviewing the visualization is efficient, but the software is unable to detect the minor orientation rotation and displacement of the IMU. Those noises are kept in the dataset, though on the other hand, can help with the robustness of the model because the users also make mistakes during the calibration.

This model could serve as a misplacement recognizer or anomaly detector for the recorded data. The used program now only informs the user to restart after the calibration fails without telling the cause of the error. If the certainty of the prediction is too low (10%, etc.), the I2S location and orientation of the specific IMU may be undesirable. The low accuracy rates compared to validation are also caused by abnormal motions during calibration trials as mentioned above including running and jumping. The original XsensMotion set also includes running trials which are filtered out by watching all the visualization in person. In this case, for inaccurate predictions, the program can output a reminder asking the user to check specific segments instead of only informing the user to restart after the calibration failure as the program in use now does.

Chapter 6

Conclusions

6.1 Conclusions

In this work, we proved that it is possible to apply CNN+GRU model on automatic I2S labelling tasks trained on walking motion dataset achieving comparable accuracy to the previous research while keeping the flexibility in testing the different amount of IMUs. For **RQ1**, the baseline model [7] with the same kernel setting for all convolutional layers is outperformed by HCFM model with distinct hyperparameters for each convolutional layer. The performance of the HCFM model trained and tested on pure walking motions is better than using the whole trial including the extra 5 seconds' N-pose. Therefore, for **RQ2**, it has been proved that the WMF method enhances the performance of HCFM model, for example, to 98.6% on the lower body. Generally speaking, adding rotation quaternion in input or utilizing it for heading correction can not improve the overall performance (**RQ3**). For the first time, the paired IMUs on shoulders are studied and proved to be unrecognizable for all CNN+GRU models mentioned as nearly full left and right shoulder switch takes place on specific subjects. Concerning the motion type of the test trial, infinity walking is tested to be the best motion with the highest accuracy rate. As the answer to **RQ4**, after majority voting, in most trials, the non-shoulder segments are about 100% correctly assigned. The real-time and high-accuracy I2S assignment achieved by the proposed model indicates the possibility of the imminent application on commercial products and being an anomaly detector of the recorded trial in near the future.

6.2 Future Work

According to the results, each model has its ability to learn certain features. Therefore they have different abilities to predict different segments. Based on the performance among distinct models on the validation set, we can train a set of coefficient

matrices that give the most accurate results utilizing all models rather than selecting the best model as the sole source of information.

The knowledge of the training set is limited. It is known that the involved subjects are the employees of the company (healthy people aged between 20 to 50), but the specific information (height, gender, etc.) of the subjects is unavailable in the training dataset. There are also some irregular motions in the training set, but no related statistics are available. Meanwhile, the test set with five subjects could be biased. To verify if the model is feasible for all users, a larger number of subjects should be involved in training and test sets in the future.

The results indicate that appending rotation quaternion to the commonly used acceleration and angular velocity triggers overfitting the validation set. Future work can be done to determine appropriate approaches to exploit orientation information.

The proposed model learns the specific l/r shoulder features for some people but meanwhile makes l/r switch mistakes for the other subjects. It is crucial to find out if this is due to individual differences in gait. According to some test experiments, the stimulated rotation of the left shoulder impacts the performance of the model more, though it does not influence the results of majority voting significantly if the rotation is 180 degrees. In the future, the model can be retrained with different sliding window sizes and sampling rates other than 60Hz, for example. If sacrificing some flexibility is acceptable, the dependency between shoulders and other body segments, for example, forearms, can be considered.

In the test experiment, the performance of the pure CNN network is close to CNN+GRU model (the former is 6% and 2% lower in accuracy with lower-body and full-body configuration respectively). The former has less than half the training time of the latter. This gives reason to choose a time-efficient model for example pure CNN without GRU layer facing the cost-performance trade-off. Research could be done to further study the efficiency of the model to provide the ability to retrain the model.

Currently, using IMU number less than the trained model in the test stage is possible. However, as the cost function of optimal linear sum assignment is used to improve the accuracy, the sensor configuration has to be known. Without using the optimization which can prevent assigning two or more IMUs to the same segment, the performance of the proposed model decreases to 84.4% and 92.2% (after majority voting) with full-body configuration. If an unknown configuration is required, future research needs to especially increase the accuracy of some segments (upper arms and hands, etc.) with notable decreases without the optimization function.

Bibliography

[1] M. Rana and V. Mittal, "Wearable sensors for real-time kinematics analysis in sports: a review," IEEE Sensors Journal, vol. 21, no. 2, pp. 1187–1207, 2020.

[2] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A review of wearable sensors and systems with application in rehabilitation," Journal of neuroengineering and rehabilitation, vol. 9, no. 1, pp. 1–17, 2012.

[3] K. Kim, M. Billinghurst, G. Bruder, H. B.-L. Duh, and G. F. Welch, "Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017)," IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 11, pp. 2947–2962, 2018.

[4] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors," Xsens Motion Technologies BV, Tech. Rep, vol. 1, pp. 1–7, 2009.

[5] N. Pannurat, S. Thiemjarus, E. Nantajeewarawat, and I. Anantavrasilp, "Analysis of optimal sensor positions for activity classification and application on a different data collection scenario," Sensors, vol. 17, no. 4, p. 774, 2017.

[6] I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay, "Optimal placement of accelerometers for the detection of everyday activities," Sensors, vol. 13, no. 7, pp. 9183–9200, 2013.

[7] T. Zimmermann, B. Taetz, and G. Bleser, "Imu-to-segment assignment and orientation alignment for the lower body using deep learning," Sensors, vol. 18, no. 1, p. 302, 2018.

[8] D. Weenk, B.-J. F. Van Beijnum, C. Baten, H. J. Hermens, and P. H. Veltink, "Automatic identification of inertial sensor placement on human body segments during walking," Journal of neuroengineering and rehabilitation, vol. 10, no. 1, pp. 1–9, 2013.

[9] M. Paulich, M. Schepers, N. Rudigkeit, and G. Bellusci, “Xsens mtw awinda: Miniature wireless inertial-magnetic motion tracker for highly accurate 3d kinematic applications.”

[10] T. Kaichi, T. Maruyama, M. Tada, and H. Saito, “Learning sensor interdependencies for imu-to-segment assignment,” IEEE Access, vol. 9, pp. 116 440–116 452, 2021.

[11] K. Kunze and P. Lukowicz, “Sensor placement variations in wearable activity recognition,” IEEE Pervasive Computing, vol. 13, no. 4, pp. 32–41, 2014.

[12] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster, “Where am i: Recognizing on-body positions of wearable sensors,” in Location- and Context-Awareness, T. Strang and C. Linnhoff-Popien, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 264–275.

[13] L. H. Grokop, A. Sarah, C. Brunner, V. Narayanan, and S. Nanda, “Activity and device position recognition in mobile devices,” in Proceedings of the 13th International Conference on Ubiquitous Computing, ser. UbiComp ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 591–592. [Online]. Available: https://doi.org/10.1145/2030112.2030228

[14] L. Wang, T. Tan, H. Ning, and W. Hu, “Silhouette analysis-based gait recognition for human identification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pp. 1505–1518, 2003.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652–660.

[17] MoCap, “Cmu graphics lab motion capture database,” http://mocap.cs.cmu.edu.

[18] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Deep learning for time series classification: a review,” Data mining and knowledge discovery, vol. 33, no. 4, pp. 917–963, 2019.

[19] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” in 2017 International joint conference on neural networks (IJCNN). IEEE, 2017, pp. 1578–1585.

[20] J. C. B. Gamboa, "Deep learning for time-series analysis," arXiv preprint arXiv:1701.01887, 2017.

[21] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," Data mining and knowledge discovery, vol. 31, no. 3, pp. 606–660, 2017.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[23] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series." in CCIA, 2018, pp. 120–129.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in European conference on computer vision. Springer, 2016, pp. 630–645.

[25] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," Data Mining and Knowledge Discovery, vol. 34, no. 5, pp. 1454–1495, 2020.

[26] L. H. Grokop, A. Sarah, C. Brunner, V. Narayanan, and S. Nanda, "Activity and device position recognition in mobile devices," in Proceedings of the 13th international conference on Ubiquitous computing, 2011, pp. 591–592.

[27] D. F. Silva, R. Giusti, E. Keogh, and G. E. Batista, "Speeding up similarity search under dynamic time warping by pruning unpromising alignments," Data Mining and Knowledge Discovery, vol. 32, no. 4, pp. 988–1016, 2018.

[28] D. Graurock, T. Schauer, and T. Seel, "Automatic pairing of inertial sensors to lower limb segments – a plug-and-play approach," Current Directions in Biomedical Engineering, vol. 2, no. 1, pp. 715–718, 2016. [Online]. Available: https://doi.org/10.1515/cdbme-2016-0155

[29] B. Bouvier, S. Duprey, L. Claudon, R. Dumas, and A. Savescu, "Upper limb kinematics using inertial and magnetic sensors: Comparison of sensor-to-segment calibrations," Sensors, vol. 15, no. 8, pp. 18 813–18 833, 2015.

[30] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," Sensors, vol. 16, no. 1, p. 115, 2016.

[31] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, "Total capture: 3d human pose estimation fusing video and inertial sensors," in Proceedings of 28th British Machine Vision Conference. University of Surrey, 2017, pp. 1–13.

[32] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," IEEE Sensors Journal, vol. 17, no. 2, pp. 386–403, 2017.

[33] A. Mannini, S. Intille, M. Rosenberger, and A. Sabatini, "Activity recognition using a single accelerometer placed at the wrist or ankle," Medicine and science in sports and exercise, vol. 45, 04 2013.

[34] V. Hernandez, T. Suzuki, and G. Venture, "Convolutional and recurrent neural network for human activity recognition: Application on american sign language," PloS one, vol. 15, no. 2, p. e0228869, 2020.

[35] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," in Workshops at the Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[36] T. McCandless and K. Grauman, "Object-centric spatio-temporal pyramids for egocentric activity recognition." in BMVC, vol. 2, 2013, p. 3.

[37] K. Kunze and P. Lukowicz, "Using acceleration signatures from everyday activities for on-body device location," in 2007 11th IEEE International Symposium on Wearable Computers, 2007, pp. 115–116.

[38] A. Mannini, A. M. Sabatini, and S. S. Intille, "Accelerometry-based recognition of the placement sites of a wearable sensor," Pervasive and mobile computing, vol. 21, pp. 62–74, 2015.

[39] R. Saeedi, B. Schimert, and H. Ghasemzadeh, "Cost-sensitive feature selection for on-body sensor localization," in Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, 2014, pp. 833–842.

[40] S. Lambrecht, J. P. Romero, J. Benito-León, E. Rocon, and J. L. Pons, "Task independent identification of sensor location on upper limb from orientation data," in 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2014, pp. 6627–6630.

[41] N. Amini, M. Sarrafzadeh, A. Vahdatpour, and W. Xu, "Accelerometer-based on-body sensor localization for health and medical monitoring applications," Pervasive and mobile computing, vol. 7, no. 6, pp. 746–760, 2011.

[42] G. Dunton, E. Dzubur, K. Kawabata, B. Yanez, B. Bo, and S. Intille, "Development of a smartphone application to measure physical activity using sensor-assisted self-report," Frontiers in Public Health, vol. 2, 2014. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpubh.2014.00012

[43] K. Alanezi and S. Mishra, "Impact of smartphone position on sensor values and context discovery," University of Colorado: Denver, CO, USA, 2013.

[44] T. Franke, P. Lukowicz, K. Kunze, and D. Bannach, "Can a mobile phone in a pocket reliably recognize ambient sounds?" in 2009 International Symposium on Wearable Computers, 2009, pp. 161–162.

[45] S. Lambrecht and J. L. Pons, "Automatic identification of sensor localization on the upper extremity," in XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013. Springer, 2014, pp. 1497–1500.

[46] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," Data Mining and Knowledge Discovery, vol. 29, no. 3, pp. 565–592, 2015.

[47] P. Schäfer, "The boss is concerned with time series classification in the presence of noise," Data Mining and Knowledge Discovery, vol. 29, no. 6, pp. 1505–1530, 2015.

[48] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A time series forest for classification and feature extraction," Information Sciences, vol. 239, pp. 142–153, 2013.

[49] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote: the collective of transformation-based ensembles," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 9, pp. 2522–2535, 2015.

[50] B. Lucas, A. Shifaz, C. Pelletier, L. O'Neill, N. Zaidi, B. Goethals, F. Petitjean, and G. I. Webb, "Proximity forest: an effective and scalable distance-based classifier for time series," Data Mining and Knowledge Discovery, vol. 33, no. 3, pp. 607–635, 2019.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, 2012.

[52] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[53] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The ucr time series archive," IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 6, pp. 1293–1305, 2019.

[54] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in Twenty-fourth international joint conference on artificial intelligence, 2015.

[55] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," Pattern Recognition Letters, vol. 42, pp. 11–24, 2014.

[56] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in International conference on machine learning. PMLR, 2013, pp. 1310–1318.

[57] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "Gvcnn: Group-view convolutional neural networks for 3d shape recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 264–272.

[58] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package)," Neurocomputing, vol. 307, pp. 72–77, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218304843

[59] K. Förster, P. Brem, D. Roggen, and G. Tröster, "Evolving discriminative features robust to sensor displacement for activity recognition in body area sensor networks," in 2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). IEEE, 2009, pp. 43–48.

[60] M. Schepers, M. Giuberti, G. Bellusci et al., "Xsens mvn: Consistent tracking of human motion using inertial sensing," Xsens Technol, vol. 1, no. 8, 2018.

[61] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster, "Where am i: Recognizing on-body positions of wearable sensors," in International Symposium on Location-and Context-Awareness. Springer, 2005, pp. 264–275.

[62] M. Ben-Ari, "A tutorial on euler angles and quaternions," Weizmann Institute of Science: Rehovot, Israel, 2014.

[63] H. Cho and S. M. Yoon, “Divide and conquer-based 1d cnn human activity recognition using test data sharpening,” Sensors, vol. 18, no. 4, p. 1055, 2018.

[64] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, “On-body device localization for health and medical monitoring applications,” in 2011 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2011, pp. 37–44.

[65] J. J. Kavanagh, “Lower trunk motion and speed-dependence during walking,” Journal of neuroengineering and rehabilitation, vol. 6, no. 1, pp. 1–10, 2009.

[66] N. Anang, R. Jailani, N. M. Tahir, H. Manaf, and N. Mustafah, “Analysis of kinematic gait parameters in stroke with diabetic peripheral neuropathy (dpn),” in 2016 IEEE Conference on Systems, Process and Control (ICSPC). IEEE, 2016, pp. 136–141.

[67] T. Imai, S. T. Moore, T. Raphan, and B. Cohen, “Interaction of the body, head, and eyes during walking and turning,” Experimental brain research, vol. 136, no. 1, pp. 1–18, 2001.

Appendix A: methods

A.1 Rotation quaternion

The fluctuation of orientation around 180(-180)°can be solved by using rotation quaternion to replace the 3-axis orientation as shown in Figure A.1. The orientation triplet (a_x, a_y, a_j) consists of three angles ranging from -180 °to 180 °indicating the rotation respectively projected to three planes perpendicular to the x, y and z-axis. Quaternion, instead, can represent the orientation with an Euler axis and the rotation angle of that axis.

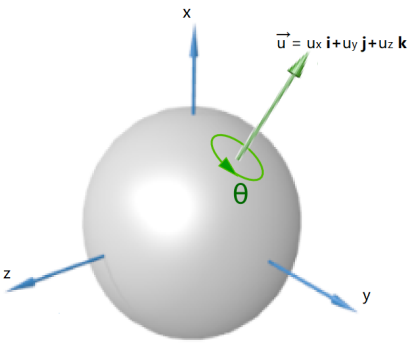


Figure A.1: The representation of a rotation in 3D sphere: Euler axis ($\hat{u}$) by an angle of θ.

Since the mathematical proof process is not the focus of this project, this part will be omitted. Only important steps will be introduced. As shown in Figure A.1, $\vec{u}$ is a unit vector:

$$\vec{u} = (u_x, u_y, u_z) = u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}$$

Let q denote the extension of Euler's formula:

$$\mathbf{q} = e^{\frac{\theta}{2}(u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})} = \cos\frac{\theta}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2}$$

Quaternion is $(C, X\ S, Y\ S, Z\ S)$, where $C = \cos(\theta/2)$, $S = \sin(\theta/2)$, and $X = u_x\mathbf{i}$, $Y = u_y\mathbf{j}$, $Z = u_z\mathbf{k}$. Using this notation, the value of orientation changes smoothly during the trial.

To our best knowledge, there is no closely related work considering the usage of orientation information. Therefore, the format of rotation quaternion is adopted due to its feature of gradual change in critical situations. The rotation quaternion is used in three ways: firstly it is directly appended to the input with acceleration and angular velocity on the right side; the second way is exploiting its information in heading correction; the third way is combining the previous two methods.

Appendix B: Results

B.1 Input methods comparison

Table B.1: The prediction accuracy changes after adding rotation quaternion in (namely the result of accuracy rate of the 10-channel model minus the corresponding 6-channel model). The red triangle pointing down means the performance is poorer using rotation quaternion; the green triangle pointing up means better performance.

No.	Accuracy type	Lower body	Upper body	Full body
Exp.1	window-wise	-0.036▼	-0.036▼	-0.024▼
	trial-wise	0	-0.018▼	-0.004▼
Exp.2	window-wise	-0.059▼	-0.056▼	-0.046▼
	trial-wise	-0.010▼	-0.018▼	0
Exp.3	window-wise	-0.007▼	0	-0.011▼
	trial-wise	0	0	0.002▲
Exp.4	window-wise	0.035▲	0.031▲	-0.010▼
	trial-wise	0	-0.053▼	-0.056▼

B.2 Model convergence speed and overfitting

In this project, we set early-stop patience of 100 epochs, and the training will be terminated if there is no further improvement in validation loss. To sum up, the 2018 model converges faster and the validation loss starts to increase earlier than HCFM model. Using heading correction increases the validation loss compared to the other models. Adding rotation quaternion in also makes the model overfits sooner as shown in Figures B.1:

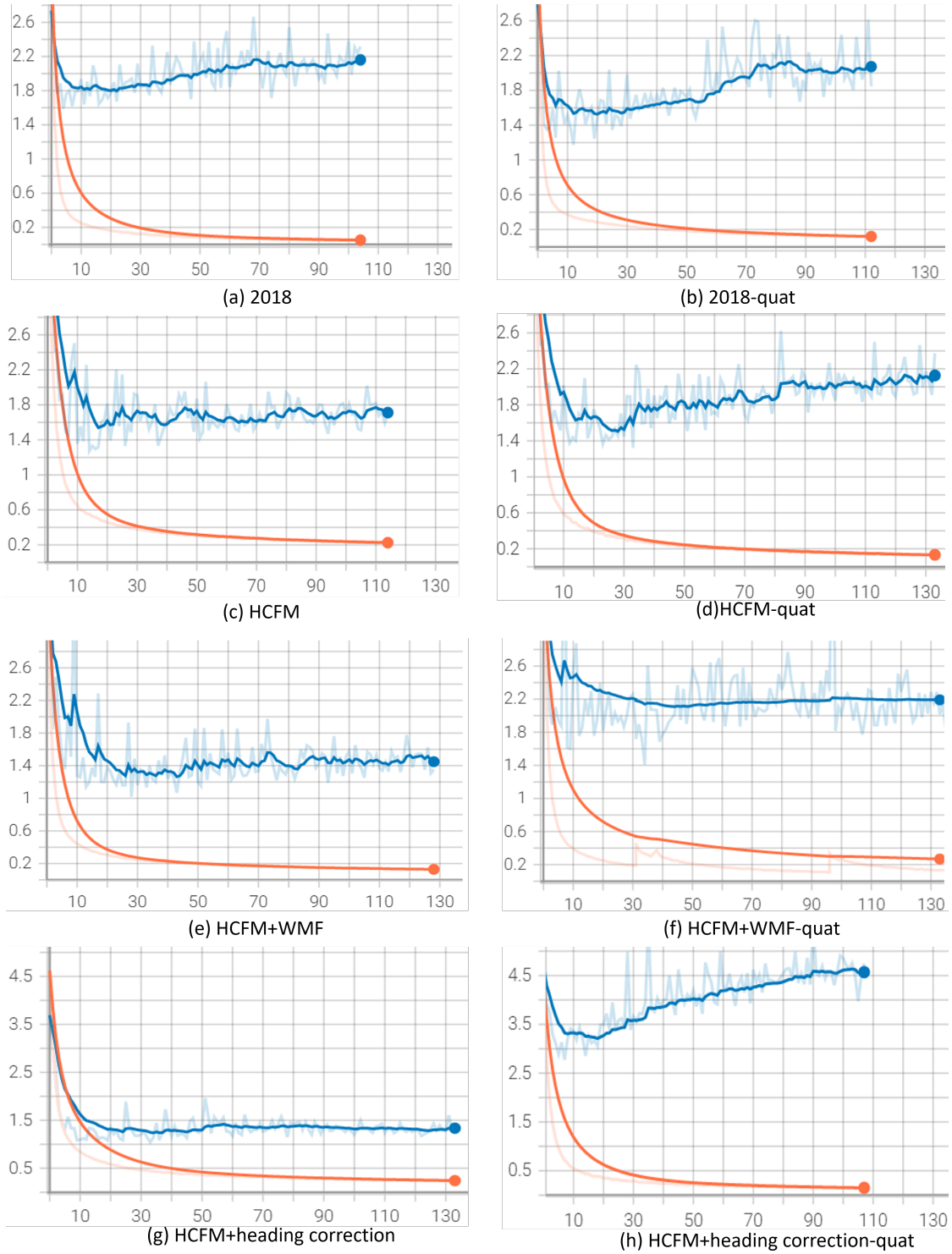
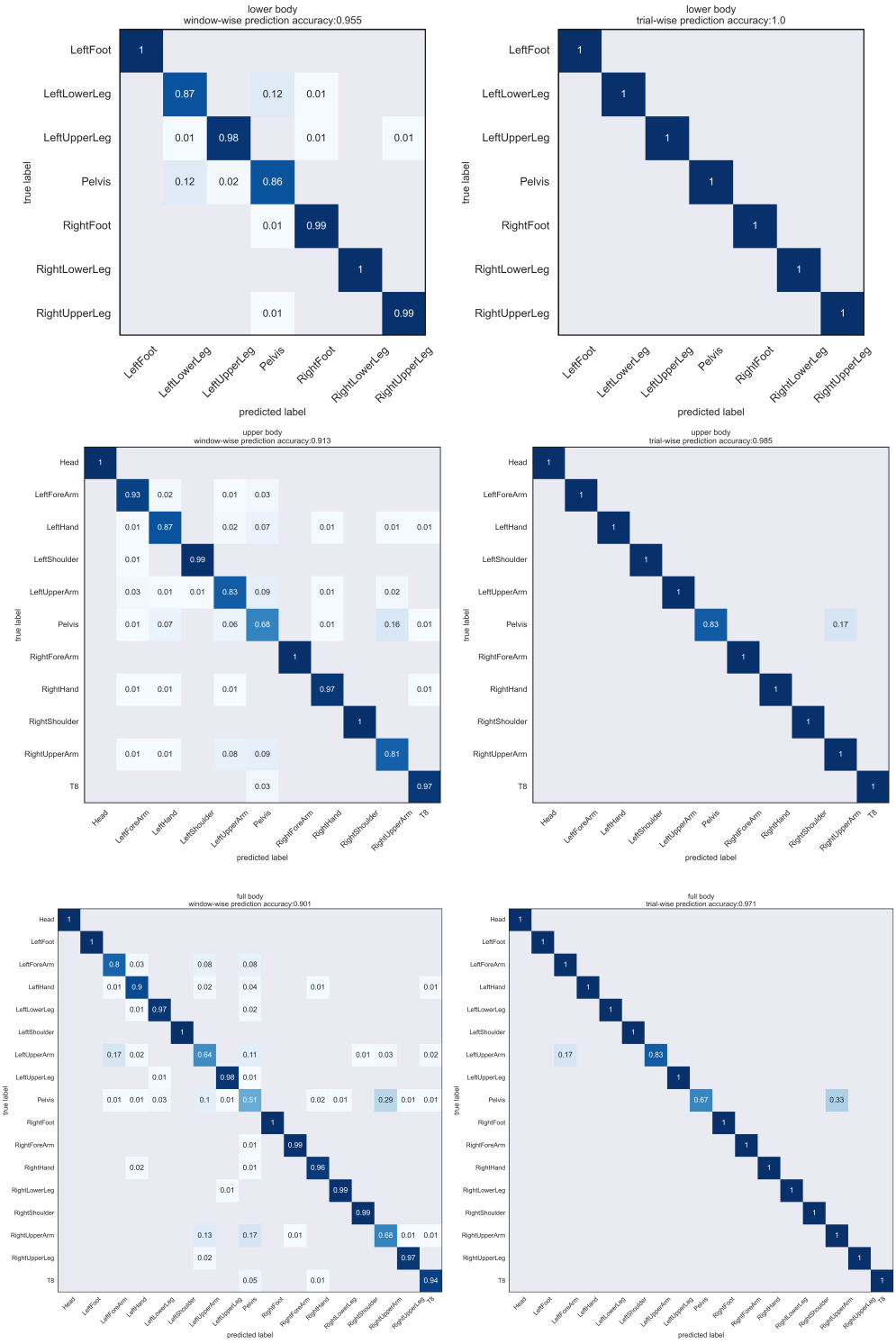


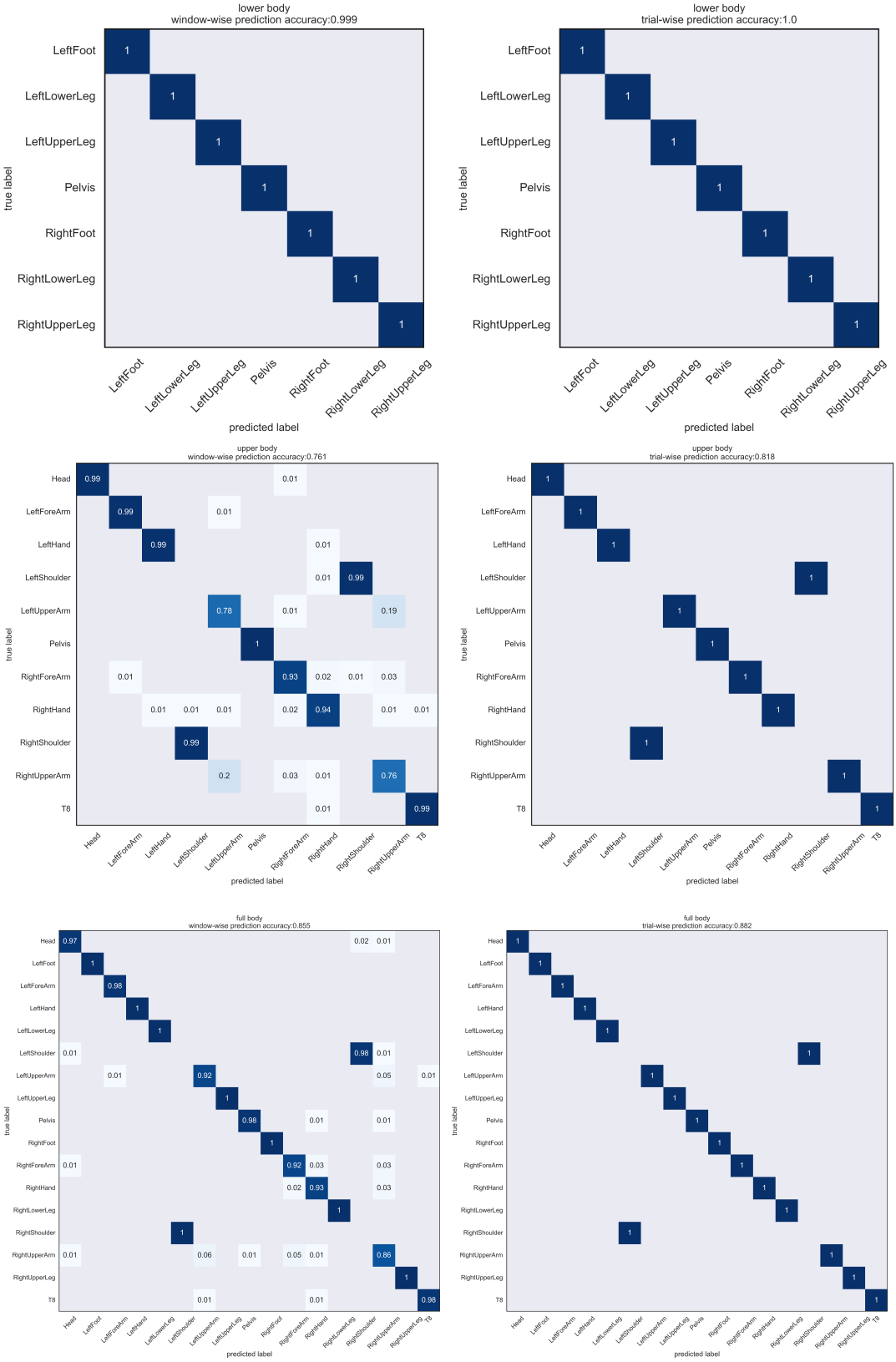
Figure B.1: The loss(y-axis)-epoch(x-axis) plot. The blue curve is for the validation loss, the orange curve is for the training loss. Darker ones are normalized curves. All plots are with full-body configuration.

B.3 Confusion matrices of 6-channel HCFM with WMF grouped by subjects



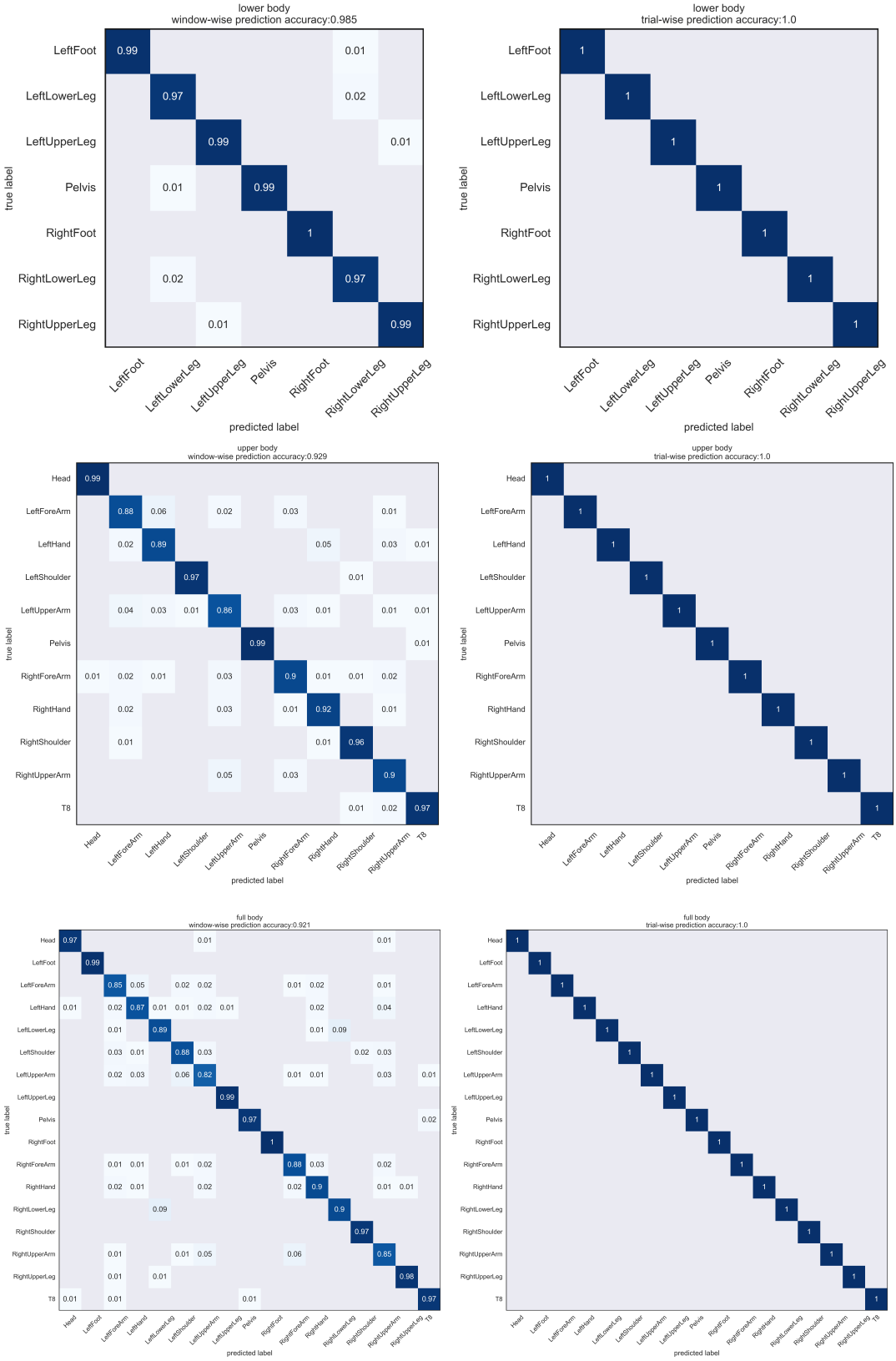
(a) Window-wise prediction. *(b)* Trial-wise prediction.

Figure B.2: Subject1



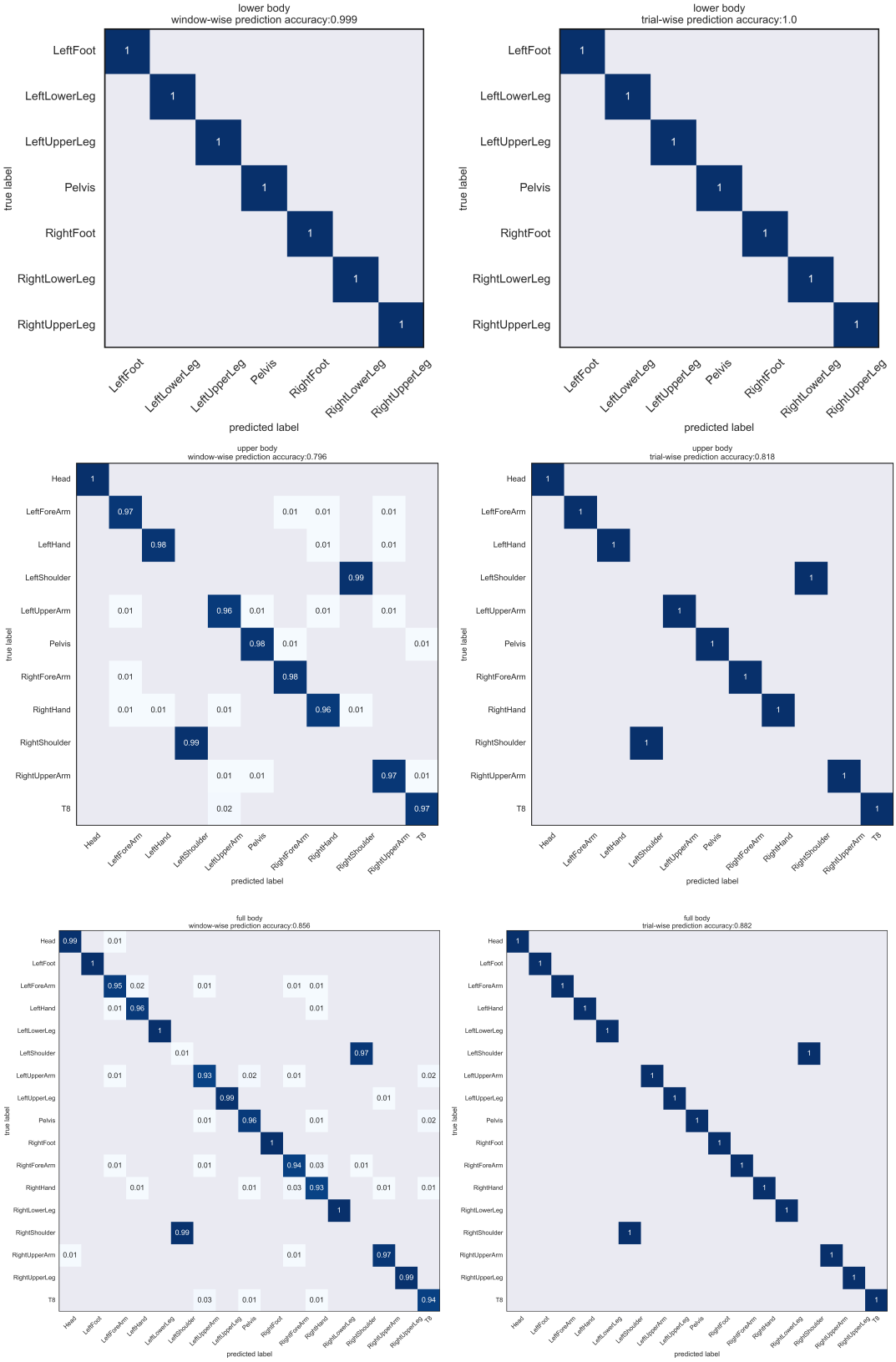
(a) Window-wise prediction. *(b)* Trial-wise prediction.

Figure B.3: Subject2



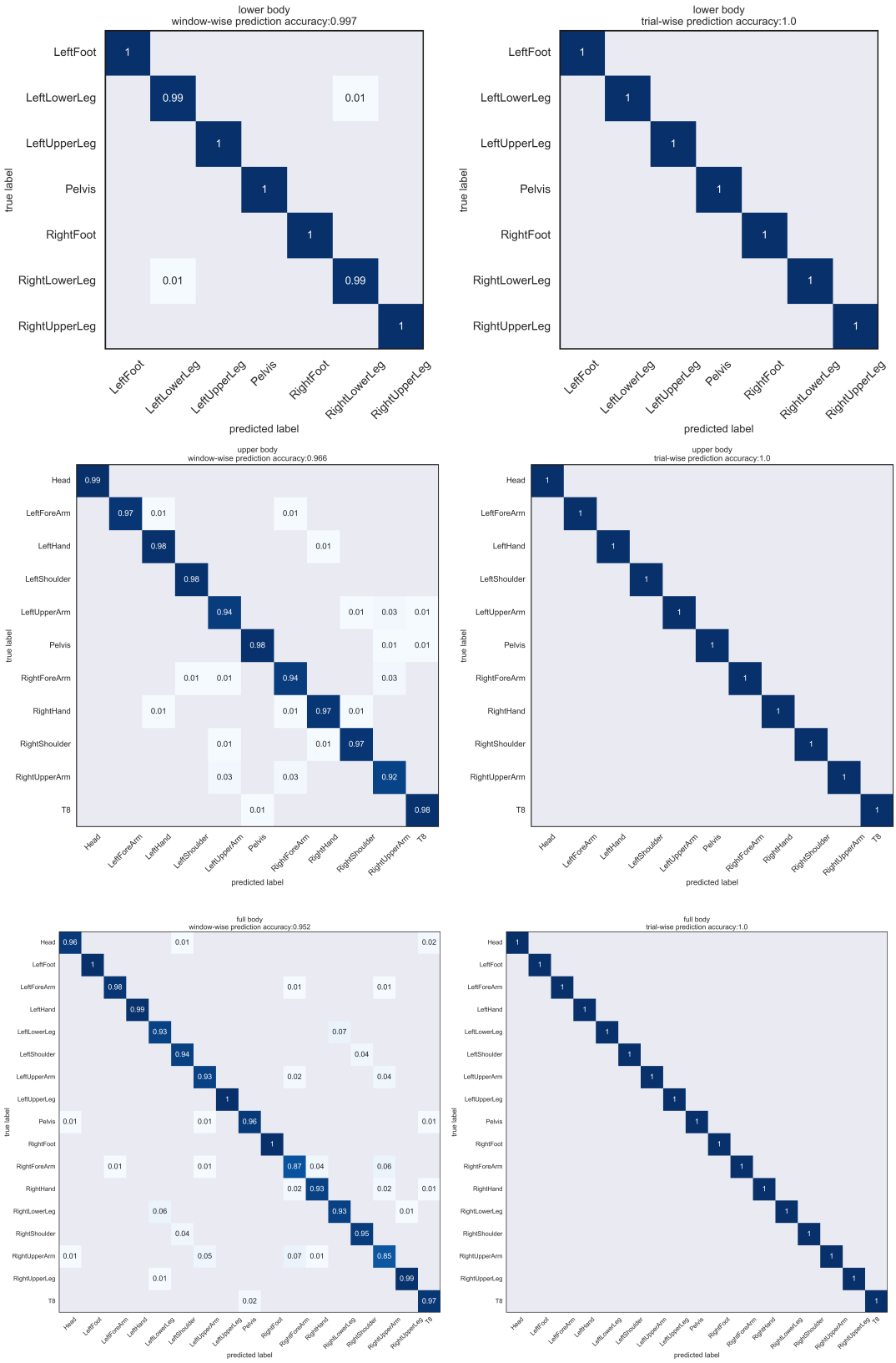
(a) Window-wise prediction. *(b)* Trial-wise prediction.

Figure B.4: Subject3



(a) Window-wise prediction. *(b)* Trial-wise prediction.

Figure B.5: Subject4



(a) Window-wise prediction. *(b)* Trial-wise prediction.

Figure B.6: Subject5

B.4 Computational performance of the test phase

Table B.2: Runtime of the testing phase of the proposed model (6-channel HCFM with WMF).

segment configurations (IMUs number)	**model loading(s)**	**testing (s)**	**total runtime(s)**	**average runtime per trial(s)**
lower body (7)	3.078	18.547	21.625	0.721
upper body (11)	3.633	29.574	33.206	1.107
full body (17)	4.143	75.373	79.517	2.512