

# An Unsupervised Model for Physical Rehabilitation Exercise Assessment

Thasina Tabashum, Farhad Mokter, Md Marufi Rahman

| Participants     | Sections   | Responsibilities   |
|------------------|--|--|
| Thasina Tabashum | Feature Engineering,<br>Performance metrics,<br>Scoring Function | Understanding existing study in details, implementing the feature engineering part and responsible for the report and presentation |
| Md Marufi Rahman | NN models, model evaluation,<br>Model tuning                     | NN Model Architectures, NN Model Tuning, Model Evaluation, and documentation of that part of the project and presentation          |
| Farhad Mokter    | RNN model  | Design and Evaluate the RNN Model along with Analysing the Scores.   |

## 1. Abstract:

With the improvement of sensor systems , computer aided assessment of physical exercises can be associated with manual evaluation. Deep Learning models can be trained with proper physical movement data and therefore, can be evaluated to distinguish the appropriate movement of a patient during his training/practice sessions. This study goal is to build models to assess ten distinct exercises.

## 2. DL Problem specification:

### 2.1 Dataset Description:

The data set consists of 10 rehabilitation movements. A sample of 10 healthy individuals repeated each movement 10 times in front of two sensory systems for motion capturing: a Vicon

optical tracker. The data is presented as positions and angles of the body joints in the skeletal models provided by the Vicon system.

The following 10 movements were selected for the data set: (1) deep squat, (2) hurdle step, (3) inline lunge, (4) side lunge, (5) sit to stand, (6) standing active straight leg raise, (7) standing shoulder abduction, (8) standing shoulder extension, (9) standing shoulder internal-external rotation, and (10) standing shoulder scaption.



*Fig 1 : ten movements*

The Vicon folder contains two subfolders, ‘Positions’ and ‘Angles,’ which include the files with the respective values for the joint displacements.

The nomenclature of the files in the data set includes the following information: movement number\_subject number\_positions/angles. For example, the data instance ‘m04\_s06\_angles’ pertains to the 4<sup>th</sup> movement (i.e., side lunge) performed by the 6<sup>th</sup> subject, and it consists of the angular displacements of the joints expressed in degrees. Similarly, ‘m08\_s02\_positions’ corresponds to the recorded Cartesian position coordinates for the 2<sup>nd</sup> subject while performing the standing shoulder extension movement expressed in millimeters.

Further, because each movement consists of 10 episodes, or repetitions, the data is also provided in a segmented form, where each file comprises the measurements for one episode of one of the movements. The corresponding data is provided in the ‘Segmented Movements’ folder. The following file nomenclature is used for the segmented movements: movement number\_subject number\_episode number\_positions/angles. In this case, the file ‘m04\_s06\_e10\_angles’ consists of the angular joint measurements for the 10<sup>th</sup> episode of the 4<sup>th</sup> movement performed by the 6<sup>th</sup> subject.

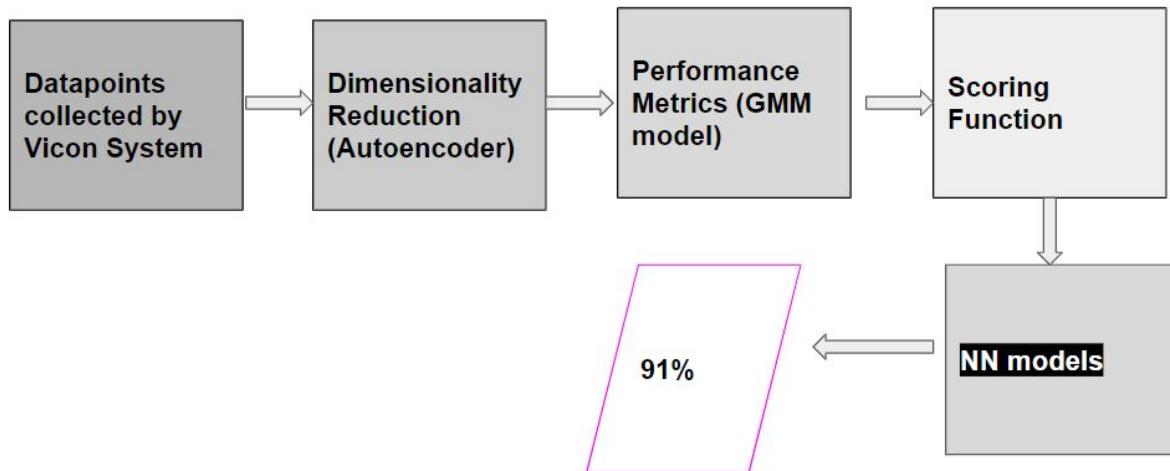
In addition, the data set provides examples of the movements performed in an incorrect manner. The rationale for it is that patients with musculoskeletal injury or constraints are assumed to be unable to, at least initially, perform the prescribed therapy movements in a correct or optimal way during a physical therapy exercise. This portion of the data can serve as a testing set, and it can be used for validation of derived mathematical models for the above described movements. In creating the set of incorrect movements, all of the subjects performed 10 episodes of the 10 movements arbitrarily in a suboptimal manner. The file nomenclature for the incorrect movements follows the same order as for the correct movements, and in addition, all files have an extension \_inc, which implies 'incorrect'. For instance, the file 'm05\_s04\_e03\_angles\_inc' corresponds to the incorrect 3<sup>rd</sup> episode performed by the 4<sup>th</sup> subject for the sit to stand movement.

Dataset Source and Description source : <https://webpages.uidaho.edu/ui-prmd/>

### 3. Design and Milestones

#### 3.1 Proposed Framework :

The proposed framework has been given below.



*Fig. 2. Overview of the proposed framework for assessment of rehabilitation exercises.*

The skeletal joint coordinates acquired by the sensory system are processed via dimensionality reduction, performance quantification, and scoring mapping to obtain movement quality scores

that are subsequently used for training an NN model. The trained NN model is afterward used to automatically generate movement quality scores for input movement data acquired by the sensory system.

On our project, we have followed the framework they proposed and tried to tune NN models to have better results. In section 4, we have demonstrated in detail the feature engineering part of the paper we followed which includes data preprocessing and dimensionality reduction. In section 5 and 6 performance metrics and scoring functions are discussed in detail. In sections 7 and 8, we have mentioned the CNN and RNN models , we have tuned.Lastly on section 8 we concluded the study.

## 4. Feature Engineering

### 4.1 Data Preprocessing:

Each exercise has performed by 10 subjects. Further, each movement consists of 10 episodes. After removing first episodes from every subject as most of the readings were missing in the first episode during the data recording. The sequence length is 117 and scaled the data between -1 and 1. The dimension of data is (90, 240 , 117) for each of the exercises. 240 is the timeframe and 100 frames are added by adding 50 in the beginning and 50 in the end. Three examples of data sequence are given on Figure 2,3,4 .

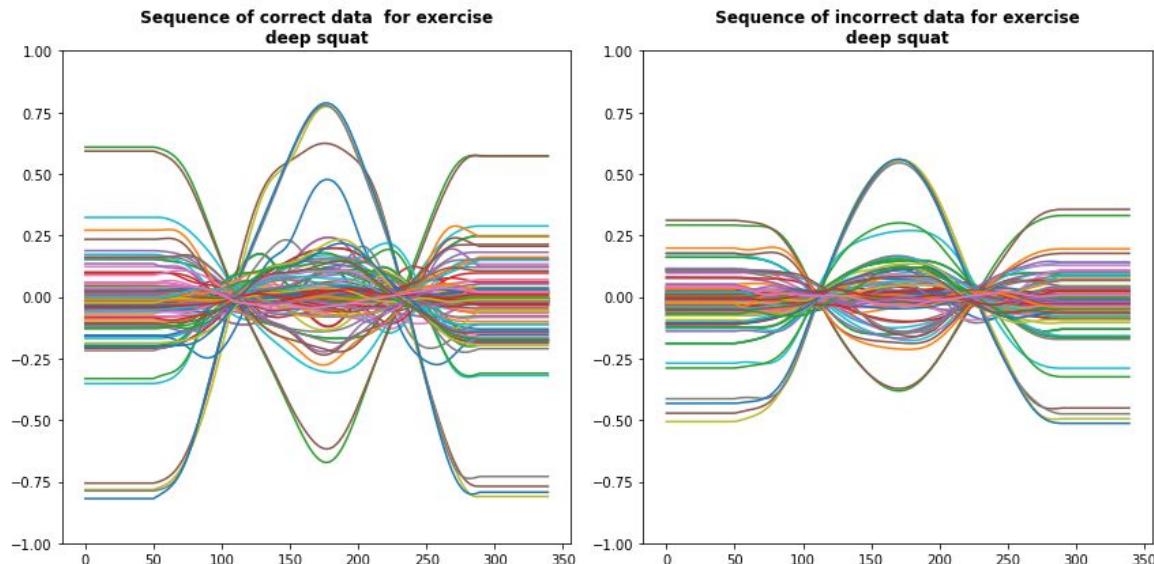
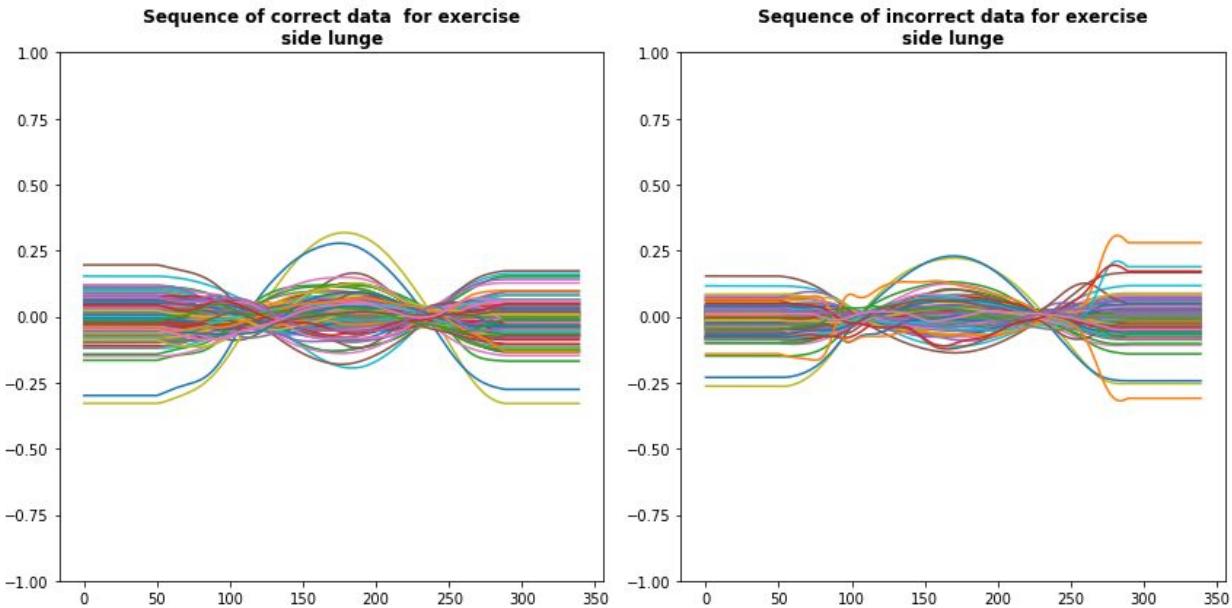
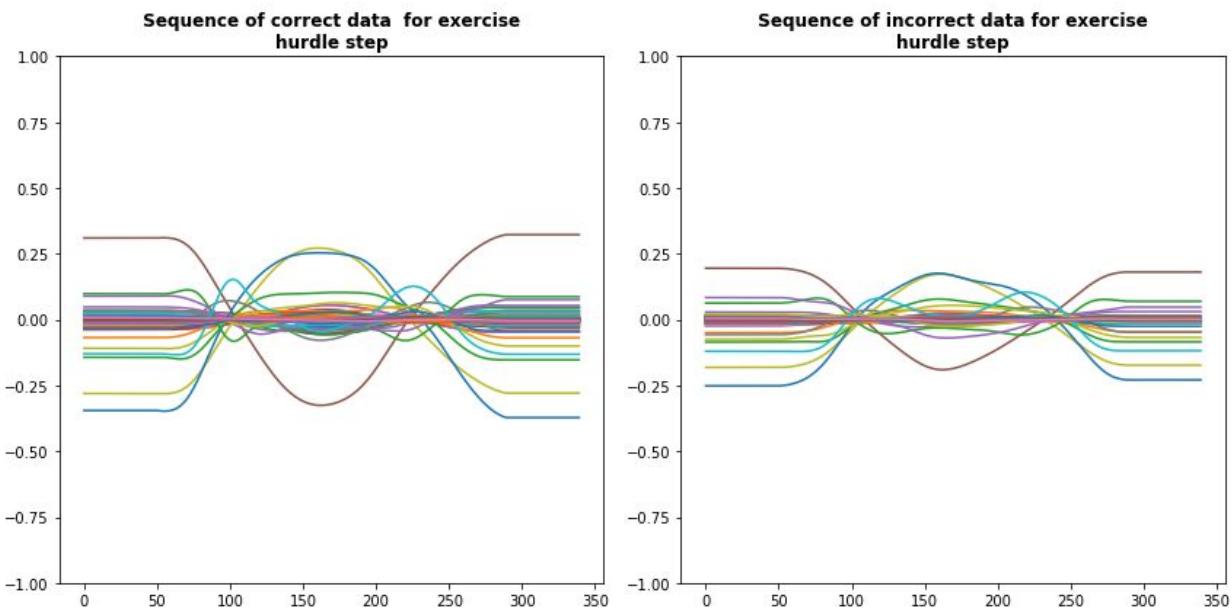


Fig 3: Correct and Incorrect Sequence of data for deep squat



*Fig 4: Correct and Incorrect Sequence of data for side lunge*



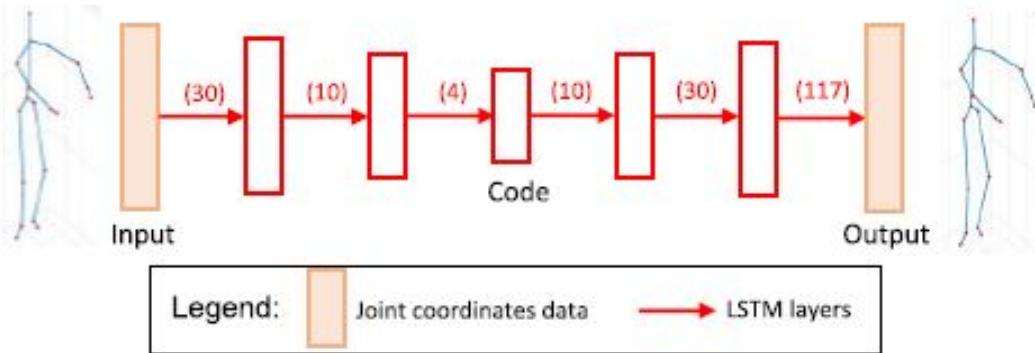
*Fig 5: Correct and Incorrect Sequence of data for hurdle step*

#### 4.2. Dimensionality Reduction

The sensory systems for motion capturing typically track between 15 and 40 skeletal joints, depending on the sensor type. The measurement data consists of 3-dimensional spatial positions and/or orientations for each joint, and therefore the dimensionality of the data ranges between 45 and 120. Dimensionality reduction of recorded data is an essential step in

processing human movements to suppress unimportant, redundant, or highly correlated dimensions.

In the proposed framework, Autoencoders are trained to find which minimize the mean squared deviation between input and output data.



*Fig 6: The proposed autoencoder architecture projects an input movement data into a code representation, and re-projects the code into the movement data.*

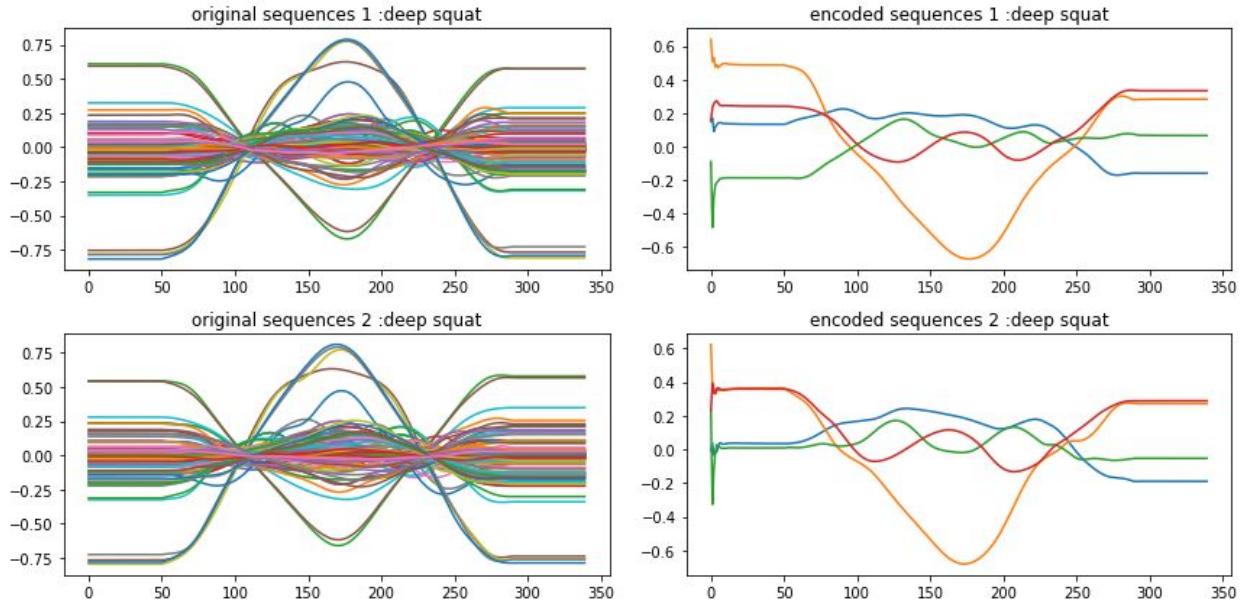
| Layer (type)             | Output Shape         | Param # |
|--------------------------|----------------------|---------|
| =====                    |                      |         |
| input_1 (InputLayer)     | [ (None, 340, 117) ] | 0       |
| lstm (LSTM)              | (None, 340, 30)      | 17760   |
| lstm_1 (LSTM)            | (None, 340, 10)      | 1640    |
| lstm_2 (LSTM)            | (None, 340, 4)       | 240     |
| lstm_3 (LSTM)            | (None, 340, 10)      | 600     |
| lstm_4 (LSTM)            | (None, 340, 30)      | 4920    |
| lstm_5 (LSTM)            | (None, 340, 117)     | 69264   |
| =====                    |                      |         |
| Total params: 94,424     |                      |         |
| Trainable params: 94,424 |                      |         |
| Non-trainable params: 0  |                      |         |

A graphical representation of the adopted architecture for the autoencoder network is presented in Fig. 5. The encoder portion consists of three intermediate layers of LSTM recurrent units with 30, 10, and 4 computational units, and the corresponding decoder portion has three intermediate layers of LSTM units.

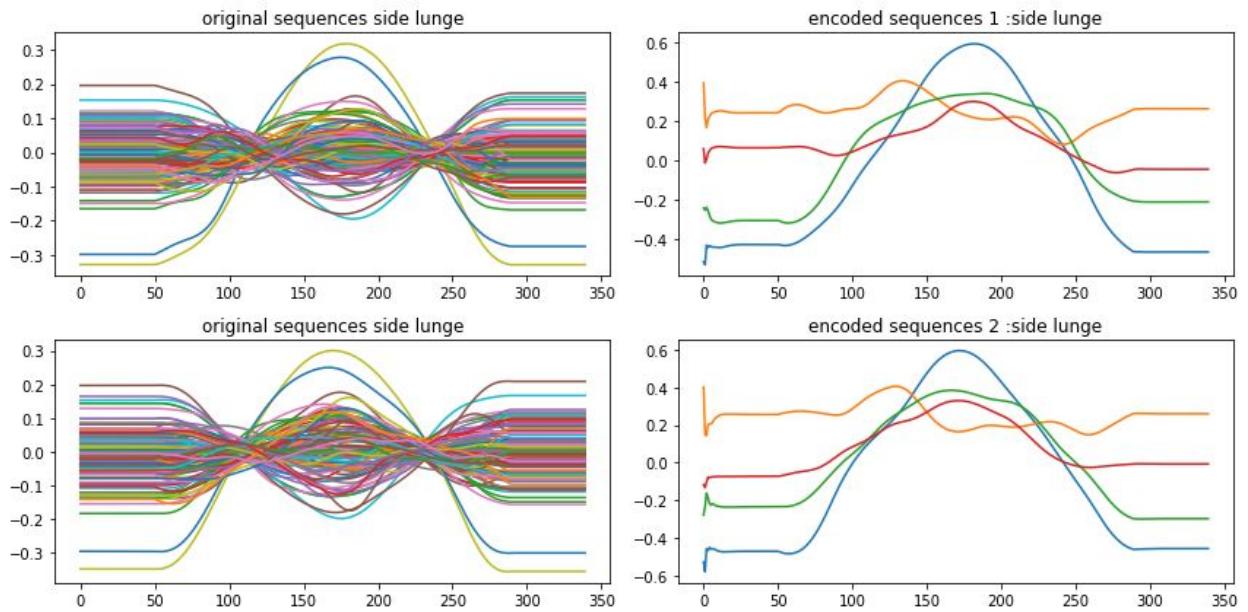
We have trained 10 autoencoder models for 10 exercises. Three exercises are (deep squat, hurdle step, side lunge) given on the section 4.2.a, 4.2.b, 4.2.c .

#### 4.2.a Encoder Output:

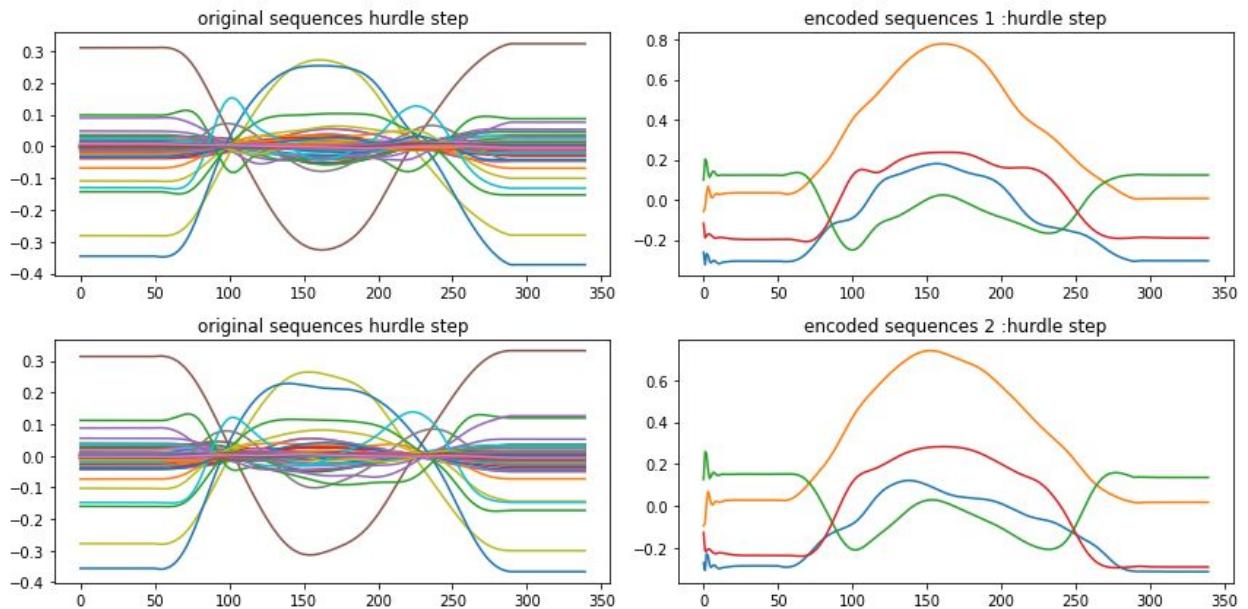
In figure 7,8,9 encoders outputs have been shown for angle data . Further figures 10,11,12 are encoders outputs of position data for the three exercises.



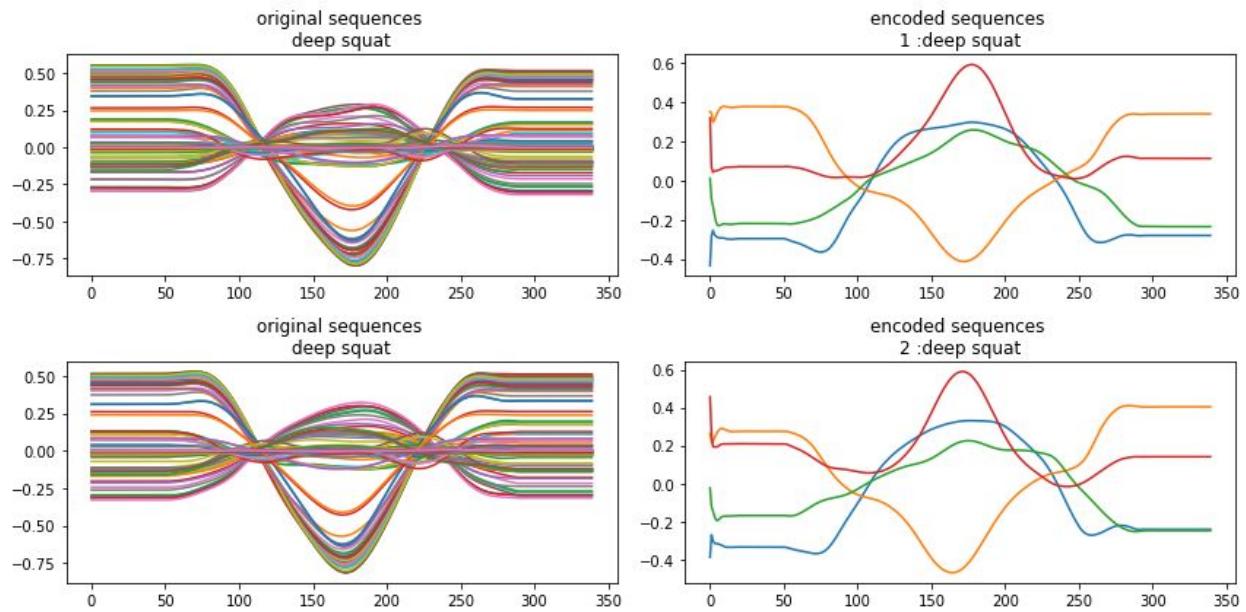
*Fig 7: Encoded sequences of deep squat angle data*



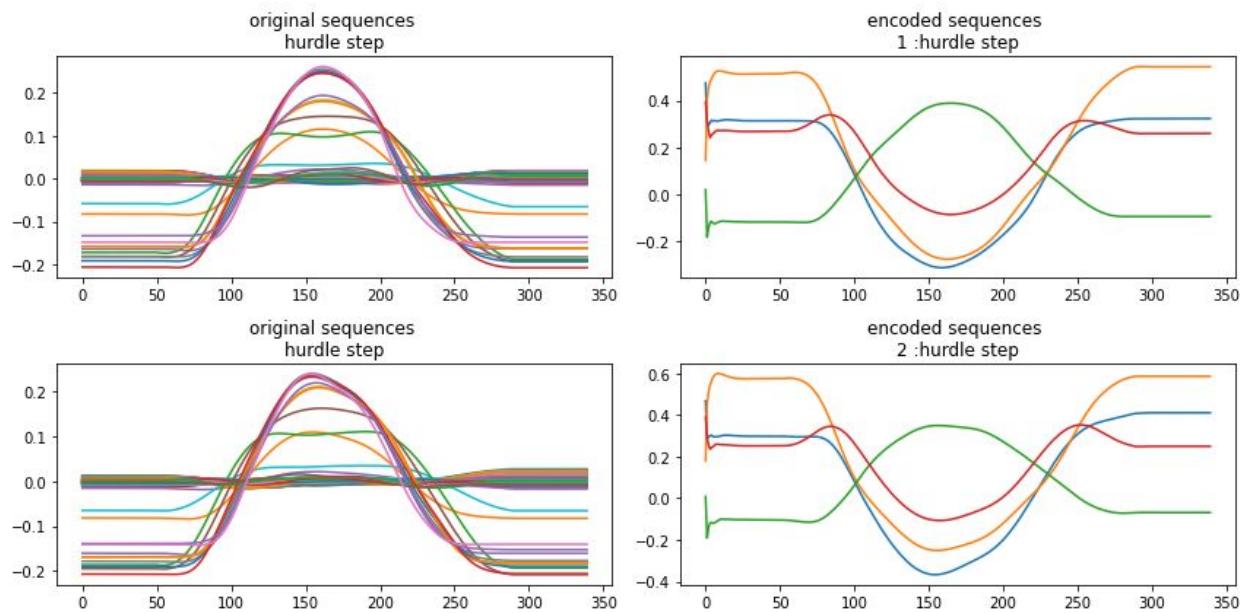
*Fig 8: Encoded sequences of side lunge angle data*



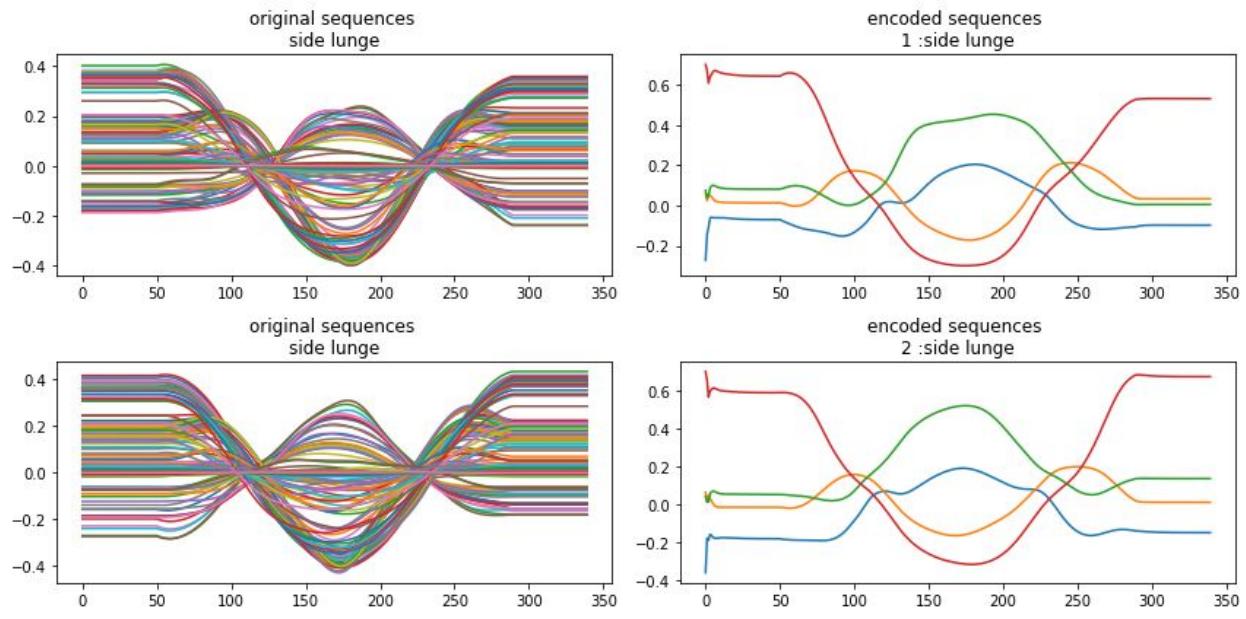
*Fig 9: Encoded sequences of hurdle step angle data*



*Fig 10: Encoded sequences of deep squat for position data*



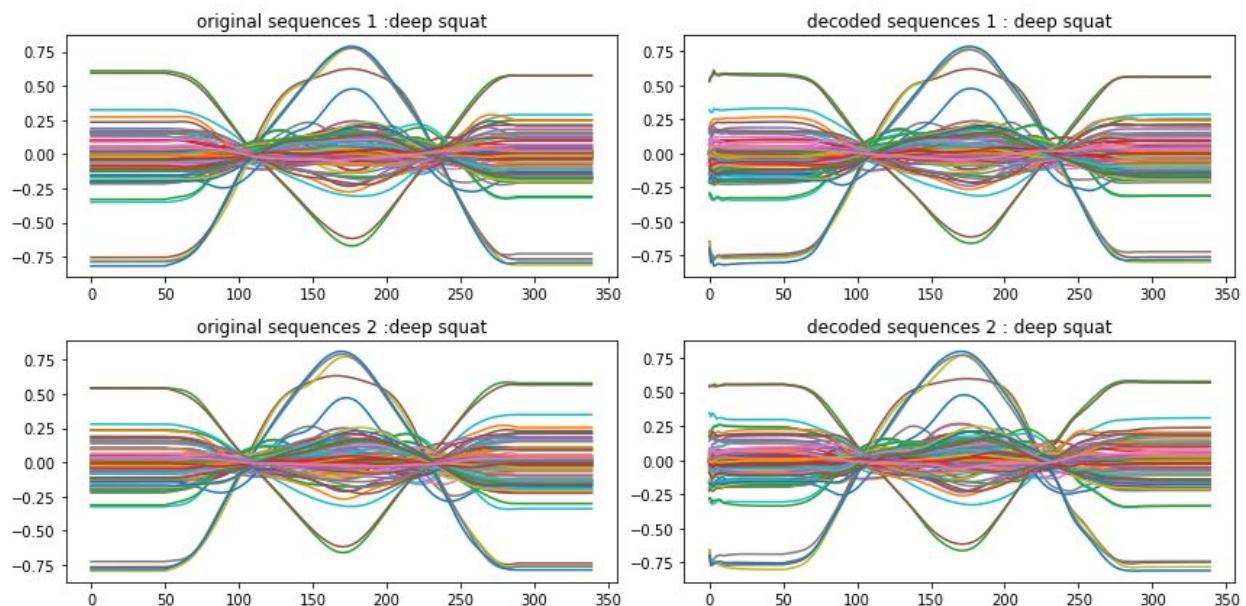
*Fig 11: Encoded sequences of hurdle step position data*



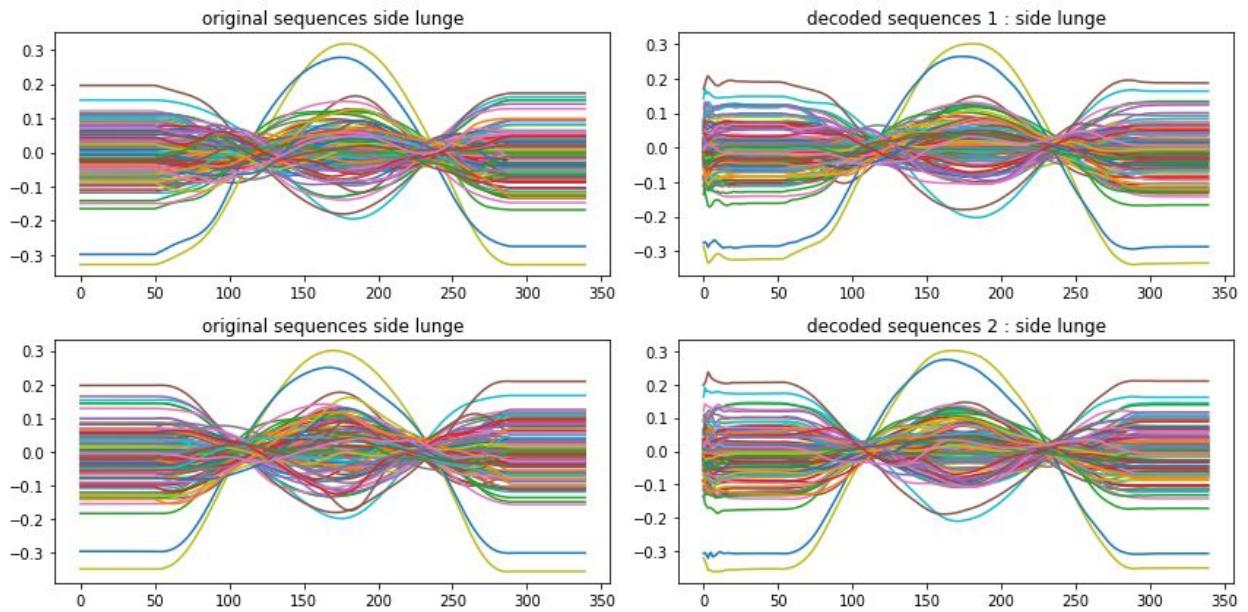
*Fig 12: Encoded sequences of side lunge for position data*

#### 4.2.b Decoder Output:

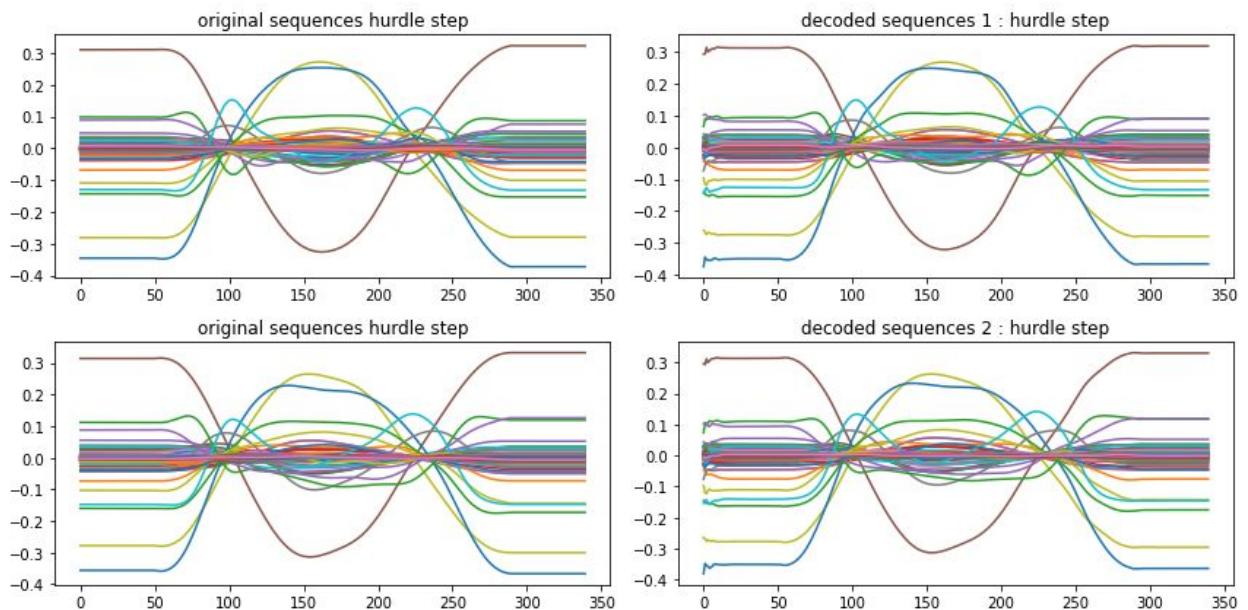
In figure 13,14,15 decoders outputs have been shown for angle data . Further figures 16,17,18 are decoder outputs of position data for the three exercises.



*Fig 13: decoded sequences of deep squat for angle data*



*Fig 14: decoded sequences of side lunge for angle data*



*Fig 15: decoded sequences of hurdle step for angle data*

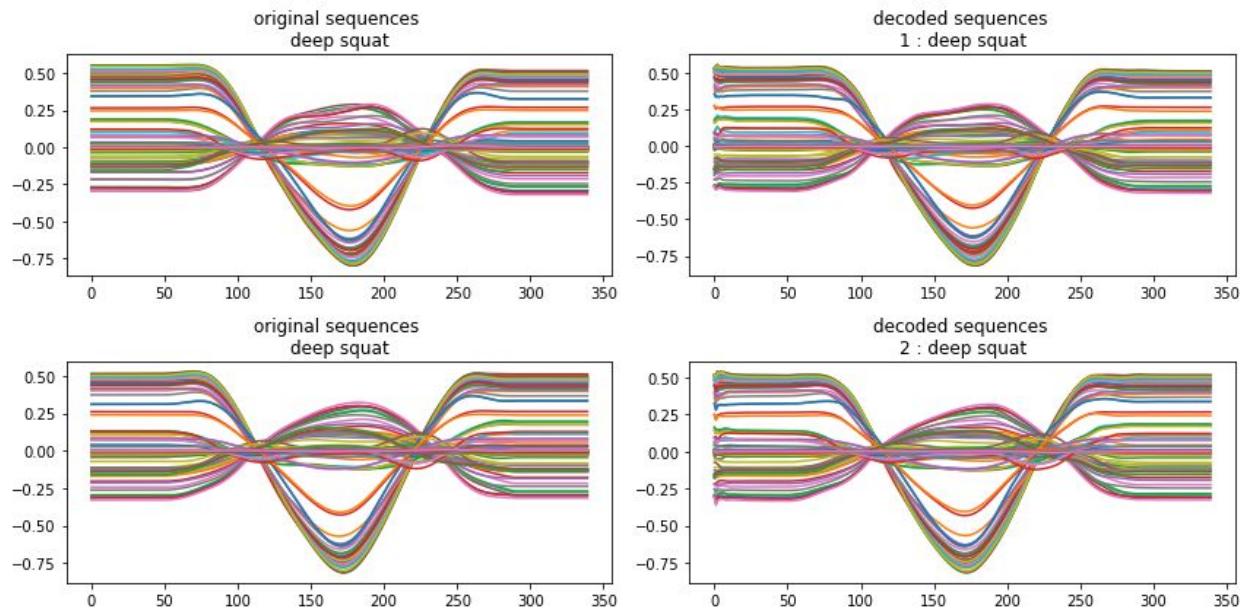


Fig 16: decoded sequences of deep squat for position data

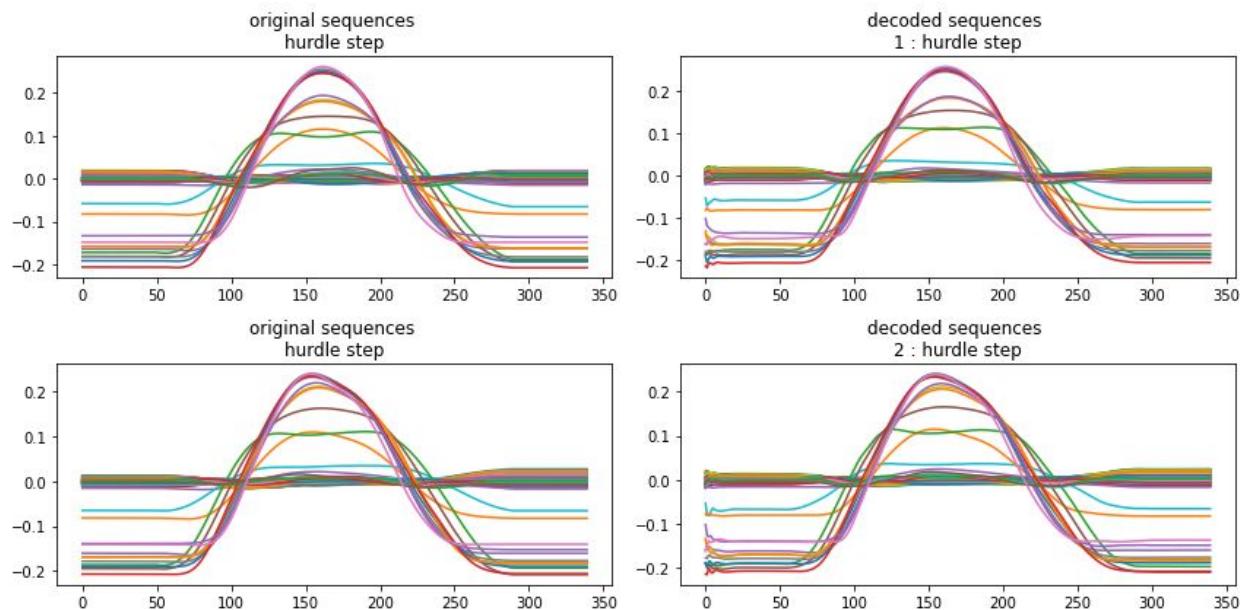
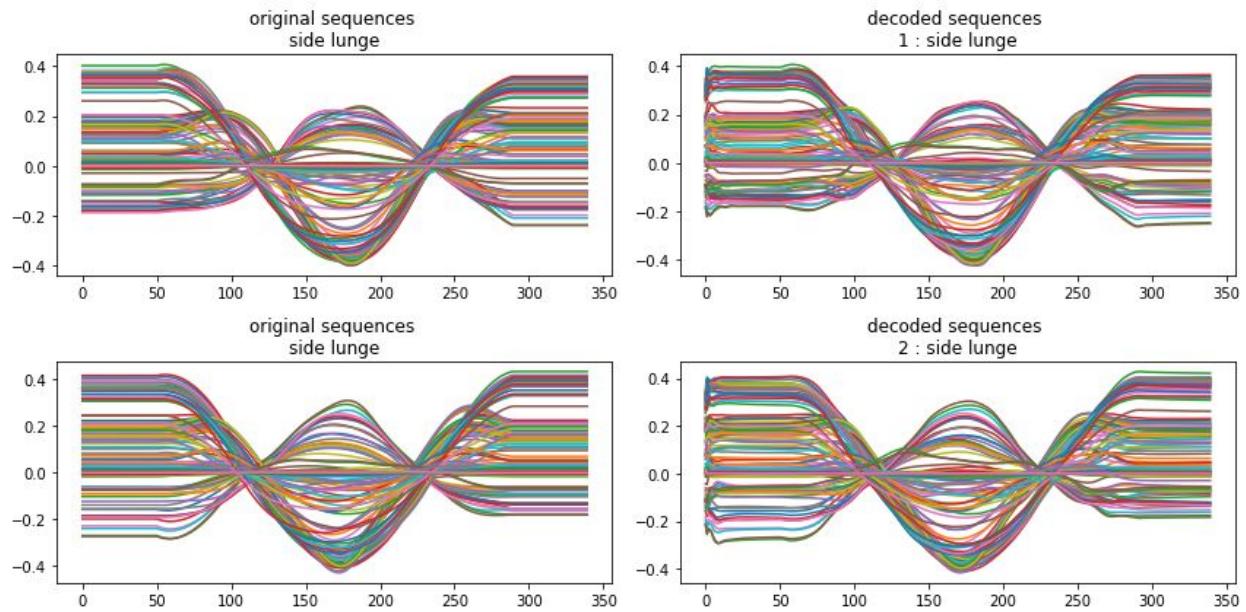


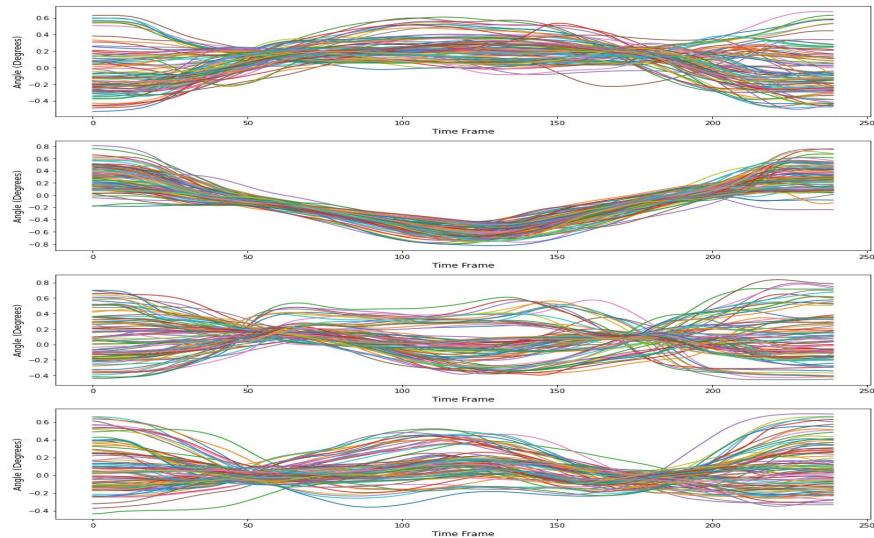
Fig 17: decoded sequences of hurdle step for position data



*Fig 18: decoded sequences of side lunge for position data*

#### 4.2.c Autoencoder Output:

In figure 19,20,21 decoders outputs have been shown for angle data . Further figures 21,22,23 are decoder outputs of position data for the three exercises.



*Fig 19 : Autoencoder output for deep squat for angle data*

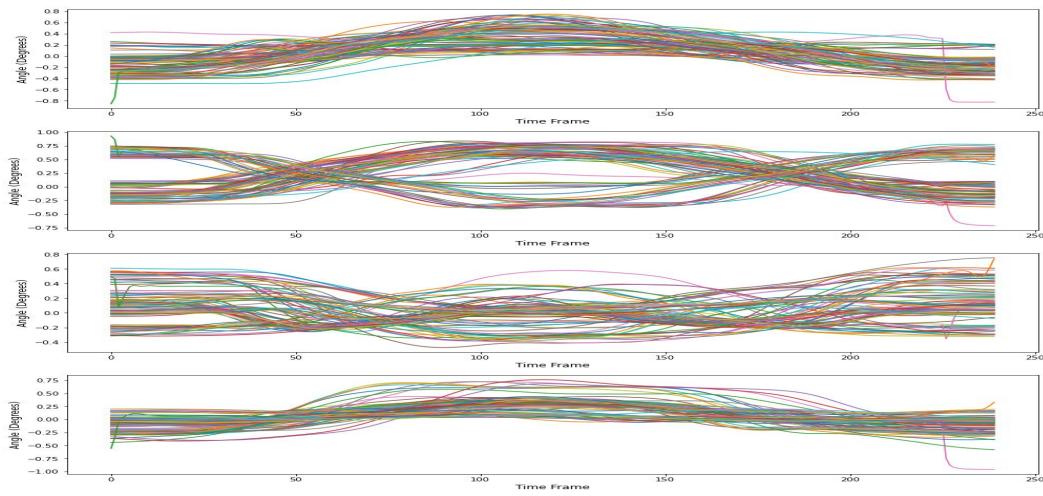


Fig 20 : Autoencoder output for hurdle step for angle data

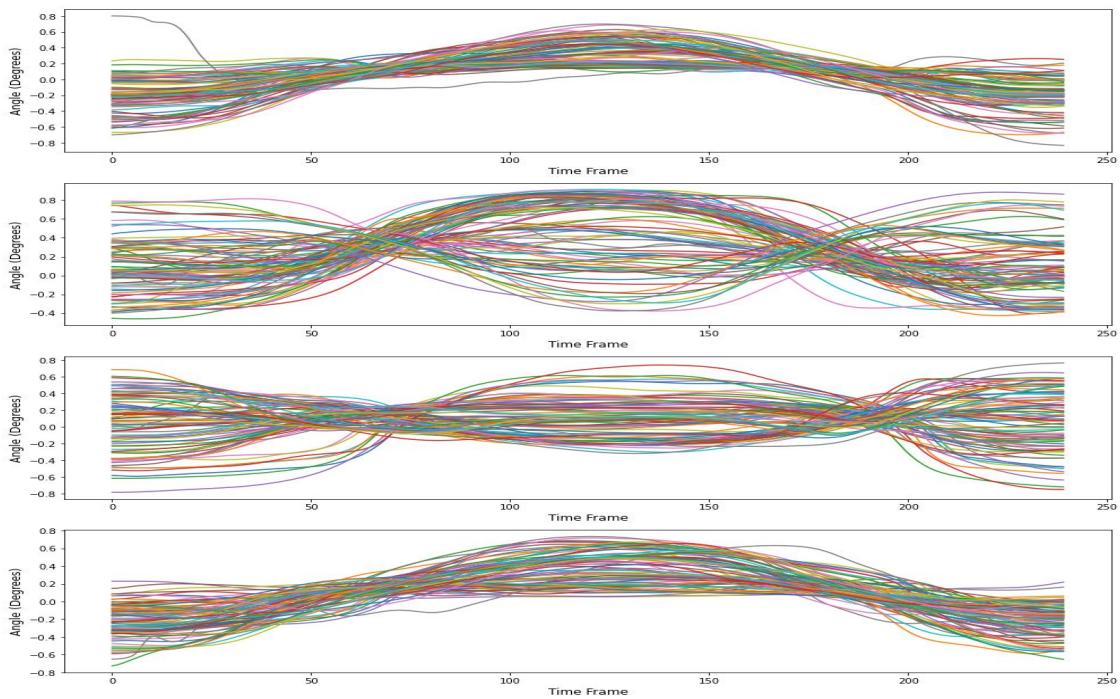
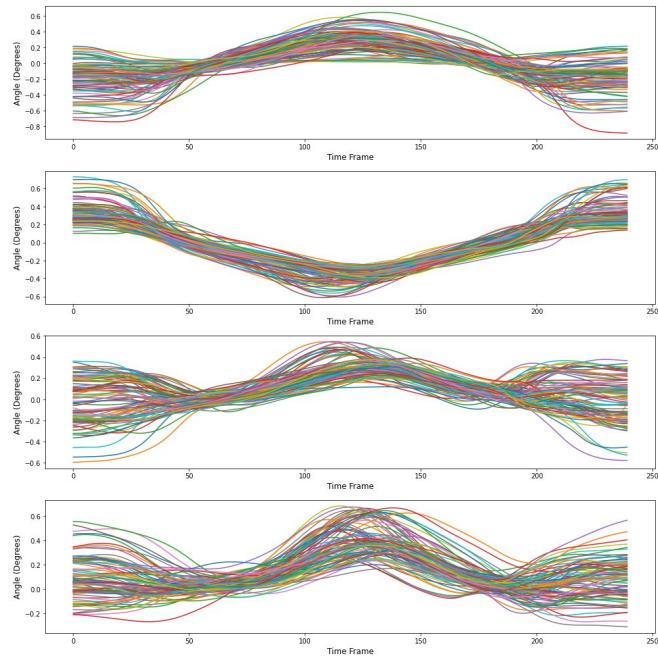
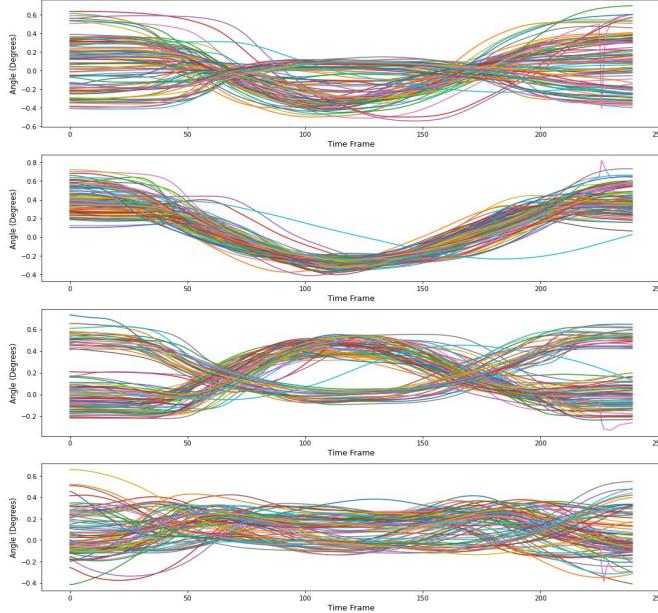


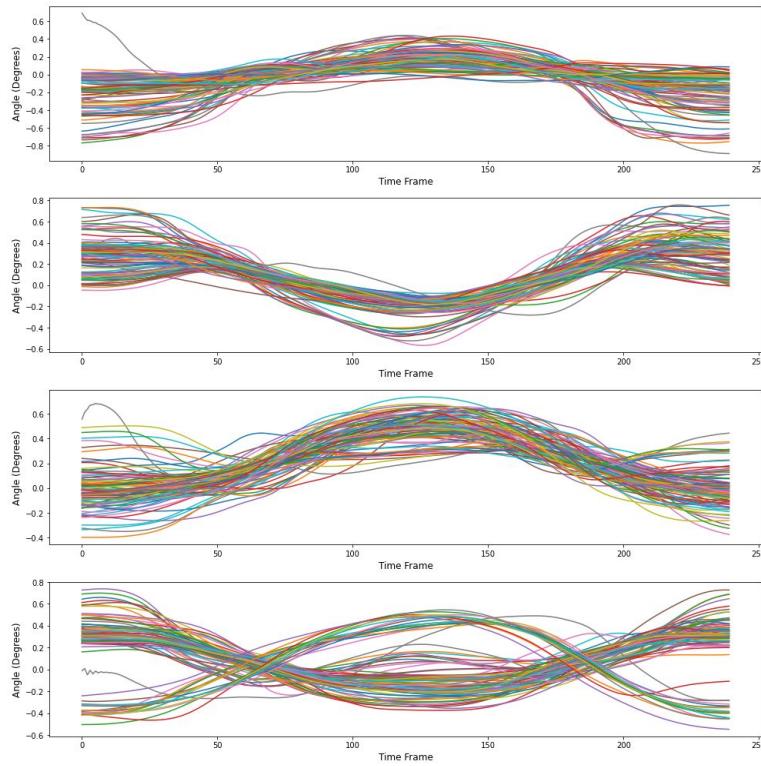
Fig 21 : Autoencoder output for side lunge for angle data



*Fig 22 : Autoencoder output for deep squat for position data*



*Fig 23 : Autoencoder output for hurdle step for position data*



*Fig 24 : Autoencoder output for side lunge for position data*

#### 4.2.d Autoencoder Models loss:

**Tabel 1 : Autoencoder model loss and val-loss for each of exercises**

| Exercises                          | Loss                   | Val-loss               |
|------------------------------------|------------------------|------------------------|
| Deep Squat                         | 0.00011004383122781292 | 0.000456993788247928   |
| Hurdle step                        | 1.8908071069745347e-05 | 5.489791146828793e-05  |
| Inline lunge                       | 6.880579803691944e-06  | 4.6327593736350536e-05 |
| Side lunge                         | 0.0001591856125742197  | 0.0008839638903737068  |
| Sit to stand                       | 0.00022150937002152205 | 0.0007559240330010653  |
| Standing active straight leg raise | 2.4286064217449166e-05 | 6.0964837757637724e-05 |
| Standing shoulder abduction        | 6504738596267998e-05   | 2.141365985153243e-05  |
| Standing shoulder extension        | 8.56148744787788e-06   | 9.572367162036244e-06  |

|  |                          |                         |
|--|--------------------------|-------------------------|
| Standing shoulder internal-external rotation | $3.16977639158722e-05$   | $6.225203833309934e-05$ |
| Standing shoulder scaption                   | $2.1394595023593865e-05$ | $0.0007893053698353469$ |

## 5. Performance Metrics:

A metric based on Gaussian mixture model (GMM) log-likelihood for the evaluation. The choice stems from the demonstrated capacity of statistical methods to encode the inherent random variability in human movements; this results in improved ability by the model-based metrics to handle spatiotemporal variations in rehabilitation data. Log-likelihood of a movement data for a given model is a natural choice for evaluation of data instances in probabilistic models.

In this study, a GMM is trained on the data with reduced dimensionality obtained by an autoencoder neural network and uses the log likelihood as a performance indicator.

GMM is a parametric probabilistic model for representing data with a mixture of Gaussian probability density functions . GMM is frequently used for modeling human movements.

Figure 25, 26 and 27 is GMM encoded output of angles for deep squat, hurdle step and side lunge. Figure 28 and 29 GMM encoded output of positions for hurdle step and side lunge.

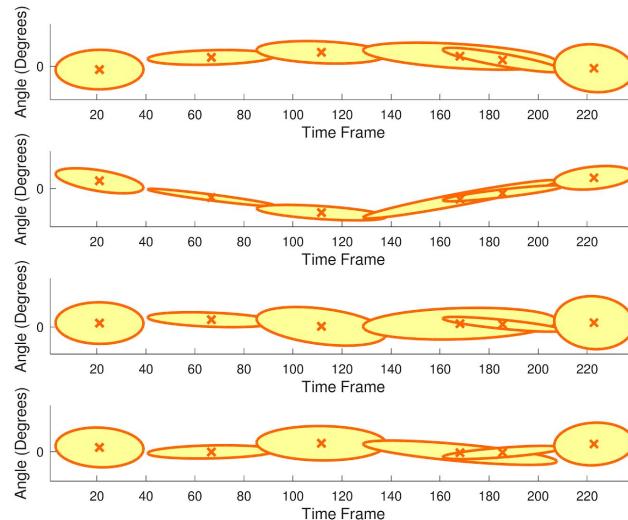
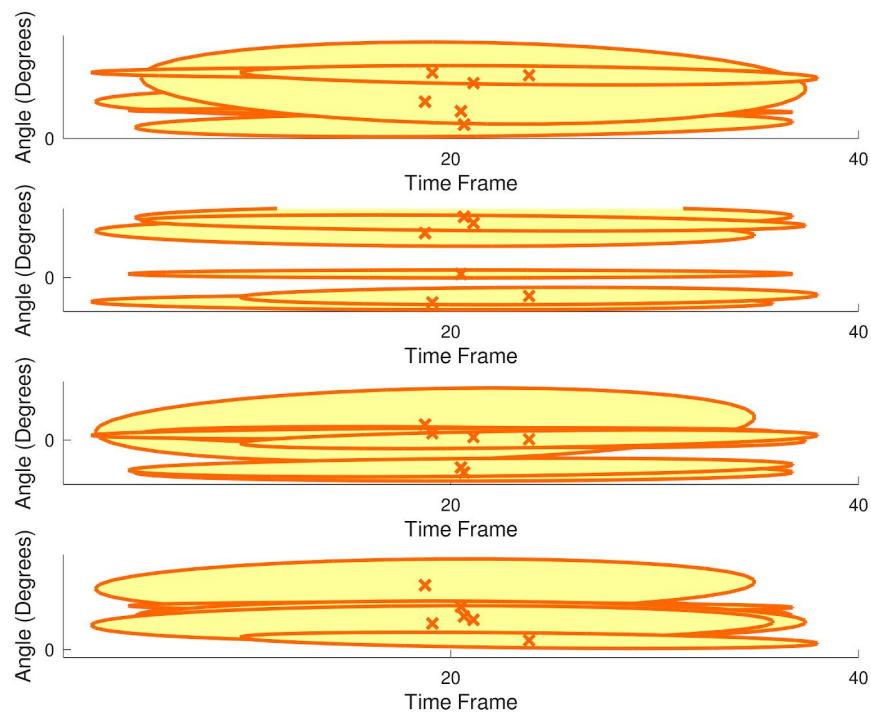
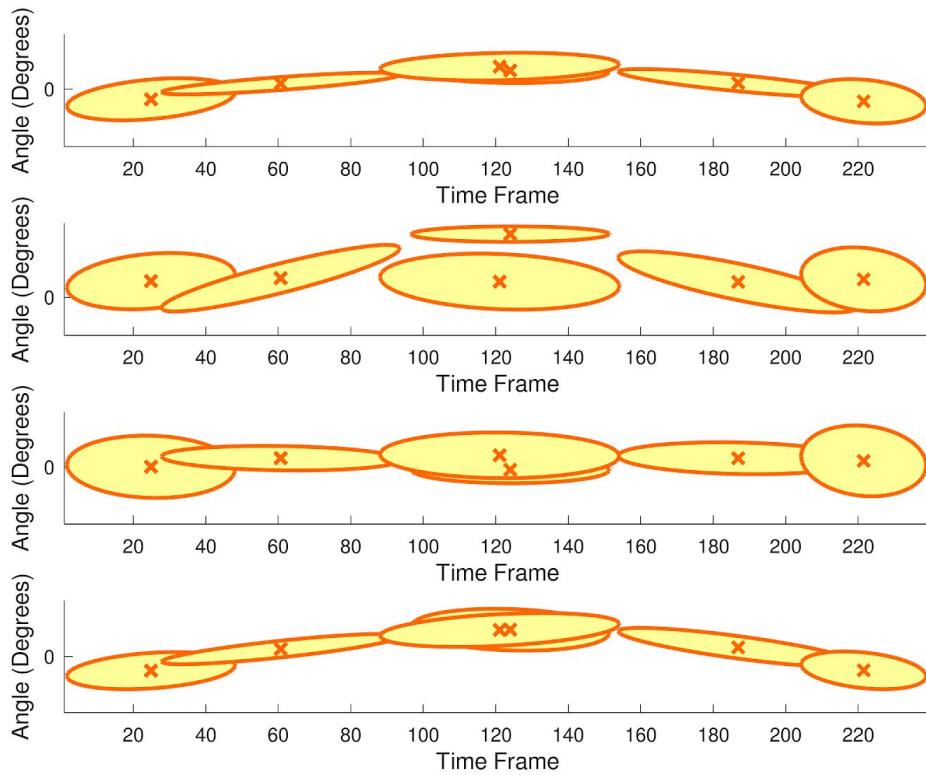


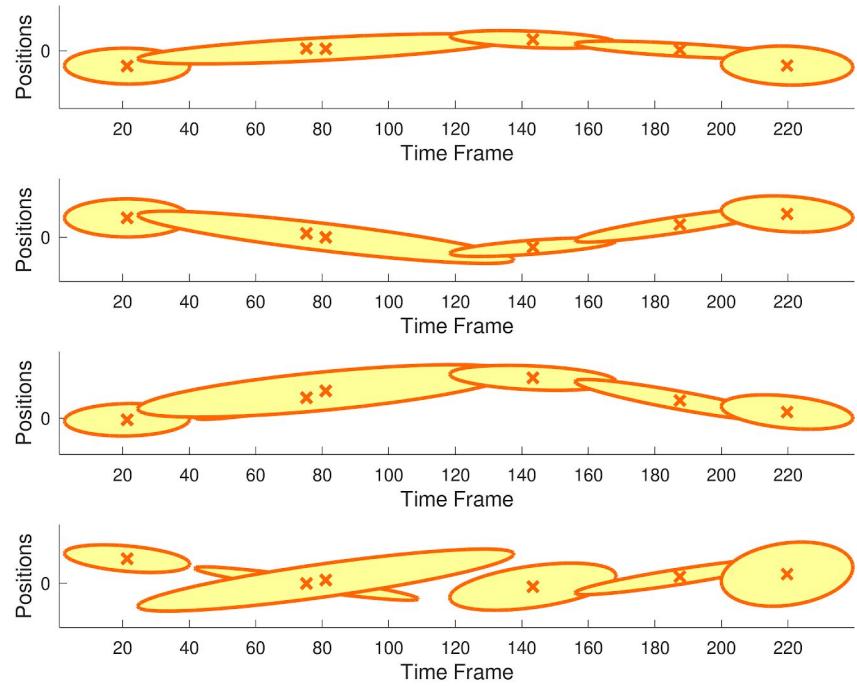
Fig 25: GMM Encoded Movements for Deep Squat of angles



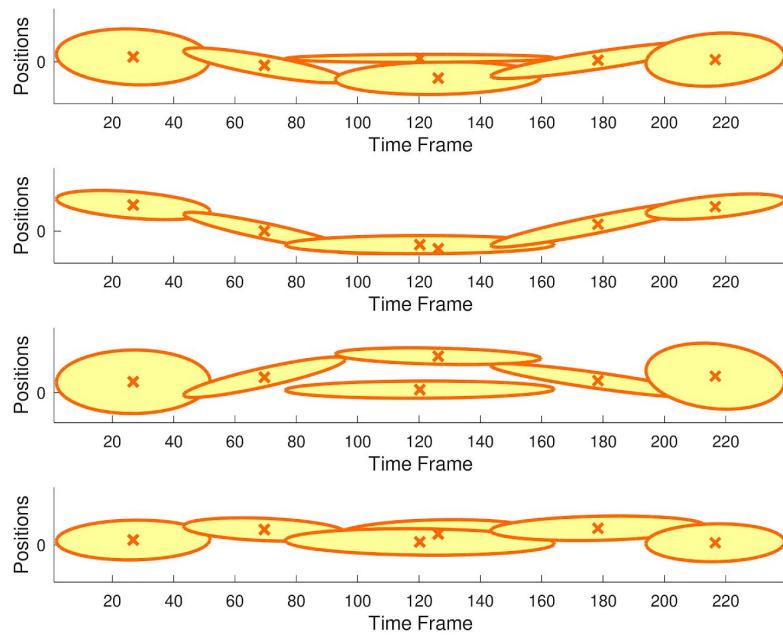
*Fig 26: GMM Encoded Movements for hurdle step of angles*



*Fig 27: GMM Encoded Movements for side lunge of angles*



*Fig 28: GMM Encoded Movements for side lunge of positions*



*Fig 29: GMM Encoded Movements for hurdle step of positions*

## 6. Scoring Function:

In the presented framework, a scoring function maps the values of the performance metrics into a movement quality score in the range between 0 and 1.

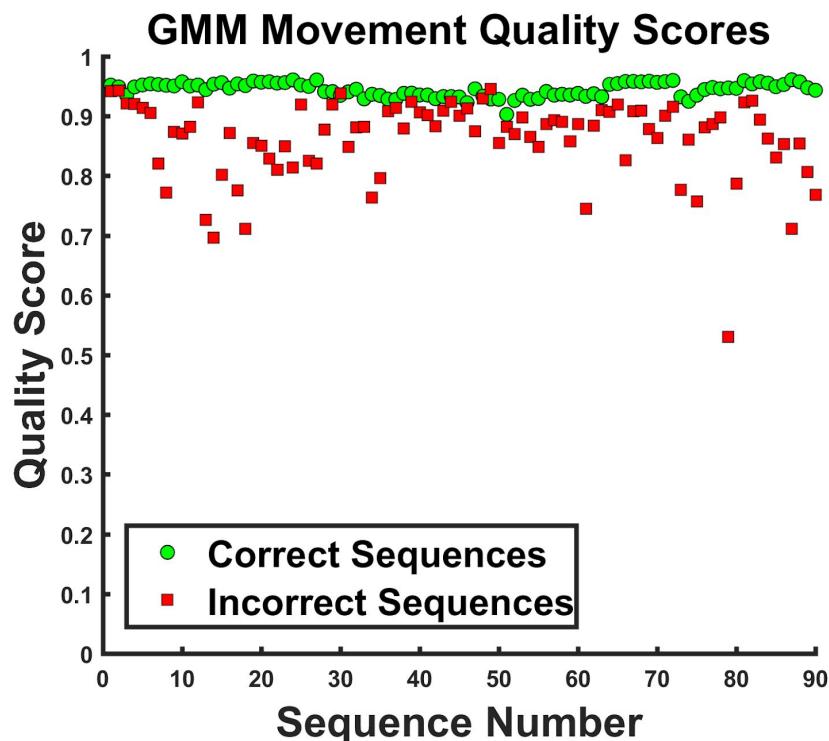
For a sequence of performance metric values of the reference movements  $x = (x_1, x_2, \dots, x_L)$  and a sequence  $y = (y_1, y_2, \dots, y_L)$  related to the patient movements, we propose the following scoring function:

$$x_k = (1 + e^{(x_k/(\mu+3\delta)-\alpha_1)} - 1) \dots \dots (1)$$

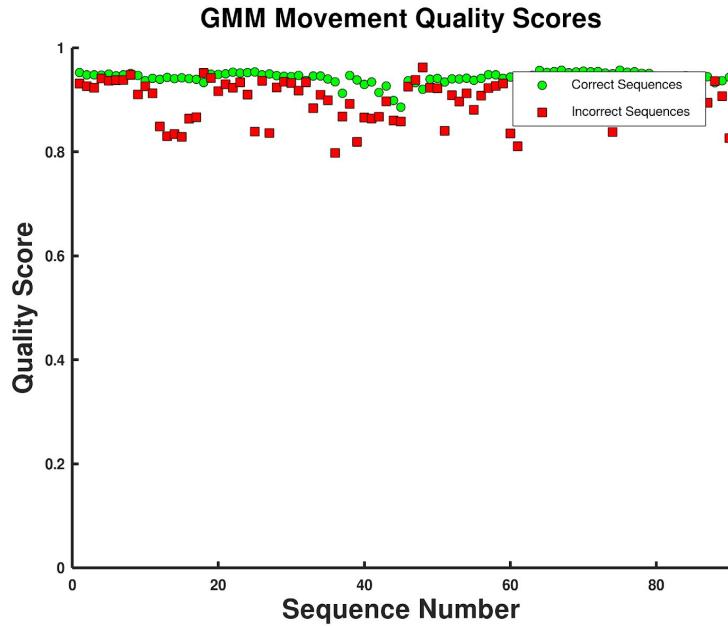
$$\bar{x}_k = (1 + e^{(x_k/(\mu+3\delta)-\alpha_1)} + (y_k - x_k)/\alpha_2 (\mu + 3\delta)) - 1 \dots \dots (2)$$

The proposed scoring function is monotonically decreasing, and is designed to preserve the distribution of the values of the performance metric. The values for the reference movements  $x_k$  are scaled by  $\mu + 3\delta$  in (1) to ensure that the resulting scores  $\bar{x}_k$  have values close to 1 for inputs  $x_k$  in the range  $(\mu - 3\delta, \mu + 3\delta)$ . Similarly, for the patient movements  $y_k$  the scoring function in (2) is designed to preserve their distribution in mapping the performance metric values into movement quality scores.

Figure 30 and 31 are quality scores for deep squat and hurdle steps.



*Fig 30: Quality Score of deep squat*



*Fig 31: Quality Score hardle step*

The green points are movement quality score for correct sequence and red points are scores for incorrect sequences.

## 7. CNN:

We trained a neural network model to predict the movement quality score. There are two inputs for correct movement like data correct and label correct while for incorrect movement it is data incorrect and label incorrect. The shape of the correct sequence (90, 240, 117) and correct label is (90,1). Same goes for incorrect movement. We tuned our CNN with an extra deep layer and found more accuracy compared to existing work. Training and validation data set has been splitted on a 70/30 ratio. For training the shape of correct\_data (124, 240, 117) and correct\_label (124,1 ) for validation it is (56, 240, 117) and (56,1). There are 1000 epochs performed for each exercise model.

|                           |             |       |
|---------------------------|-------------|-------|
| leaky_re_lu_5 (LeakyReLU) | (None, 200) | 0     |
| dropout_5 (Dropout)       | (None, 200) | 0     |
| dense_3 (Dense)           | (None, 100) | 20100 |
| leaky_re_lu_6 (LeakyReLU) | (None, 100) | 0     |
| dropout_6 (Dropout)       | (None, 100) | 0     |
| dense_4 (Dense)           | (None, 1)   | 101   |
| activation_1 (Activation) | (None, 1)   | 0     |

---

Total params: 302,201  
Trainable params: 302,201  
Non-trainable params: 0

*Fig 32: Model Architecture*

**Table 2: Mean absolute deviation & RMS deviation for each of exercises(Using Angle data)**

| Exercises                          | Mean absolute deviation | RMS deviation        |
|------------------------------------|-------------------------|----------------------|
| Deep Squat                         | 0.010262117281516455    | 0.01321338856330796  |
| Hurdle step                        | 0.0817762679938183      | 0.1536407348633146   |
| Inline lunge                       | 0.007134171914768174    | 0.013895806025068218 |
| Side lunge                         | 0.007793545254825314    | 0.010772385267499575 |
| Sit to stand                       | 0.013028141958367453    | 0.02125511493724838  |
| Standing active straight leg raise | 0.013576450519497409    | 0.019292298090150872 |
| Standing shoulder abduction        | 0.012232145881737055    | 0.01693986935765386  |
| Standing shoulder extension        | 0.015611735906737891    | 0.023132788035164106 |

|  |                      |                      |
|--|----------------------|----------------------|
| Standing shoulder internal-external rotation | 0.009675207733175716 | 0.016763743975679955 |
| Standing shoulder scaption                   | 0.009024251317994376 | 0.017048252090633497 |
| Avg  | 0.018011             | 0.030595             |

Our CNN model will predict how accurate the movement quality is. It's quality score will be the mean of 90 predictions of those 90 repetitions of the exercise.

**Table 3 : Movement Quality score prediction for each of exercises(Using Angle data)**

| Exercises                                    | Movement quality prediction |
|--|-----------------------------|
| Deep Squat                                   | 91.93515014648438 %         |
| Hurdle step                                  | 70.8264389038086 %          |
| Inline lunge                                 | 96.89152526855469 %         |
| Side lunge                                   | 92.14415740966797 %         |
| Sit to stand                                 | 90.99810791015625 %         |
| Standing active straight leg raise           | 92.55709075927734 %         |
| Standing shoulder abduction                  | 92.2518310546875 %          |
| Standing shoulder extension                  | 91.20230865478516 %         |
| Standing shoulder internal-external rotation | 92.58548736572266 %         |
| Standing shoulder scaption                   | 93.32059478759766 %         |
| Avg  | 90.47%                      |

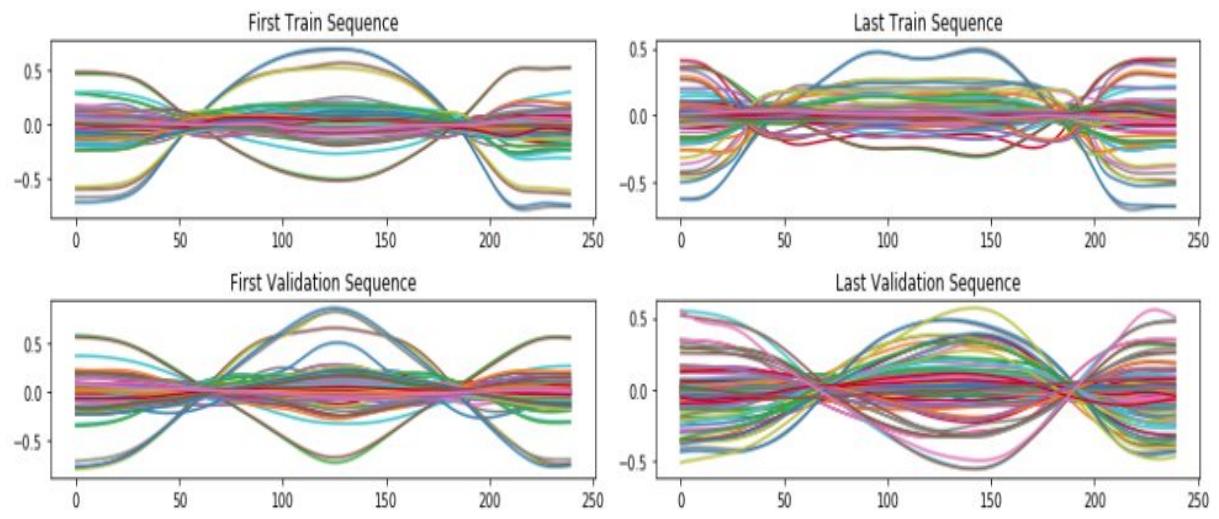


Fig 33: Deep Squat Train and Validation Sequence

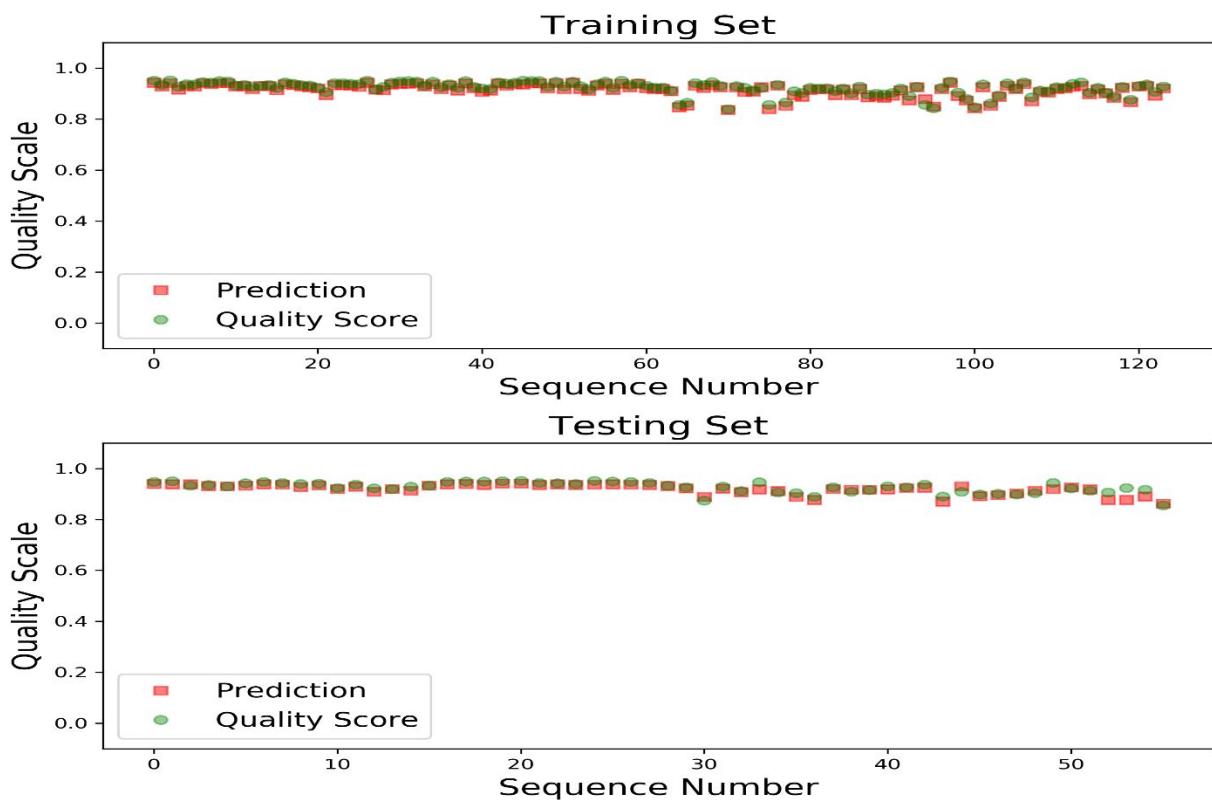
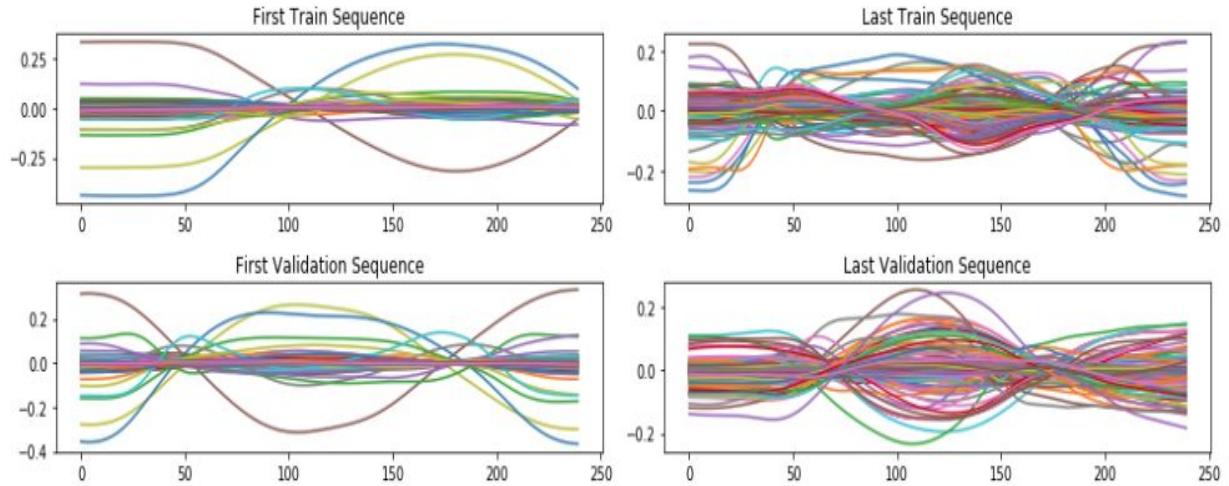
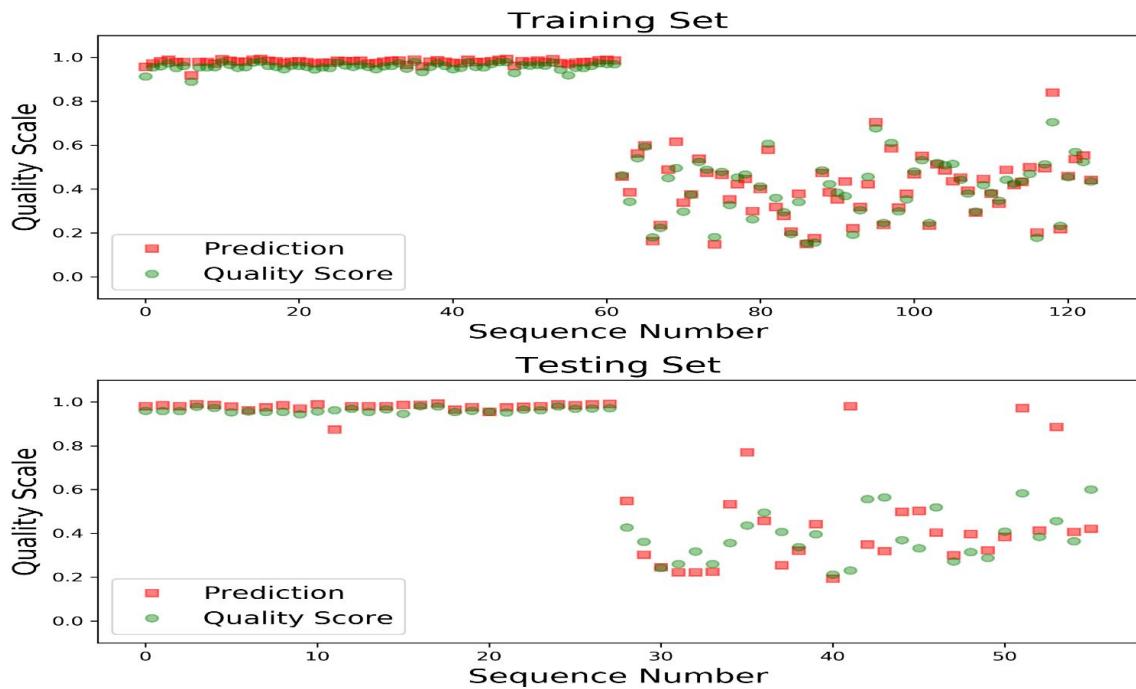


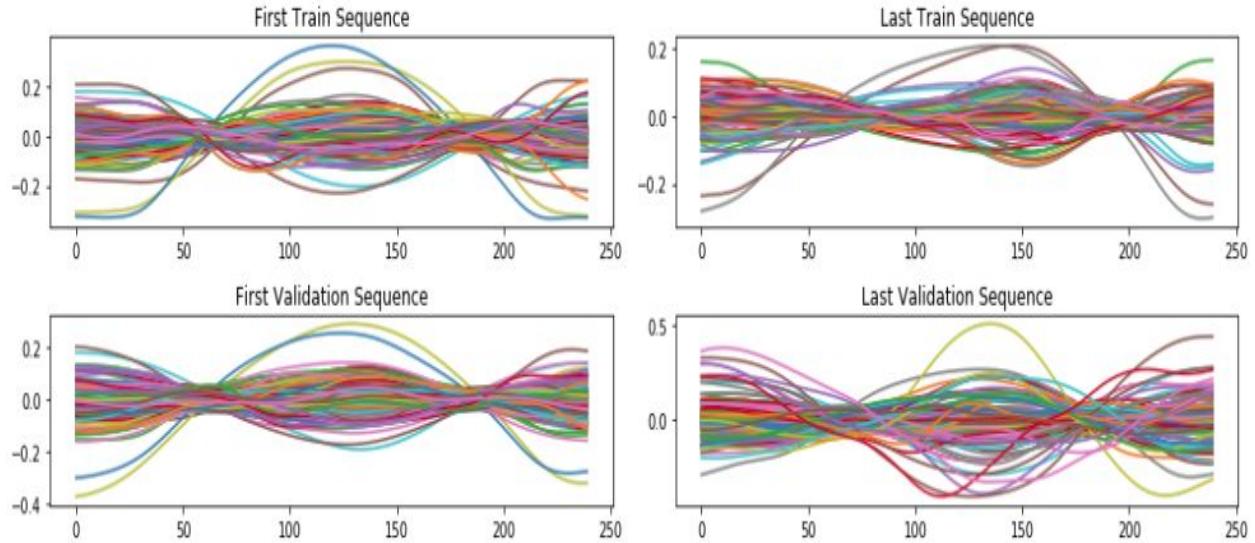
Fig 34: Movement Quality Score of deep squat



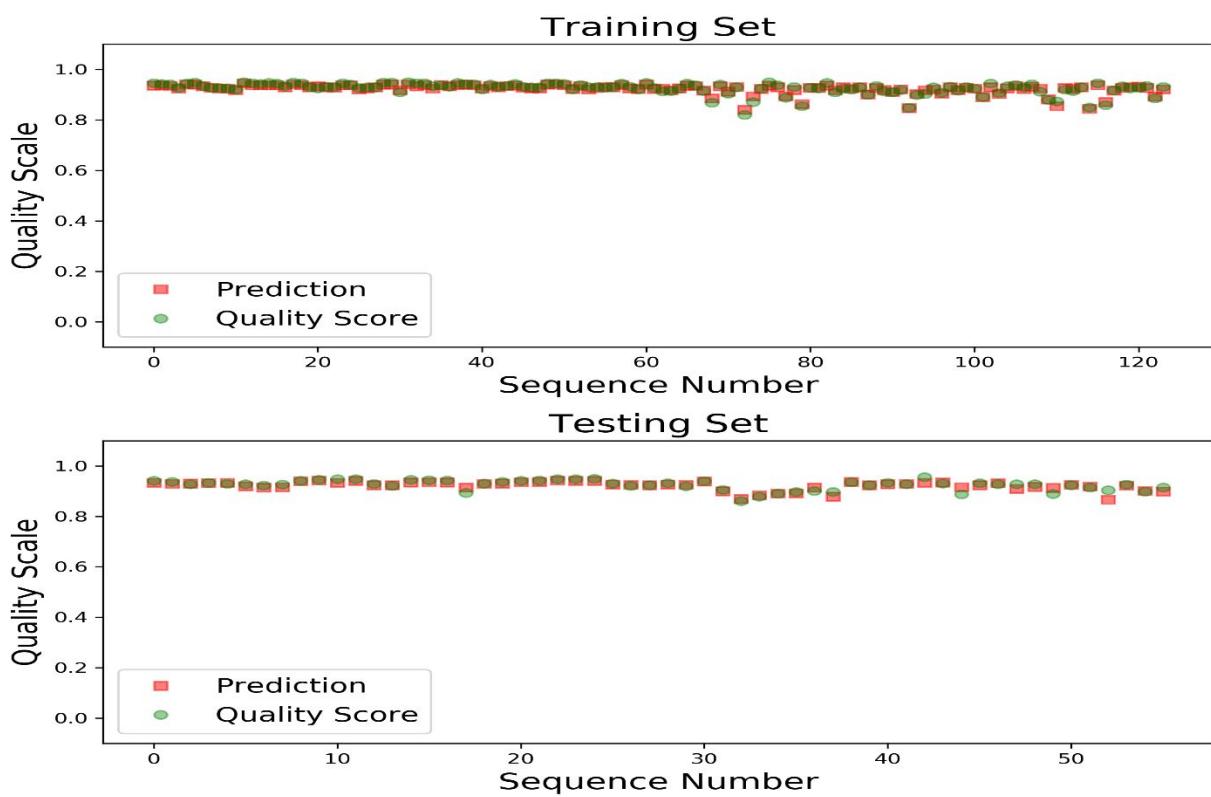
*Fig 35: Hurdle Step Train and Validation Sequence*



*Fig 36: Movement Quality Score of Hurdle squat*



*Fig 37: Side Lunge Train and Validation Sequence*



*Fig 38 : Movement Quality Score of Inline Lunge*

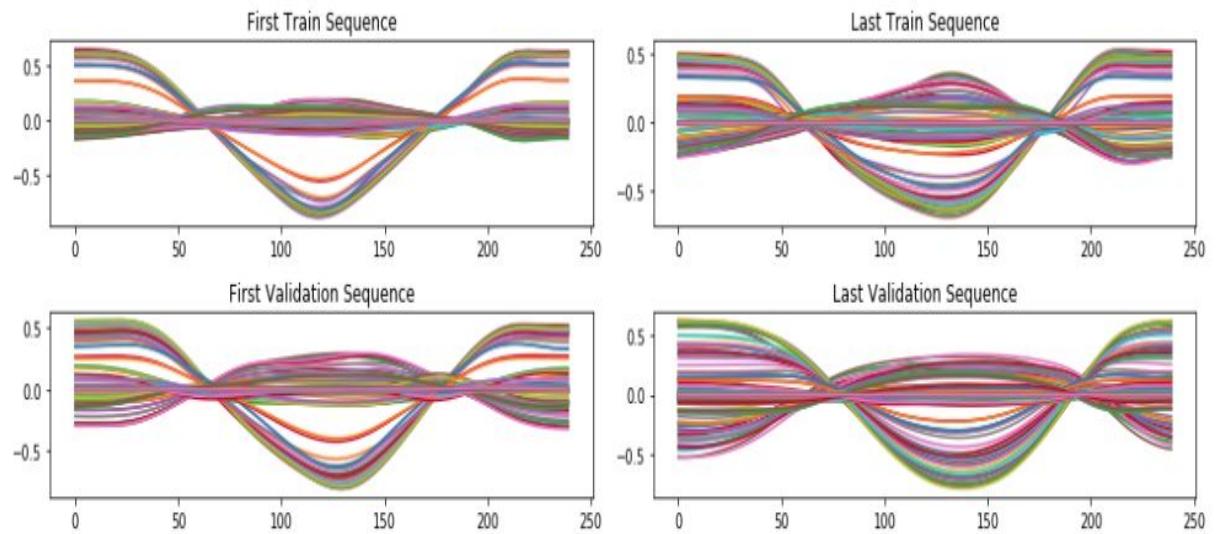
We also consider to use position data for CNN.

**Table 5 : Mean absolute deviation & RMS deviation for each of exercises(Using Position data)**

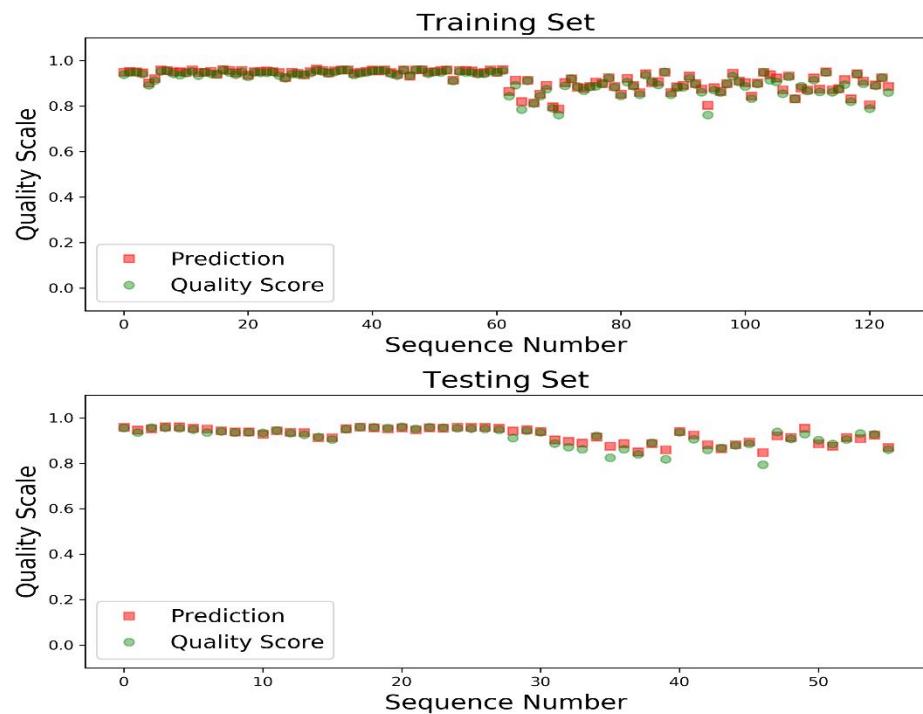
| Exercises                                    | Mean absolute deviation | RMS deviation         |
|--|-------------------------|-----------------------|
| Deep Squat                                   | 0.0108799006460993795   | 0.015969158527668396  |
| Hurdle step                                  | 0.011778462371359513    | 0.01976695273776133   |
| Inline lunge                                 | 0.0037638566805740675   | 0.0073740660345101156 |
| Side lunge                                   | 0.01127764260342966     | 0.01650261564714388   |
| Sit to stand                                 | 0.013199167849318265    | 0.3041339804019248    |
| Standing active straight leg raise           | 0.00915159536809694     | 0.0200708773420611    |
| Standing shoulder abduction                  | 0.031303165688362326    | 0.36425277191613403   |
| Standing shoulder extension                  | 0.02561301177821503     | 0.03262167508001736   |
| Standing shoulder internal-external rotation | 0.01440919829339189     | 0.023293320021282496  |
| Standing shoulder scaption                   | 0.012586421201744987    | 0.020604679731959823  |
| Avg  | 0.014396                | 0.082459              |

**Table 6: Movement Quality score prediction for each of exercises(Using Position data)**

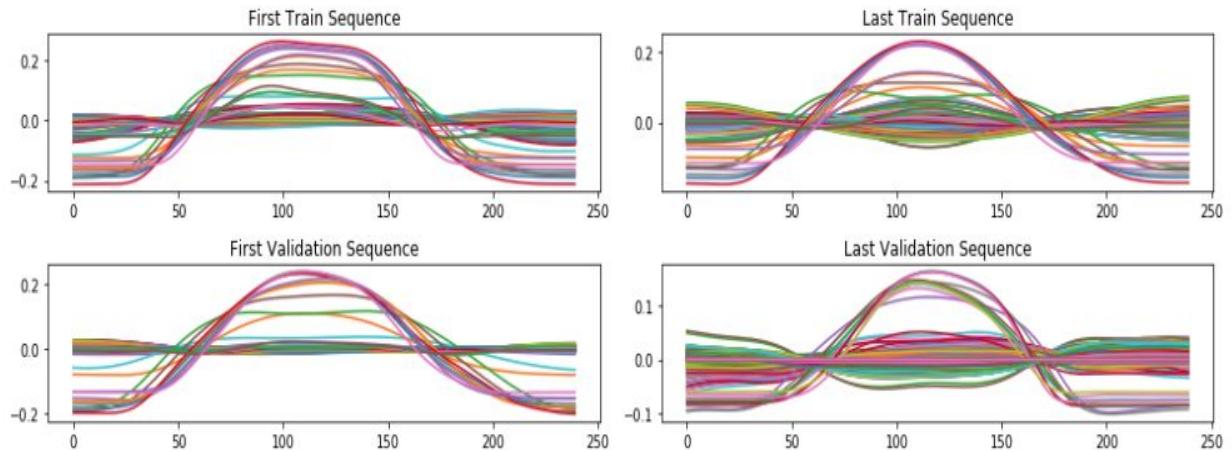
| Exercises                                    | Movement quality prediction |
|--|-----------------------------|
| Deep Squat                                   | 92.41056823730469 %         |
| Hurdle step                                  | 92.2544174194336 %          |
| Inline lunge                                 | 97.2772445678711 %          |
| Side lunge                                   | 92.90628051757812 %         |
| Sit to stand                                 | 91.21318817138672 %         |
| Standing active straight leg raise           | 92.9926986694336 %          |
| Standing shoulder abduction                  | 87.99068450927734 %         |
| Standing shoulder extension                  | 88.08271026611328 %         |
| Standing shoulder internal-external rotation | 89.572509765625 %           |
| Standing shoulder scaption                   | 91.57023620605469 %         |
| Average Accuracy                             | 91.63%                      |



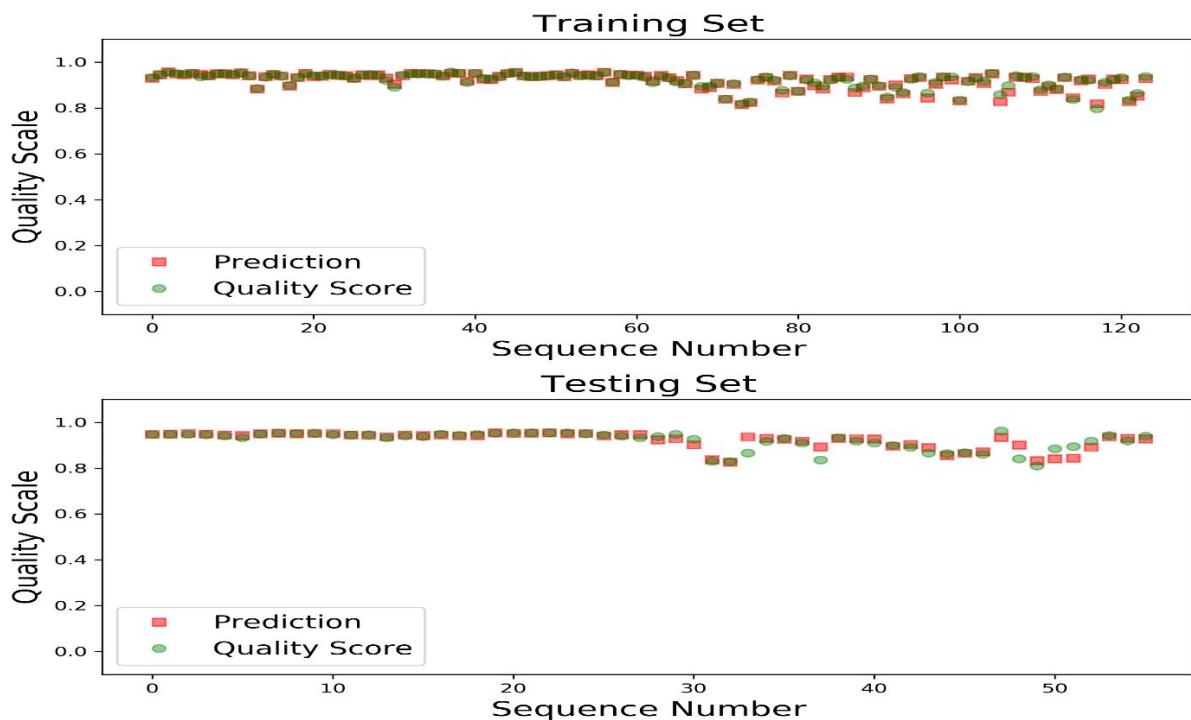
*Fig 39: Deep Squat Train and Validation Sequence*



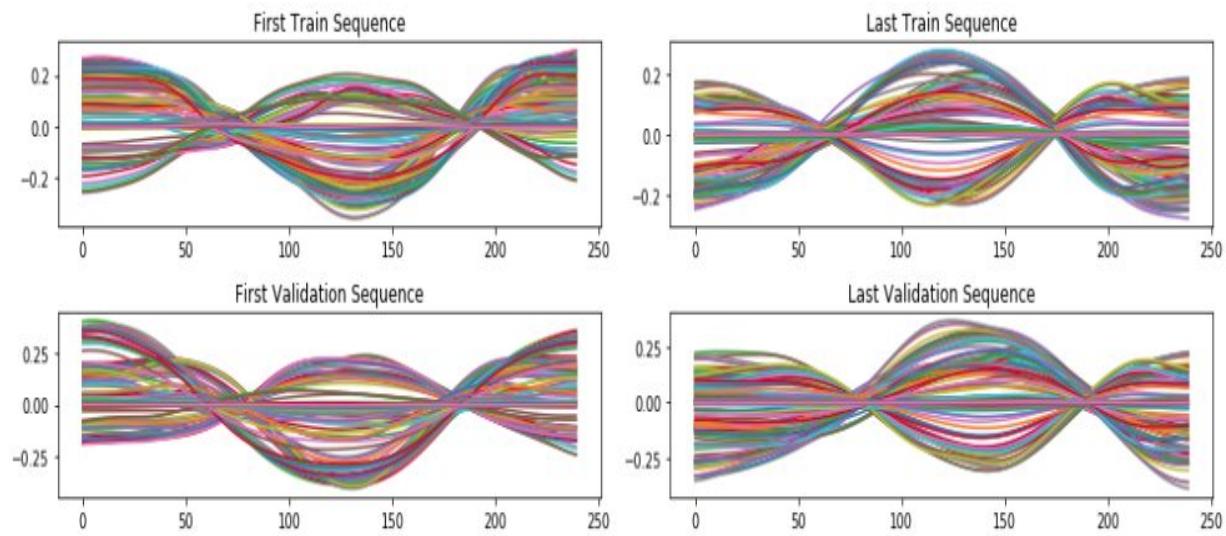
*Fig 40: Movement Quality Score of deep squat*



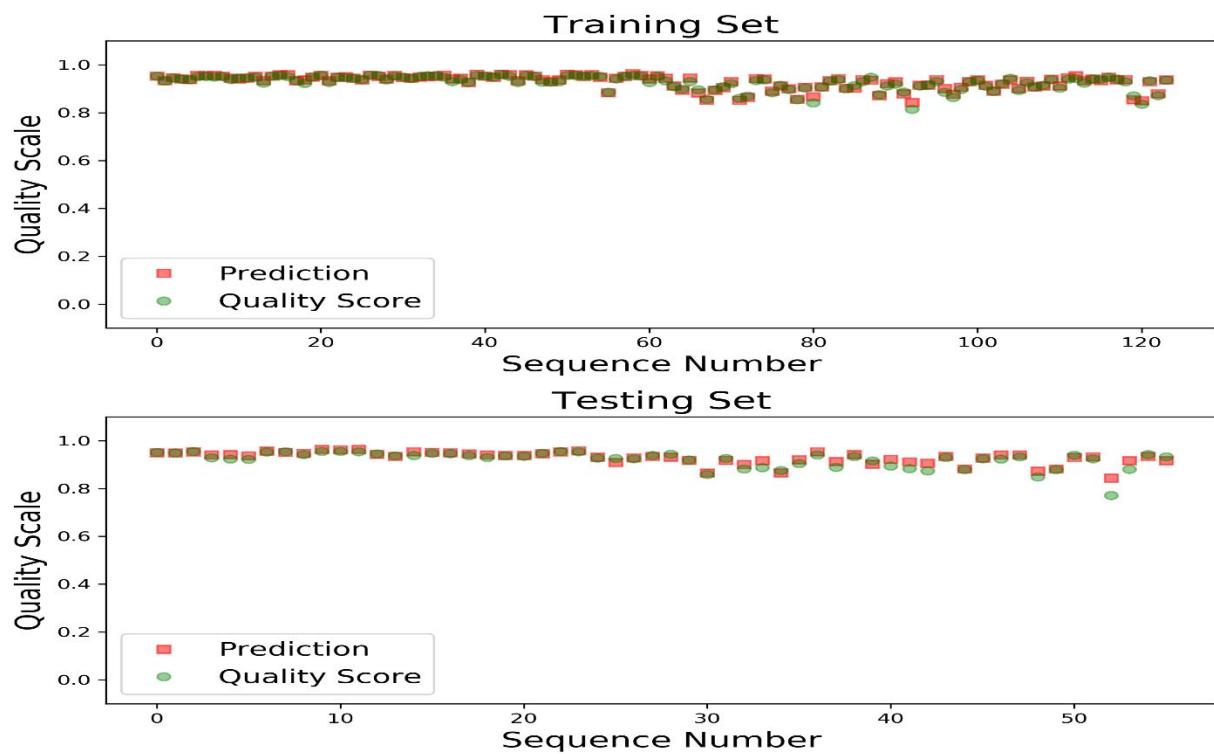
*Fig 41: Hurdle Step Train and Validation Sequence*



*Fig 42 : Movement Quality Score of Hurdle squat*



*Fig: 43 Side lunge Train and Validation Sequence*



*Fig 44 : Movement Quality Score of Inline Lunge*

## 8. RNN:

In this section, we will develop an RNN, specifically Long Short-Term Memory network model (LSTM) for evaluating in our dataset.

LSTM network models are a type of recurrent neural network (RNN) that are able to learn and remember over long sequences of input data. They are intended for use with data that consists of long sequences of data, up to 200 to 400 time steps (Typically). So, we considered it might be a good option for our problem.

The model can support multiple parallel sequences of input data, such as each axis of the accelerometer, gyroscope, and other sensor data. The model learns to extract features from sequences of observations and how to map the internal features to different activity and movement types.

The benefit of using LSTMs for sequence classification is that they can learn from the raw time series data directly, and in turn do not require domain expertise to manually engineer input features. The model can learn an internal representation of the time series data and ideally achieve comparable performance to models fit on a version of the dataset with engineered features. However, for our purpose we did some feature engineering as mentioned in Section 4.

**8.1 Training Data:** The training dataset is in CSV format where columns are separated by whitespace. Each of these files can be loaded as a NumPy array. The *load\_data()* function is used to load the dataset from given the full path to the file and returns the loaded data as a NumPy array. For training data, we used both correct and incorrect movements in terms of both angle and positions. The data for correct and incorrect are stored in two distinct files for each exercise. The corresponding labels are also stored in two different csv files.

Data shape for sequences: (90, 240, 117 )

Data shape for labels: (90, 1)

**8.2 Network Architecture and Training Parameters:** We will define the model as having a two bidirectional LSTM layers which is followed by a dropout layer intended to reduce overfitting of the model to the training data. Finally, a dense fully connected layer is used to interpret the features extracted by the LSTM layer, before a final output layer is used to make predictions. We used “tanh” and “sigmoid” as activation functions for two dense layers respectively.

The efficient Adam version of stochastic gradient descent will be used to optimize the network, and the Binary cross entropy will be used as a loss function. The other parameters are set to Batch size of 10, 50000 epochs, and validation split of 0.3 (i.e 70% of the input dataset is used for training, and 30% for validation).

The summary of the Network is given below:

| Layer (type)                                 | Output Shape    | Param # |
|--|-----------------|---------|
| bidirectional_3 (Bidirection (None, 240, 40) |                 | 22080   |
| dropout_4 (Dropout)                          | (None, 240, 40) | 0       |
| dense_3 (Dense)                              | (None, 240, 30) | 1230    |
| dropout_5 (Dropout)                          | (None, 240, 30) | 0       |
| bidirectional_4 (Bidirection (None, 20)      |                 | 3280    |
| dropout_6 (Dropout)                          | (None, 20)      | 0       |
| dense_4 (Dense)                              | (None, 1)       | 21      |
| activation_2 (Activation)                    | (None, 1)       | 0       |
| Total params: 26,611                         |                 |         |
| Trainable params: 26,611                     |                 |         |
| Non-trainable params: 0                      |                 |         |

*Fig 45: Parameters of RNN.*

**Table 7: RNN generated Mean absolute deviation & RMS deviation for each of exercises (Using Angle data)**

| Exercises    | Mean Absolute Deviation | RMS Deviation       |
|--------------|-------------------------|---------------------|
| Deep Squat   | 0.016538421727248596    | 0.05168879675206005 |
| Hurdle step  | 0.0410309745836258      | 0.06713349605531332 |
| Inline lunge | 0.04083583738497326     | 0.0522473742609631  |
| Side lunge   | 0.032884103128569465    | 0.06763909040129423 |
| Sit to stand | 0.02636703241177968     | 0.04035516808902225 |

|  |                      |                     |
|--|----------------------|---------------------|
| Standing active straight leg raise           | 0.04041850277457919  | 0.06348237451165081 |
| Standing shoulder abduction                  | 0.03714038000651769  | 0.06316627471145711 |
| Standing shoulder extension                  | 0.044631842707225255 | 0.08345307895249386 |
| Standing shoulder internal-external rotation | 0.045470766263008125 | 0.07507104183795454 |
| Standing shoulder internal-external rotation | 0.031255450862816406 | 0.05169744460573718 |

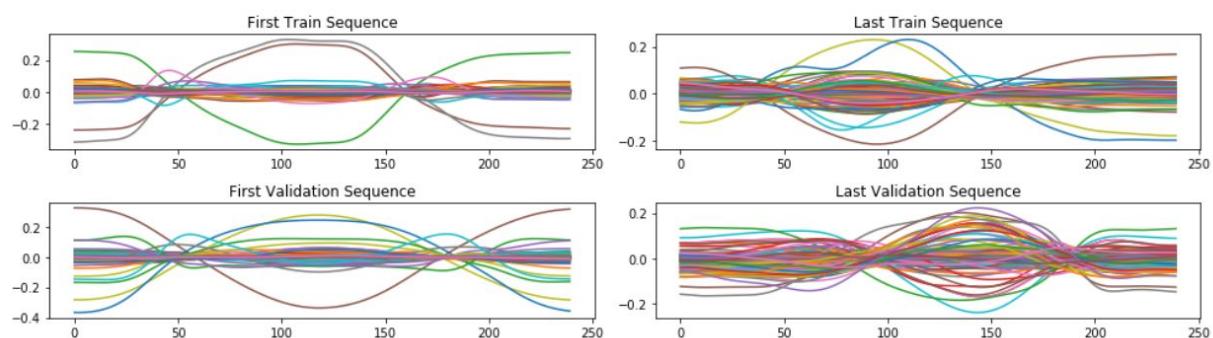


Fig 46: Hurdle step Training and validation Sequence

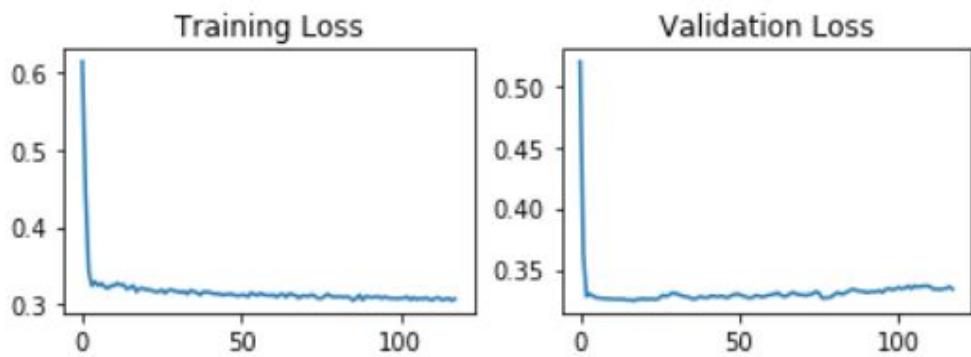


Fig 47: Training and Validation Loss for Hurdle step

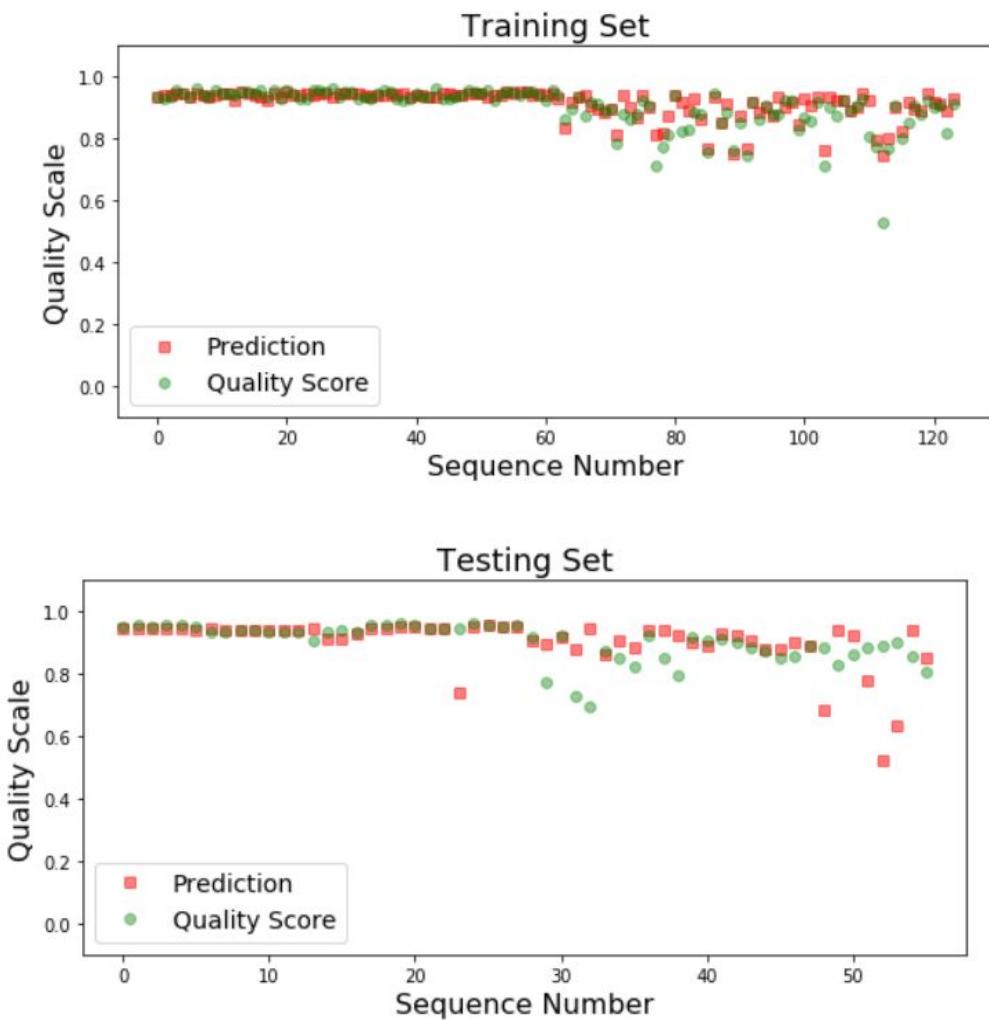


Fig 48: Quality Scores for Hurdle Step

## 9. Discussion :

For training the deep neural networks, the movement quality scores based on the GMM log-likelihood calculated with autoencoder-reduced data are employed. Only the case of between-subject is considered, since for the within-subject case the number of repetitions per subject is too small to train NNs.

For CNN we tried to predict movement quality score both from angle and position data of vicon sensor. Adding an extra layer and using the autoencoder output data makes our CNN giving higher accuracy than existing work. For training and testing we could not do subject wise validation, future work can be done one predicting unseen subject data. After that we can understand how good our model is on those novel data.

## 10. Links of Project Repository

**Data after prepossessing and autoencoders output for every movement can be found here**

<https://drive.google.com/drive/folders/1xO6kzKTjakLkbNP-pEbeHeQlLQQqDjn?usp=sharing>

**Autoencoder code :** [https://drive.google.com/open?id=1c\\_iduFccgpjskYjZkLk\\_tDIFJVoz\\_XXL](https://drive.google.com/open?id=1c_iduFccgpjskYjZkLk_tDIFJVoz_XXL)

**CNN results:** <https://drive.google.com/drive/folders/1QjktuzjJTiTIP7QQ6vDjtGUhZDZPz5H0?usp=sharing>

<https://drive.google.com/drive/folders/1QjktuzjJTiTIP7QQ6vDjtGUhZDZPz5H0?usp=sharing>

**CNN code:** [https://drive.google.com/drive/u/1/folders/1-8wnBCpyDsqHpLwXOi70AYnz1PeAB\\_vA](https://drive.google.com/drive/u/1/folders/1-8wnBCpyDsqHpLwXOi70AYnz1PeAB_vA)

[https://drive.google.com/drive/u/1/folders/1-8wnBCpyDsqHpLwXOi70AYnz1PeAB\\_vA](https://drive.google.com/drive/u/1/folders/1-8wnBCpyDsqHpLwXOi70AYnz1PeAB_vA)

[ Our whole project drive is given ( but it is not organised well enough . anyone might not understand where is everything for the first time, still we are giving the link )

<https://drive.google.com/drive/folders/1QjktuzjJTiTIP7QQ6vDjtGUhZDZPz5H0?usp=sharing> ]

## 11. References:

- <https://arxiv.org/abs/1901.10435>
- <https://ieeexplore.ieee.org/abstract/document/8957502>
- <https://webpages.uidaho.edu/ui-prmd/>
- <https://github.com/avakanski/A-Deep-Learning-Framework-for-Assessing-Physical- Rehabilitation-Exercise>