In [1]:

```
from CreatingGraphs import createLRGraphs
createLRGraphs()
```

Accuracy Measurements:

Here is the accuracy of our algorithm when the training set, test set, and cross validation set is passed in

```
Training Accuracy: 1.0
Test Accuracy: 0.9991666666666666
Validation Accuracy: 0.9983333333333333
```

Inaccurate Trajectories

Here is some more information on the trajectories it predicted incorrectly. It displays the indexes of the incorrect trajectories, followed by the actual diffusion type and the incorrect predicted diffusion type.

1 = Ballistic Diffusion, 2 = Confined Diffusion, 3 = Random Walk, 4 = Very Confined Diffusion

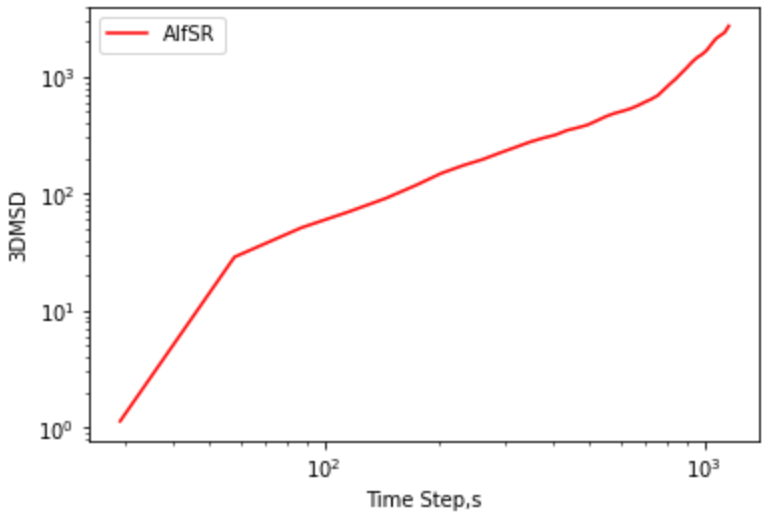
```
Indexes of incorrect predictions in testing:
3289,
Actual Diffusion Types:
3,
Incorrect predictions:
1,

Indexes of incorrect predictions in validation:
3746, 3886,
Actual Diffusion Types:
3, 3,
Incorrect predictions:
1, 1,
```

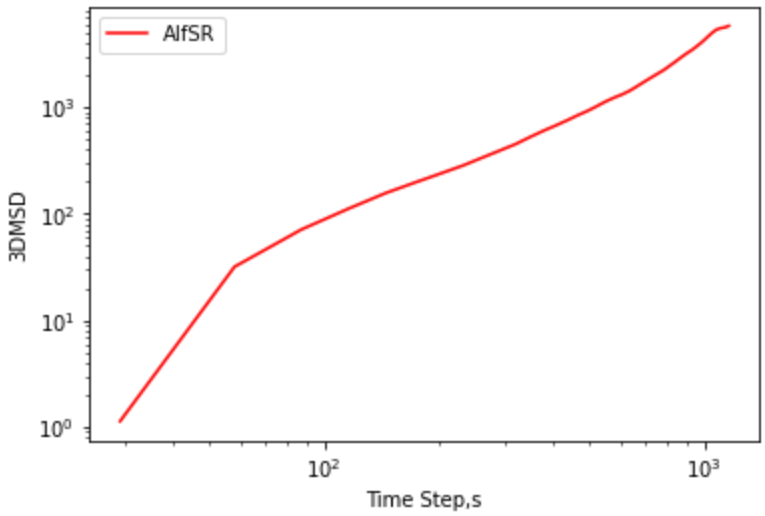
Graphs of Incorrect Trajectories:

Here is the graphs of the trajectories that were predicted incorrectly

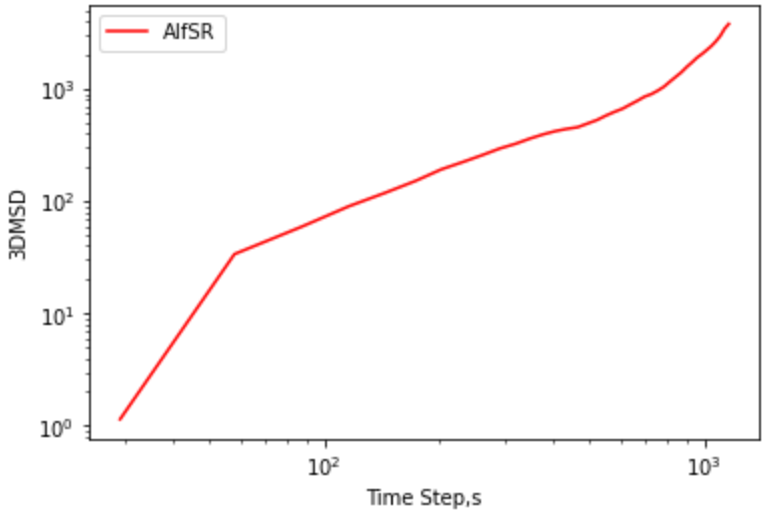
data/02_01_Simulated_trajectories/Simple_cases/Random_walk/trajectories/random_289.tck



data/02_01_Simulated_trajectories/Simple_cases/Random_walk/trajectories/random_746.tck



data/02_01_Simulated_trajectories/Simple_cases/Random_walk/trajectories/random_886.tck



Predictions of Dr. Mzyk's Data:

Here is the predictions of our algorithm when Dr. Mzyk's data is passed in

```
M0: 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 2, 2, 2, 2,
M1: 2, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 1,
M2: 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
```

Analytics of Predictions:

Here is some percentages and information derived from the predictions of the algorithm

```
M0:     1: 0.00%        2: 86.667%      3: 0.000%       4: 13.333%
M1:     1: 5.88%        2: 88.235%      3: 0.000%       4: 5.882%
M2:     1: 0.00%        2: 100.000%     3: 0.000%       4: 0.000%
Ovr:    1: 1.96%        2: 92.157%      3: 0.000%       4: 5.882%
```