# 1 LABELLING HUMAN KINEMATICS DATA USING CLASSIFICATION MODELS

- 2 February 18<sup>th</sup>, 2022
- 3 Machine Learning Classification models can produce highly accurate sensor labels for
- 4 motion data captured in the motion capture studio
- 5
- 6 Yuan Shi,<sup>1</sup> Nihir Chadderwala,<sup>2</sup> Ujjwal Ratan<sup>3</sup>
- 7 <sup>1</sup> Amazon Web Services, Singapore, Singapore
- 8 <sup>2</sup>Amazon Web Services, Dallas, TX, USA
- <sup>9</sup> <sup>3</sup>Amazon Web Services, Seattle, WA, USA
- 10
- 11 Conflict of Interest Disclosure: None.
- 12
- 13 Correspondence Address:
- 14 Ujjwal Ratan
- 15 Amazon Web Services,
- 16 2205 7<sup>th</sup> Avenue,
- 17 Seattle, WA 98121
- 18 Phone: (206) 615-5400, Email: ujjwalr@amazon.com
- 19
- 20
- 21
- 22

# 23 Abstract

24 The goal of this study is to develop a classification model that can accurately and 25 efficiently label human kinematics data. Kinematics data provides information about the 26 movement of individuals by placing sensors on the human body and tracking their 27 velocity, acceleration and position in three dimensions. These data points are available in 28 C3D format that contains numerical data transformed from 3D data captured from the 29 sensors. The data points can be used to analyse movements of injured patients or patients 30 with physical disorders. To get an accurate view of the movements, the datasets generated by the sensors need to be properly labelled. Due to inconsistencies in the data capture 31 32 process, there are instances where the markers have missing data or missing labels. The 33 missing labels are a hindrance in motion analysis as it introduces noise and produces 34 incomplete datapoints of sensor's positioning in 3 dimensional space. Labelling the data 35 manually introduces substantial effort in the analysis process. In this paper, we will 36 describe approaches to pre-process the kinematics data from its raw format and label the 37 data points with missing markers using classification models.

- 39
- 40
- 41
- 42
- 43

44

45

<sup>38</sup> *Keywords:* machine learning, c3d, biomechanics, classification model

4	6
4	6

### Introduction

47 Biomechanics refer to the study of mechanical laws relating to the movement of living 48 organisms.(1) When applied to humans and combined with kinematics, it helps us capture 49 data which is used to infer a variety of human actions. For example, kinematics data is 50 used to understand human's movements in each gesture, through providing quantitative 51 data to evaluate each individual's flexibility(2). It is also used to analyse patients' 52 movements to track their recovery and assist towards rehabilitation(3). Kinematics data is 53 captured in dedicated labs setup as motion capture studios. In some cases, the sensor data 54 may not be labelled completely. This can happen for a variety of reasons like the sensor 55 not being attached securely or some sensors getting blocked from a camera from certain 56 angles. This introduces noise and missing values in the dataset and poses a challenge for 57 further analysis of this data. Capturing movements of these patients and labelling them 58 can be very tedious and an expensive process. To solve this problem, we describe a 59 method where we use the labelled data from these sensors for training a classification 60 model that can then label the unlabelled sensor records with high accuracy, thereby 61 reducing manual efforts.

The primary objective of this study is to use a classification model to label the sensor data corresponding to the mounted location on the human subject's body. We took the records with pre-labelled information and built classification models that would identify the right class (sensor label) for the unlabeled dataset. In this study, we firstly clean and standardize the raw C3D files through data transformation functions. We then use the processed data to train 4 different machine learning models to classify the sensor data points into one of the multi-class labels. Lastly, we evaluate the performance of the

models to auto generate labels with test dataset. Interpretation and future work isprovided at the end of the paper.

71

## Method

72 In this section, we go into the details of our method starting with describing the raw 73 dataset, data processing and feature engineering, modelling approaches including the 74 choice of algorithms and the evaluation methods.

75

## 76 Data Description

77 For the purposes of this study, we downloaded the raw data from CMU Graphics Lab 78 Motion Capture Database in C3D format(4), which is a file format that has been widely 79 used in Biomechanics, Animation and Gait Analysis laboratories to record synchronized 80 3d kinematics data. It contains information needed to read, display, and analyse 3d81 motion data and additional analog data from force plates, electromyography, 82 accelerometers, and other sensors. The dataset contains numerical data extracted from 83 sensors attached to the human body. It is composed of a list of time-series points. Each 84 point is composed of x, y and z co-ordinates, time of capturing, frame number and the 85 labeled location. The choice of number of sensors and locations mounted vary in each 86 setting as it subjects to the purpose of specific biomedical study. For more details, readers 87 can refer to CMU Graphics Lab Motion Capture Database (4) to visualize how the 88 sensors are mounted on the human body to capture motion data.

89

90 The *C3D* dataset being captured can be converted into a standard dataframe (Table 1)
91 using the *c3d* python library(5).

time	X	У	Z	cam	err	frame	point_label
1.0125	0	0	0	FALSE	FALSE	1	R_ASIS
1.0125	0	0	0	FALSE	FALSE	1	L_ASIS
1.0125	0	0	0	FALSE	FALSE	1	SACRUM
1.0125	0	0	0	FALSE	FALSE	1	R_THIGH_1
1.0125	0	0	0	FALSE	FALSE	1	R_THIGH_2

### 92 Table 1: C3D Sample File in DataFrame

93 Note: This table captures the structure of a raw C3D file when its converted into a dataframe.

94

95 As shown in Table 1, a typical C3D dataset has 8 features, correspond to time, , x, y, z, 96 *cam, err, frame and point\_label.* Here, *time* refers to the capture time of the point starting 97 from 0, x, y and z correspond to captured location at x, y and z axis respectively. cam 98 suggests if there is any camera observing the sensor. To ensure accuracy of captured data, 99 it is required to ensure that at least one camera is observing a sensor at the indicated time. 100 Otherwise, the data point is advised to be removed from the dataset. err suggests whether 101 there is error in capturing the 3d location at this frame, and *frame* refers to the time frame 102 of the current position with continuous integer starting from 1. Point labels are the 103 labelled targets which indicate the location where the sensor is attached to. 104 In our experiment, we downloaded a collection of C3D files of *subject #26* provided by

105 Qualisys illustrating human gait(6). Among all the collections, we selected all the files

106 titled as hybrid walking motions, containing 5 C3D files in total.

107 Data Processing and Feature Engineering

Data processing is an important stage to prepare data for machine learning models. In addition, data processing gives us a chance to gain insight into the data and perform feature engineering. Using C3D python library (5), we were able to extract x, y and z coordinates for each frame of the motion. We followed the following data processing steps :

112 1) Identify all missing values and convert them to NaN

An initial investigation of the dataset showed that all the missing values captured by sensors have been recorded as 0. As column z has only positive data, while both x and y can have negative, positive and 0 values, we used z as reference to identify missing values and convert all the corresponding values from all axis to *NaN* if z=0.

- 117 2) <u>Removal of missing values.</u>
- 118 Removal of missing values includes:
- a) removal of *sensors* which failed to capture locations with 50% or more
- 120 of the total frames. After applying the filtering criteria, we were left with 19 labels
- among the initial 25 labels in total.
- b) removal of *frames* with consecutive missingness >3 as too many
  consecutive missingness will result in bad interpretation during data processing.

## 124 <u>3) Interpolation</u>

Polynomial interpolation was later carried out with sensors that contain missing values. We chose polynomial interpolation with 3-degree as it is simple and is still able to approximate complicated curves. As showed in the equation below, we refer to P(x) as the polynomial function with degree of 3.

$$L_{n,j}(x) = \prod_{k \neq j} \frac{x - x_k}{x_j - x_k}$$
$$P(x) = \sum_{j=0}^{n=3} y_j L_{n,j}(x)$$

# 129 <u>3) Feature Engineering</u>

130	Featur	e engineering was conducted on the sensor data based on kinetic understanding.
131	The se	nsors are attached to human body, each part of human body will move according to
132	a spec	ific trajectory, thus the spatial change of each sensor can be a good indicator of
133	where	the sensor is mounted on the human body. Based on this assumption, we included
134	the fol	lowing features in our training set:
135	1)	Absolute location: $x$ , $y$ and $z$ values directly generated from the device.
136		Here, t refers to the time point of measurement and $x_t$ , $y_t$ and $z_t$ refer to
137		the absolute position of respective axis at time <i>t</i> .
138	2)	Relative location: the relative rank of each sensor at time point $t$ . It is
139		calculated through ranking each point $x$ , $y$ and $z$ values among all the $x$ , $y$
140		and $z$ at each time $t$ accordingly. Let's take $x$ for example.
141		$x_{tR} = Rank(x_t)$
142 143	3)	Relative change (1-dimensional): the change to the current frame from
144		previous <i>i</i> frame(s).
145		$oldsymbol{\chi}_{t,Ri} = oldsymbol{\chi}_t - oldsymbol{\chi}_{t-i}$
146		
147	4)	Relative change (3-dimenisonal): the change from previous frame to the
148		current frame.
		$D_{t} = \sqrt[2]{(x_{t_{A}} - x_{t-1_{A}})^{2} + (y_{t_{A}} - y_{t-1_{A}})^{2} + (z_{t_{A}} - z_{t-1_{A}})^{2}}$

In the classification models, the raw data as absolute location (feature 1) plus time
is used in our MLP base and LSTM models, while raw data and enhanced features
1-4 are used in MLP and XGBoost classification.

152 After feature engineering, the first 4 files are connected together as training153 dataset and the file 5 is left as testing dataset.

- 154 Modelling Algorithms
- 155 Our goal of the experiment was to accurately classify the sensor data into one of
- the 19 sensor labels in our dataset. We assumed the point\_label column as the
- 157 target variable for this classification task. We selected multilayer perceptron
- 158 (MLP) base model using scikit-learn, XGBoost and Long Short-Term Memory
- 159 (LSTM) networks for this work.
- 160
- 161 Baseline Model with Multi-Layer Perceptron

162 We decided to use Multilayer Perceptron (MLP) as our base model (Figure 1) as they are

163 good for both classification and regression problem and can work very well with tabular164 dataset.

165

Figure 1: Model topology of MLP baseline. In the basic model, we used raw kinematic
data including time, frame, x, y and z as model input, with one hidden layer and output of
23 values corresponding to the 23 unique sensor labels.

169

170 MLPs are universal function approximators as shown by Cybenko's theorem(7),

and it is a classical neural network where we have input layer, with one or more

- 172 hidden layer and an output layer.
- 173
- 174 =  $\varphi(\Sigma_{=1} + ) = (+)$
- 175

176 Here, *w* denotes the vector of weights, *x* is the vector of inputs, *b* is the bias and  $\varphi$ 177 refers to the non-linear activation function.

178 Training MLP involves multiple passes on the dataset while at the same time 179 adjusting weights and biases in relation to the error with the goal of reducing the 180 error. MLP model adjusts the weights and biases using a technique called 181 backpropagation. During forward pass, our input vector passes through the input 182 layer and activation function. The result is then compared with the ground truth 183 value to calculate the error using a loss function. In the backward pass, we 184 compute the gradients using stochastic gradient descent algorithm and adjust the 185 weights and biases. Weights and biases are adjusted in order to reduce the error when making classification. 186

### 187 <u>Enhanced MLP</u>

188 On top of all the absolute features that we included in MLP baseline model, we

189 further included enhanced features as mentioned in 2.2 data processing section.

As evident from the diagram (Figure 2), additional features are fed into the model as inputs. Features such as frame and time were removed for this model while new features such as changes of position in x, y and z from previous time point to current time point were added. Training the model using feature engineered data helped improve the model.

194

Figure 2: Model topology of MLP enhanced. In this enhanced model, we included
features as introduced in method section while retaining model structure as baseline
model.

198

199

200

### 201 <u>XGBoost</u>

202 XGBoost was initially proposed by Chen and Guestrin in 2016(8) and it 203 implements machine learning algorithms under the Gradient Boosting 204 framework(9). Specifically, XGBoost is a decision-tree-based ensemble machine 205 learning algorithm based on optimized distributed gradient boosting library and is 206 thus highly efficient, flexible and portable. We followed the default 207 hyperparameters, except for *objective* being changed to *multi:softmax* and the 208 number class being changed to the corresponding 19 classes in the raw C3D 209 dataset.

210 LSTM

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture(10) with a LSTM unit composed of a cell, an input gate, an output gate and a forget gate. For the model topology (Table-2), a sequential model which is a linear stack of layers is used. the first layer is an LSTM layer with memory units and it returns sequence. A dropout layer is applied to avoid overfitting of the model; after that, a dense layer with activation algorithm of *relu* is added, followed by a dense layer with *softmax* function for classification.

218 For the hyperparameters, we set target to maximize *categorical\_crossentropy*,

219 with 80 epochs and batch size of 100.

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 50)	10800
dropout_2 (Dropout)	(None, 50)	0

	(	0328	
dense_4 (Dense)	(None, 19)	2451	
Trainable params: 19 Non-trainable param	9,779 s: 0		

238 <u>F1 score</u>

239 We are dealing with multi-class classification for a set of time-series data points and the

240 purpose is to classify each set of data into one of the classes. The total dataset is randomly

split into training and testing dataset with ratio of 0.8, 0.2. Evaluation is carried out on the

testing dataset. As the data labels are not uniformly distributed, we chose the F1 score as

the harmonic mean of the precision and recall.

$$F1 \ score = 2 * \frac{precision * recall}{precision + recall}$$

$$accuracy = \frac{1}{N} \sum_{k=1}^{|G|} \sum_{x:g(x,y,z)} I(g(x,y,z) = \widehat{g})$$

Where *precision* (also called positive predictive value) is the fraction of true positive samples among the predicted true samples, while *recall* (also known as sensitivity) is the fraction of true positive samples among all the positive samples.

## 248 <u>Confusion matrix</u>

A confusion matrix is also known as an error matrix. It is a table layout that allows visualization of the performance of a supervised learning model. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa.

253		R	esults			
254	Both training dataset and testing dataset were applied with the same data cleaning					
255	strategies with polynomial interpolation and all the feature engineering as described in the					
256	methodology. After that, we built four models with MLP baseline, MLP enhanced model,					
257	XGBoost enhanced model and LSTM using training dataset. The performance was					
258	measured on the testing dataset.					
259	Model Accuracy					
260	The Table 3 below showed performance (F1 score) of the four models with the same					
261	testing dataset. As shown in the table, XGBoost showed highest performance with fl					
262	score at 0.94, follow	ved by MLP enhanced	d (F1 score=0.86), LSTM(F1 sc	ore=0.65) and		
263	MLP baseline (F1 sc	core=0.64).				
264	Table 3: Performance comparison among 4 models					
	Model	Algorithm	Feature	F1 score		
	MLP baseline	MLP	absolute location, frame	0.64		
	MLP enhanced	MLP	absolute location, relative location, changes	0.86		

XGboost enhanced XGBoost absolute location, relative location, 0.94 LSTM LSTM absolute location 0.66 Note: This table captures the results of the various algorithms and models used in the evaluation with

Note: This table captures the results of the various algorithms and models used in the evaluation with
 their corresponding F1 scores.

267 Confusion matrix to show agreement between truth and prediction from the model

As XGboost showed highest performance among all the 4 models, we chose predictions from XGboost and use a confusion matrix to demonstrate the agreement between true labels and model output (Figure 3). As illustrated by the side bar, lighter colors refer to higher number while darker colors indicate lower number. In the confusion matrix, we can observe that the matrix diagonal are in lighter color, suggesting high agreement between true labels and predicted labels. Nevertheless, Mislabelling was observed, especially more frequent between  $R_HEEL$  and  $R_MT5$ .

275 Figure 3: Confusion matrix of true labels and predicted labels from model w	with XGBoost
---------------------------------------------------------------------------------	--------------

276

# 277 <u>Feature Importance</u>

As XGboost showed highest accuracy among all the model, we further examined the model by checking feature importance. As we can see in figure 4, it showed that relative location of y together with z has the highest importance in the modelling, followed by absolute position of z.

- 282 *Figure 4: Feature Importance from model with XGBoost.*
- 283
- 284

## Discussion

Capturing human kinematics data requires mounting sensors to pre-defined positions on different parts of the subject's body, which may lead to missing labels. In current motion capture pipelines, manual validation and labelling is a time consuming and labour intensive postprocessing step and may become a bottleneck for downstream analysis.

289 Prior to this, Holzreiter(11) used neural network to estimate the positions of sorted 290 markers from a shuffled set through pairing up the marker locations with the shuffled set 291 using the nearest neighbour search. Meyer *et al.*(12) estimated the skeletal configuration 292 by least-squares optimization and exploited the skeletal model to automatically label the 293 markers. Besides, Han et al.(13) proposed auto-labelling approaches specifically for 294 labelling hands' kinematics data through keypoint regression problem solved with 295 convolutional neural networks. In addition, Saeed et al.(14) used a data-driven approach 296 for automatic labelling through permutation estimation where shuffled markers are ranked 297 based on a pre-defined order.

While the previous work has no labelled data and tried to classify the kinematic data to different locations according to the temporal profile, our work targets to label kinematic data with missing labels. Specifically, the problem we are focusing on is the data quality of the captured data with majority of the data being labelled and a few missing data points due to technical errors. After effective feature engineering and modeling with 3 selected model algorithms, we are able to achieve decent classification outome with F1-score over 90%.

305 The list of features, model performance and feature importance will vary based on the 306 type of motion being captured by the sensors. In this experiment, we used the motion data 307 of a walking subject, on which over 20 sensors were mounted. Since the motion was 308 horizontal, all sensors moved by similar distance along the x axis as the subject moved. 309 As a result, the relative position of x did not contribute as much to differentiation of 310 sensors, as compared to y and z axes, because the sensors were mounted at different 311 hights on the subject's body. To apply the models to other movements, we need to retrain 312 the model with least effort since the list of features, model performance and feature 313 importance will vary based on the type of motions being captured by the sensors. We also 314 noticed that appropriate feature engineering was essential to get higher accuracy. By 315 using relative locations of the sensors along the x, y and z axes, we were able to improve 316 the performance of the model by over 31% compared to the baseline model that used just 317 the absolute locations of the sensors. This shows that relative location was the key factor 318 in determining the appropriate sensors.

Conclusion

319

320 We tested different data transformation and machine learning algorithms to develop a 321 multi-class classification model that can label kinematics data with high accuracy. It was 322 also important to apply techniques on kinematics timeseries data as it helped us to 323 improve the model. Calculating the 1-dimensional and 3-dimensional relative change to 324 the frames helped with creation of new features. These techniques improved the model 325 performance considerably. XGBoost model gave the best performance compared to the 326 neural architecture models. The models can be used to accurately label the three-327 dimensional motion data which can provide insights into movements of a patient with 328 injury or a patient with disability. Analyzing these movements can further help in either 329 creating a recovery plan or an exoskeleton that can aid in recovery.

330

## 331 **Reference**

- 1. Hatze H. The meaning of the term biomechanics. J Biomech. 1974;7(12):189–190.
- An K, EY C. Kinematic analysis of human movement. Ann Biomed Eng.
   1984;12(6):585-97. doi: 10.1007/BF02371451. PMID: 6534225. Ann Biomed Eng.
   1984;12(6):585-97.
- Bahl JS, Nelson MJ, Taylor M, Solomon LB, Arnold JB, Thewlis D.
   Biomechanical changes and recovery of gait function after total hip arthroplasty for
   osteoarthritis: a systematic review and meta-analysis. Osteoarthr Cartil.
   2018;26(7):847–63.
- 340 4. CMU Graphics Lab Motion Capture Database [Internet]. Available from:
  341 http://mocap.cs.cmu.edu/faqs.php
- 342 5. Lab TEC. c3d 0.3.0 [Internet]. Python Library. Available from:

- 343 https://pypi.org/project/c3d/
- 344 6. C3D.ORG. No Title. p. https://www.c3d.org/sampledata.html.
- 345 7. Cybenko G. Approximation by superpositions of a sigmoidal function. Math
  346 Control Signals, Syst. 1989;2(4):303–314.
- 347 8. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proceedings of
- 348 the ACM SIGKDD International Conference on Knowledge Discovery and Data349 Mining. 2016.
- 350 9. Friedman JH. Greedy Function Approximation: A Gradient Boosting Machine.351 1999;
- 352 10. Schmidhuber SHJ. Long short-term memory. Neural Comput. 1997;9(8):1735–
  353 1780.
- 354 11. Holzreiter S. Autolabeling 3D tracks using neural networks. Clin Biomech.
  355 2005;20(1):1–8.
- Meyer J, Kuderer M, Müller J, Burgard W. Online marker labeling for fully
  automatic skeleton tracking in optical motion capture. 2014 IEEE Int Conf Robot
  Autom. 2014;
- HAN S, LIU B, WANG R, Ye Y, TWIGG CD, KIN K. Online Optical Markerbased Hand Tracking with Deep Labels. ACM Trans Graph. 2018;37(4).
- 361 14. Ghorbani S, Etemad A, Troje N. Auto-labelling of Markers in Optical Motion
  362 Capture by Permutation Learning. Comput Vis Pattern Recognit. 2019;167–78.
- 363







