CPSC-5616: Assignment 4 Report

Distance Estimation of Robot Using IMU Data:

Deep Learning Approach

Group Member:

Roya Bakhshi

Leyon Ibn Kamal

1. Introduction

In this project, we design an advanced distance estimation system in robotics using the Inertial Measurement Unit (IMU) data captured via the Sensor Logger app on an Android device. The first part of the project lays a strong foundation by establishing a baseline using traditional regression techniques, namely Random Forest Regression (RFR) and Support Vector Regression (SVR). The IMU data is recorded in four different CSV files for the movement of the robot over a distance of 50 meters under different conditions of speed and navigation curves. This phase focuses on the extraction of important features that are crucial for distance estimation and lays a solid foundation for higher-order analysis.

With this ground, the second part of our work introduces a sophisticated deep learning approach to improve distance estimation accuracy. We will present a hybrid deep neural network designed by combining a model using a Convolution Neural Network and a Gated Recurrent Unit mechanism, in order to find rich temporal patterns in raw sensor data.

This deep learning model, trained on the dataset, compares and contrasts the effectiveness of traditional regression techniques with a more advanced analytical framework. The results of this research are important in advancing our knowledge and understanding of robotic navigation systems, opening the door to more innovations yet to come.

2. Dataset Preparation

Data Visualization

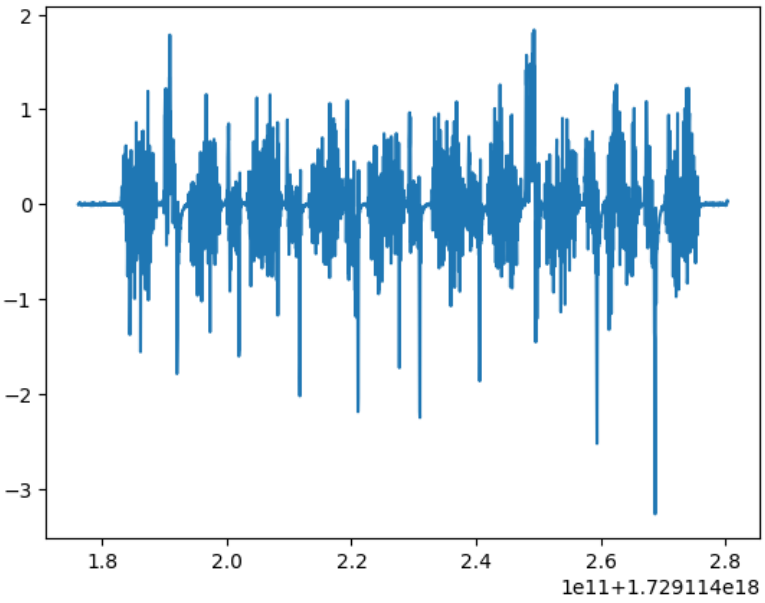
To understand the nature of our data, er first visualize different graphs available to us using the dataset. Data visualization will allow us to understand the complexity of the data, the amount of noise in it, a rough idea on how to clean the data and selecting the appropriate data for training or machine learning models.

Straight Slow Accelerometer Data

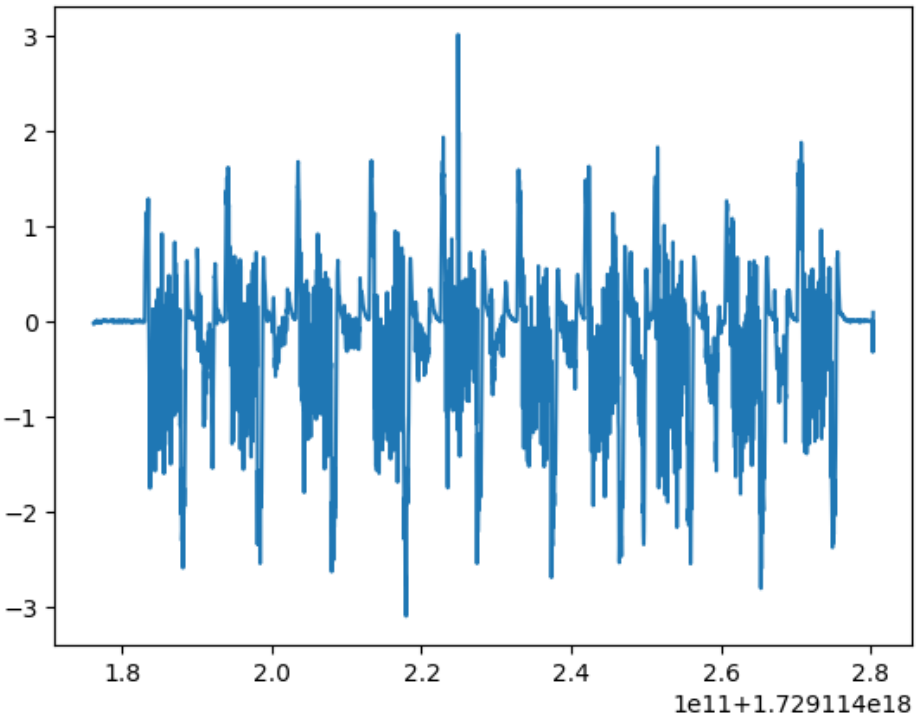
	time	seconds_elapsed	z	y	x
0	1729114176181976300	0.151976	0.138256	-0.022989	-0.004539
1	1729114176219911000	0.189911	0.199427	-0.032416	-0.003736
2	1729114176257845800	0.227846	0.118507	-0.017880	-0.005607
3	1729114176295780400	0.265780	0.130477	-0.019850	0.007658
4	1729114176333715200	0.303715	0.096210	-0.008136	0.011244
...	...	...	...	...	...
2742	1729114280198955500	104.168956	-0.003205	0.001204	0.001735
2743	1729114280236890000	104.206890	-0.001100	-0.001666	0.001348
2744	1729114280274824700	104.244825	-0.013041	0.005332	0.008535
2745	1729114280312759000	104.282759	1.260686	-0.324834	0.040107
2746	1729114280350694000	104.320694	-0.286513	0.092544	0.030367

2747 rows × 5 columns

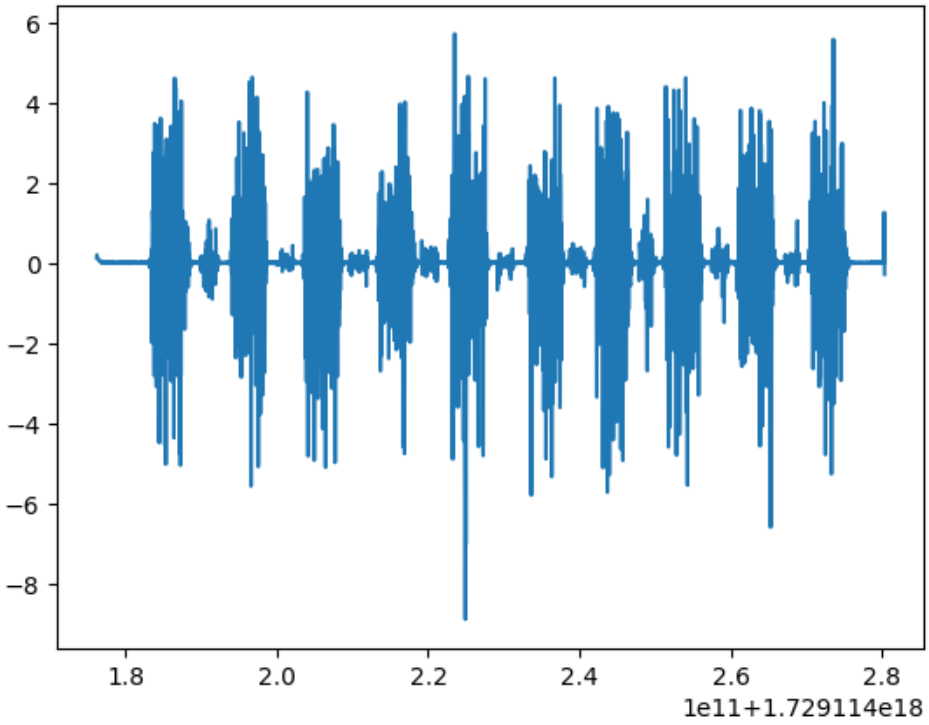
X Axis


2
1
0
−1
−2
−3
1.8
2.0
2.2
2.4
2.6
2.8
1e11+1.729114e18

Y Axis


3
2
1
0
−1
−2
−3
1.8
2.0
2.2
2.4
2.6
2.8
1e11+1.729114e18

Z Axis

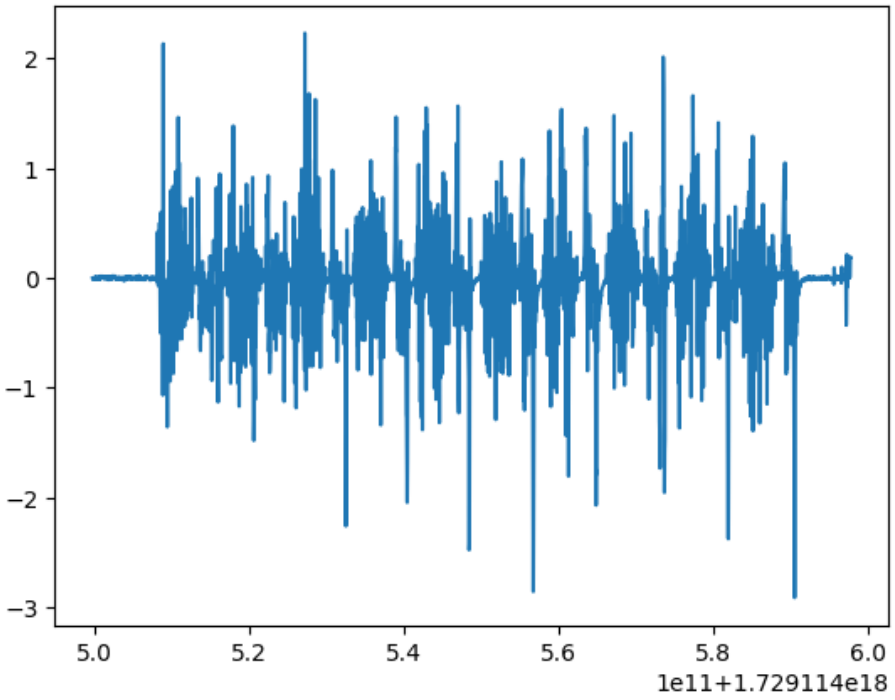


Straight Fast Accelerometer Data

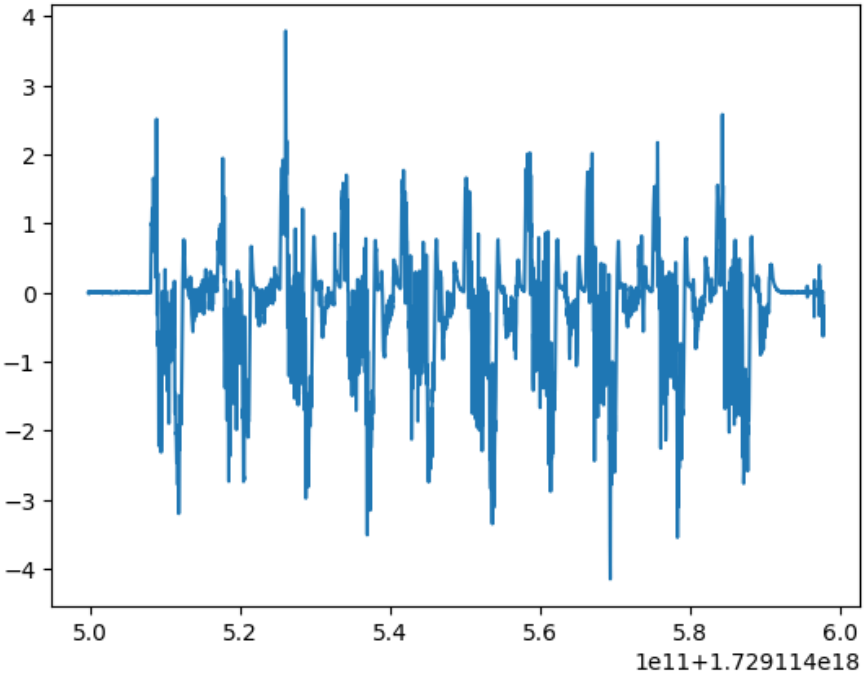
	time	seconds_elapsed	z	y	x
0	1729114499696945200	0.025945	0.000000	0.000000	0.000000
1	1729114499734879500	0.063879	-0.110129	0.009605	-0.011992
2	1729114499772813600	0.101813	0.054107	-0.020007	-0.013428
3	1729114499810748000	0.139748	-0.054379	0.005427	-0.014331
4	1729114499848682000	0.177682	0.046551	0.003205	-0.004356
...	...	...	...	...	...
2581	1729114597605043500	97.934043	0.036628	-0.004948	0.003000
2582	1729114597642977500	97.971978	0.084298	-0.001620	0.012393
2583	1729114597680911600	98.009912	0.033235	0.011912	0.011345
2584	1729114597718845700	98.047846	4.336559	-0.633932	0.193295
2585	1729114597756780000	98.085780	2.206986	-0.200281	0.184710

2586 rows × 5 columns

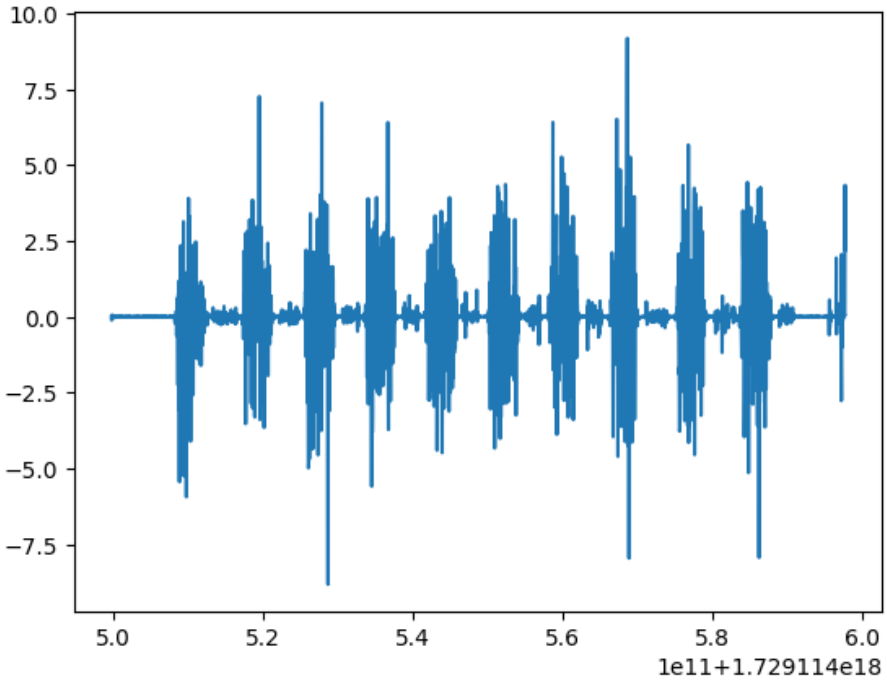
X Axis


2
1
0
−1
−2
−3
5.0
5.2
5.4
5.6
5.8
6.0
1e11+1.729114e18

Y Axis


4
3
2
1
0
−1
−2
−3
−4
5.0
5.2
5.4
5.6
5.8
6.0
1e11+1.729114e18

Z Axis

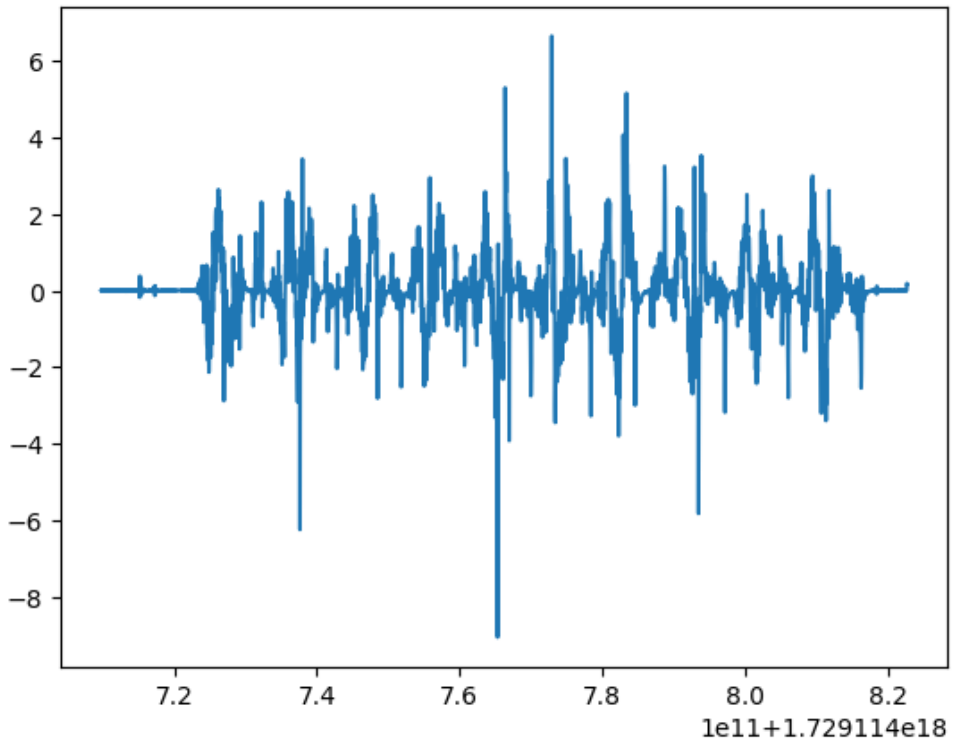


Curve Slow Accelerometer Data

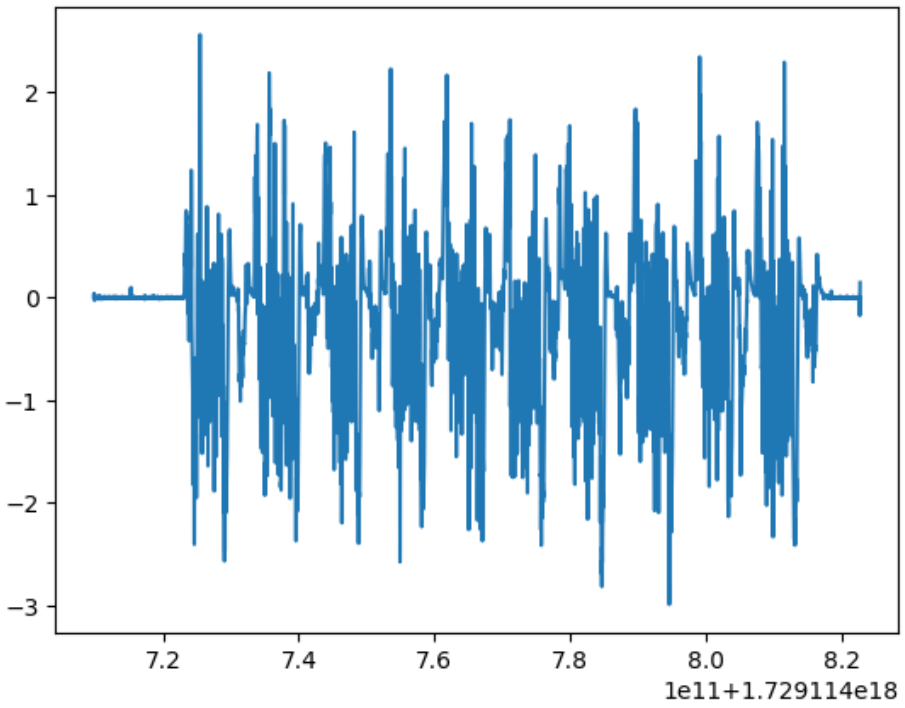
	time	seconds_elapsed	z	y	x
0	1729114709796172300	0.052172	0.000000	0.000000	0.000000
1	1729114709834106600	0.090107	-0.256169	0.037665	-0.009615
2	1729114709872040700	0.128041	0.160883	-0.024792	0.024385
3	1729114709909975000	0.165975	-0.163143	0.028428	-0.004211
4	1729114709947909400	0.203909	0.098420	-0.015674	0.012540
...	...	...	...	...	...
2973	1729114822575113700	112.831114	-0.000735	0.003801	-0.001142
2974	1729114822613048000	112.869048	-0.003042	-0.002776	0.010107
2975	1729114822650982400	112.906982	-0.002964	0.010719	0.002897
2976	1729114822688916700	112.944917	1.575144	-0.176965	0.185672
2977	1729114822726851000	112.982851	-0.666391	0.145894	0.161224

2978 rows × 5 columns

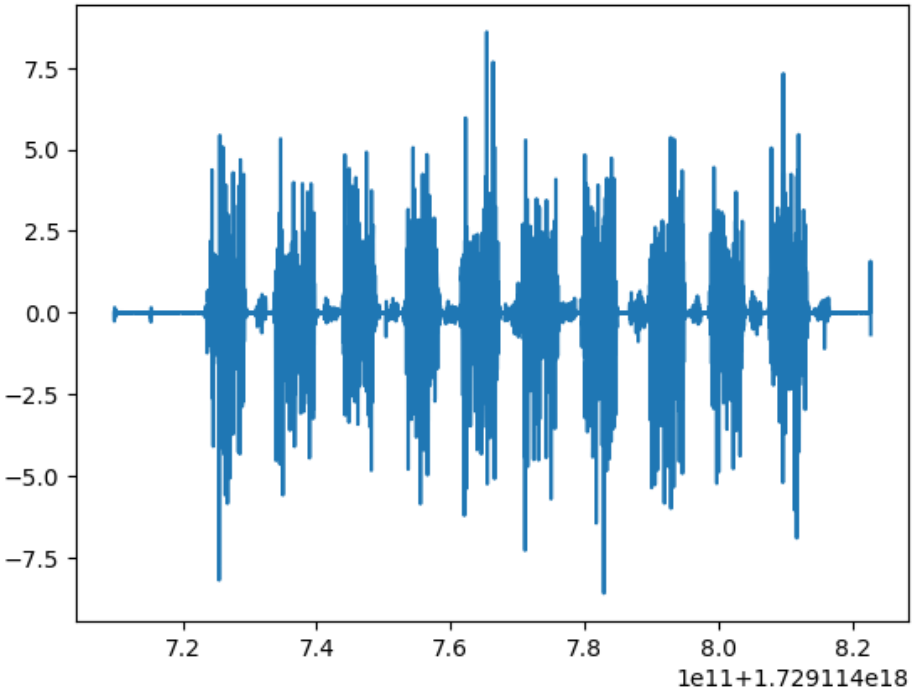
X Axis



Y Axis


2
1
0
−1
−2
−3
7.2
7.4
7.6
7.8
8.0
8.2
1e11+1.729114e18

Z Axis

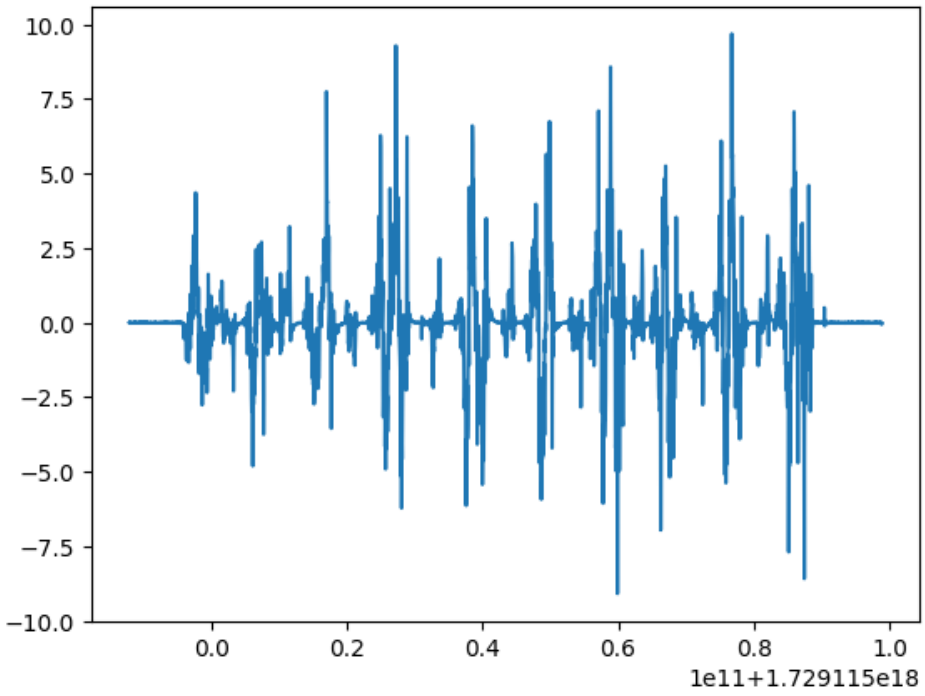

7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
7.2
7.4
7.6
7.8
8.0
8.2
1e11+1.729114e18

Curve Fast Accelerometer Data

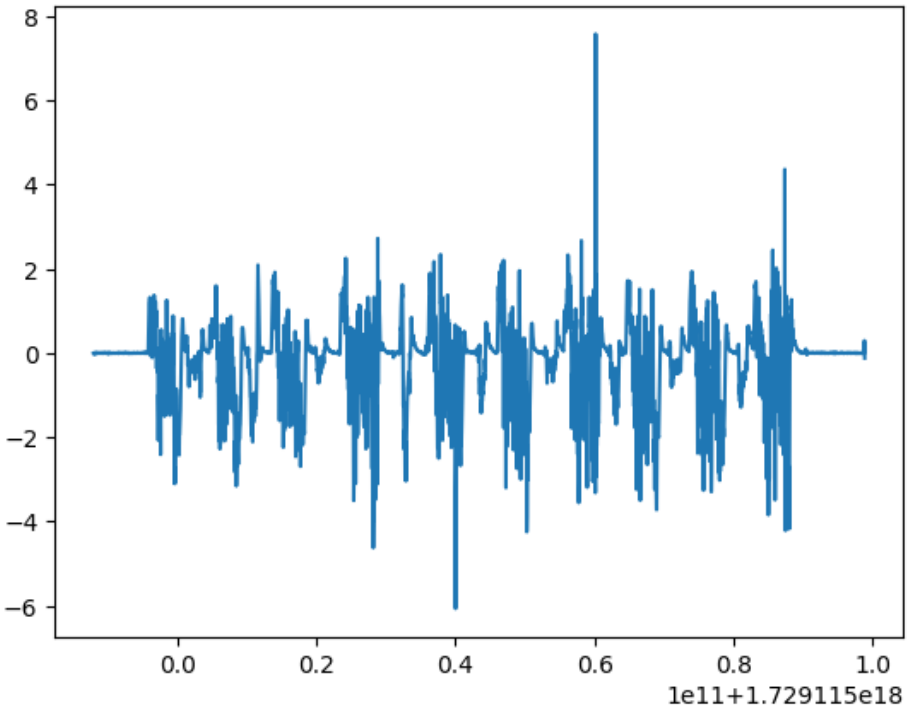
	time	seconds_elapsed	z	y	x
0	1729114987950340400	0.028340	0.000000	0.000000	0.000000
1	1729114987987895800	0.065896	0.001596	0.005874	-0.011253
2	1729114988025830400	0.103830	0.044533	-0.006611	0.016885
3	1729114988063764700	0.141765	-0.041039	-0.010302	-0.018762
4	1729114988101699300	0.179699	0.039904	-0.030601	0.003223
...	...	...	...	...	...
2920	1729115098718901500	110.796902	0.001064	-0.003592	-0.005816
2921	1729115098756836000	110.834836	0.003590	-0.013272	-0.008080
2922	1729115098794770400	110.872771	0.001181	0.001149	0.010580
2923	1729115098832705000	110.910705	-1.269129	0.293492	-0.059588
2924	1729115098870639600	110.948640	0.556180	-0.121369	-0.024277

2925 rows × 5 columns

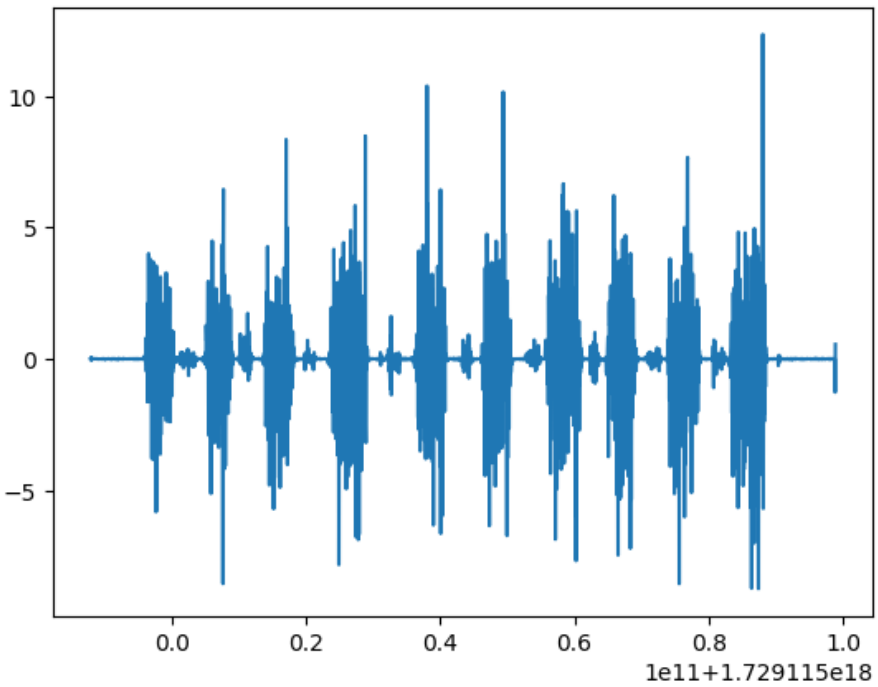
X Axis



Y Axis


8
6
4
2
0
−2
−4
−6
0.0
0.2
0.4
0.6
0.8
1.0
1e11+1.729115e18

Z Axis


10
5
0
−5
0.0
0.2
0.4
0.6
0.8
1.0
1e11+1.729115e18

- **Data Cleaning**

Based on observations of different graphs, we've concluded that the Z-AXIS graph created from the accelerometer data shows the most clear pattern that we can use for our models to get good results.

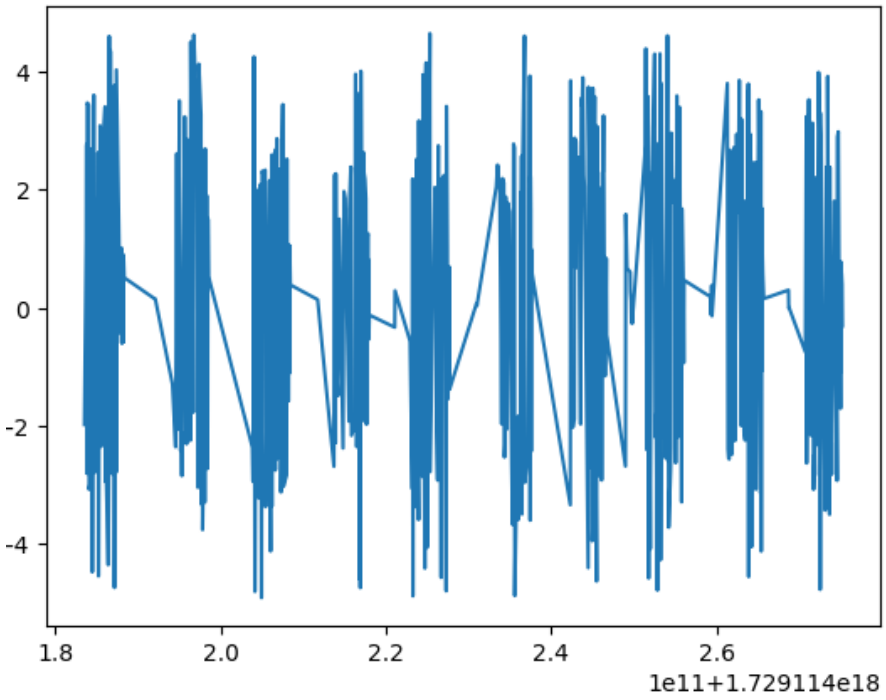
Next step is to clean our data from noise, we computed the magnitude of the points, and created active and not active labels to remove the non active labels.
After that we looked at the magnitude and set thresholds. Values higher or lower than certain thresholds were considered noise and removed so that the data have a standard range of data in it. Value lower than the threshold is when the robot is idle and we don't need idle data for our purpose. Value higher than the threshold is noise from either sensor anomalies or sudden movements. Cleaning the noise in the way we did allowed us to get clear graphs.

Cleaning Straight Slow Accelerometer

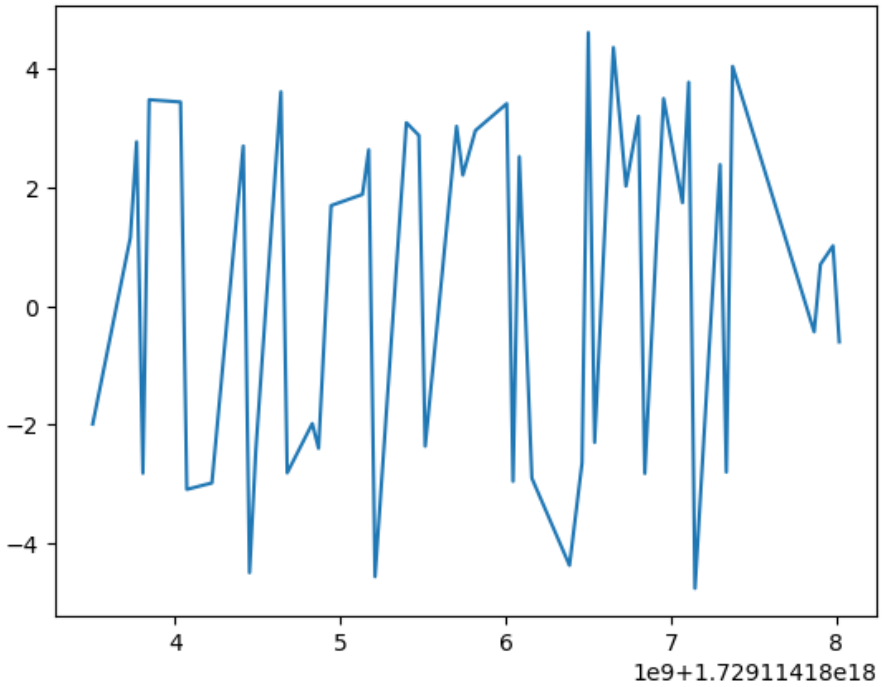
	time	seconds_elapsed	z	y	x	magnitude	active
0	1729114176181976300	0.151976	0.138256	-0.022989	-0.004539	0.140228	False
1	1729114176219911000	0.189911	0.199427	-0.032416	-0.003736	0.202079	False
2	1729114176257845800	0.227846	0.118507	-0.017880	-0.005607	0.119980	False
3	1729114176295780400	0.265780	0.130477	-0.019850	0.007658	0.132200	False
4	1729114176333715200	0.303715	0.096210	-0.008136	0.011244	0.097205	False
...	...	...	...	...	...	...	...
2742	1729114280198955500	104.168956	-0.003205	0.001204	0.001735	0.003838	False
2743	1729114280236890000	104.206890	-0.001100	-0.001666	0.001348	0.002409	False
2744	1729114280274824700	104.244825	-0.013041	0.005332	0.008535	0.016472	False
2745	1729114280312759000	104.282759	1.260686	-0.324834	0.040107	1.302480	False
2746	1729114280350694000	104.320694	-0.286513	0.092544	0.030367	0.302616	False

2747 rows × 7 columns

Z Axis



This graph represents the one full movement:

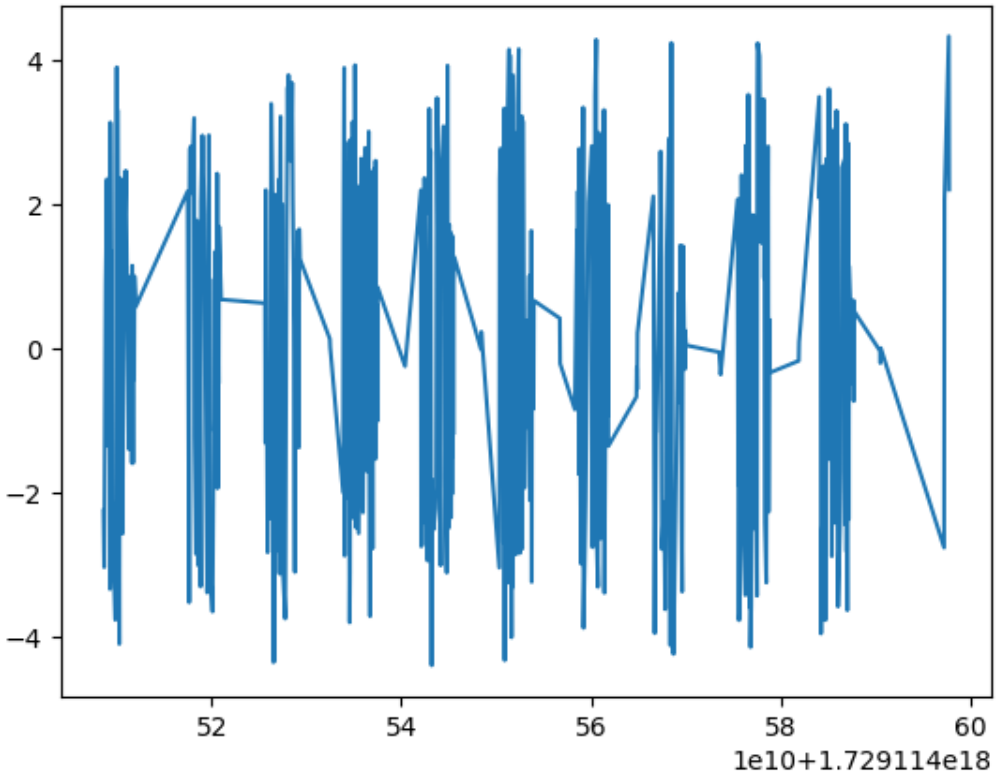


Cleaning Straight Fast Accelerometer

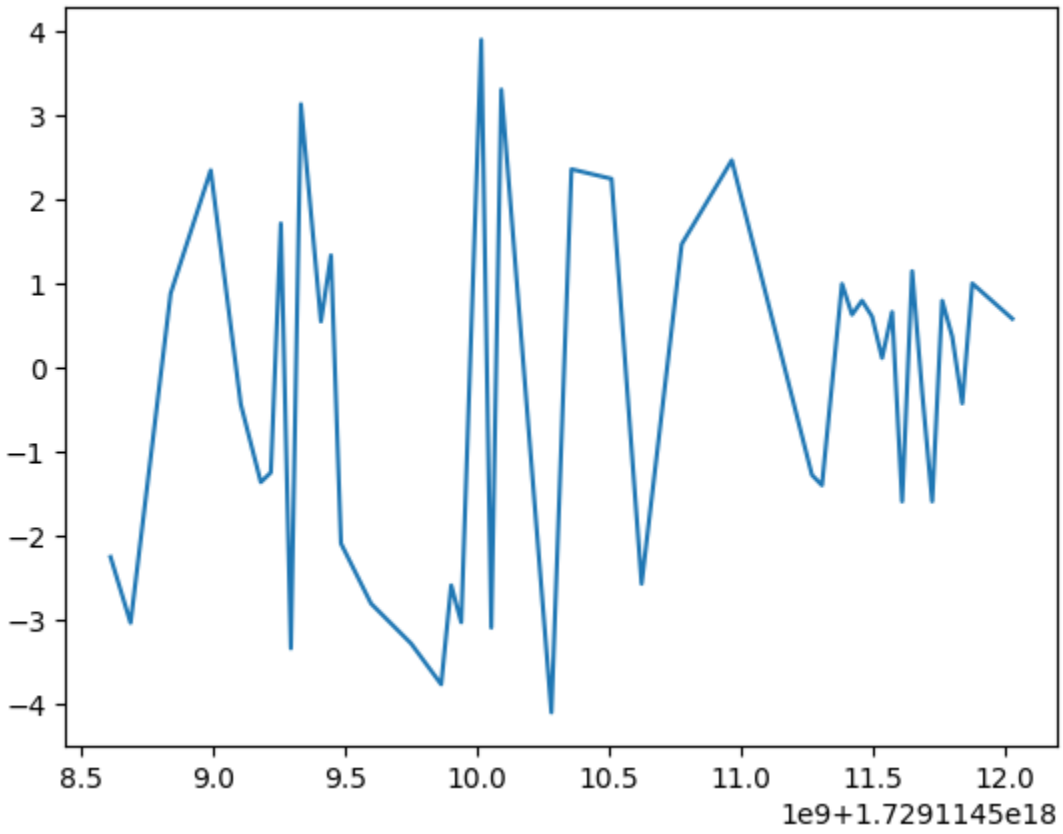
	time	seconds_elapsed	z	y	x	magnitude	active
0	1729114499696945200	0.025945	0.000000	0.000000	0.000000	0.000000	False
1	1729114499734879500	0.063879	-0.110129	0.009605	-0.011992	0.111196	False
2	1729114499772813600	0.101813	0.054107	-0.020007	-0.013428	0.059229	False
3	1729114499810748000	0.139748	-0.054379	0.005427	-0.014331	0.056497	False
4	1729114499848682000	0.177682	0.046551	0.003205	-0.004356	0.046864	False
...	...	...	...	...	...	...	...
2581	1729114597605043500	97.934043	0.036628	-0.004948	0.003000	0.037082	False
2582	1729114597642977500	97.971978	0.084298	-0.001620	0.012393	0.085220	False
2583	1729114597680911600	98.009912	0.033235	0.011912	0.011345	0.037083	False
2584	1729114597718845700	98.047846	4.336559	-0.633932	0.193295	4.386910	True
2585	1729114597756780000	98.085780	2.206986	-0.200281	0.184710	2.223740	True

2586 rows × 7 columns

Z Axis



This graph represents the one full movement:

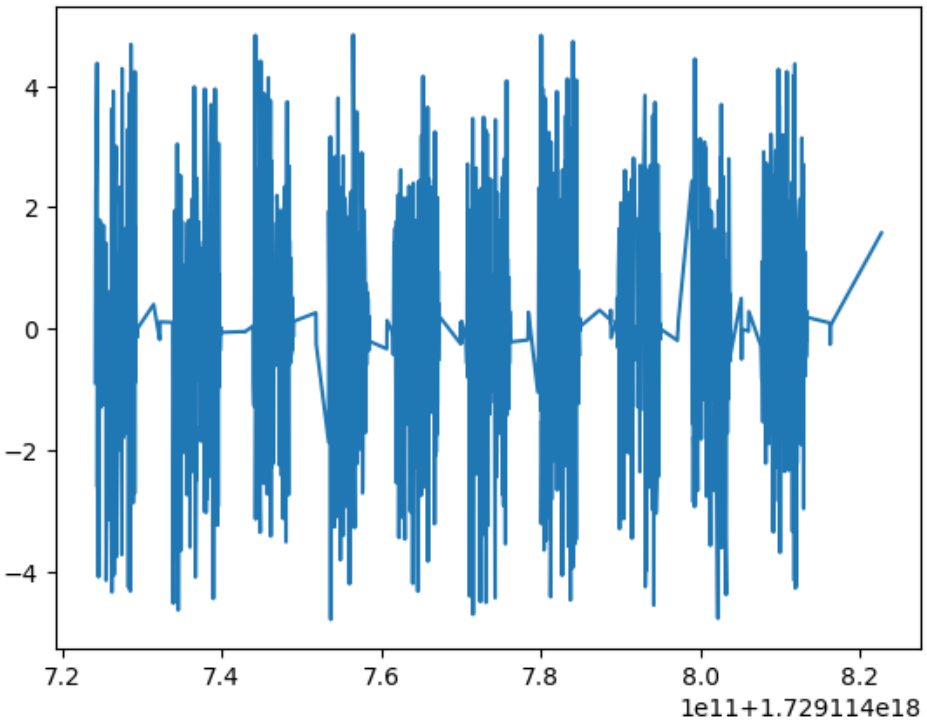


Cleaning Crve Slow Accelerometer

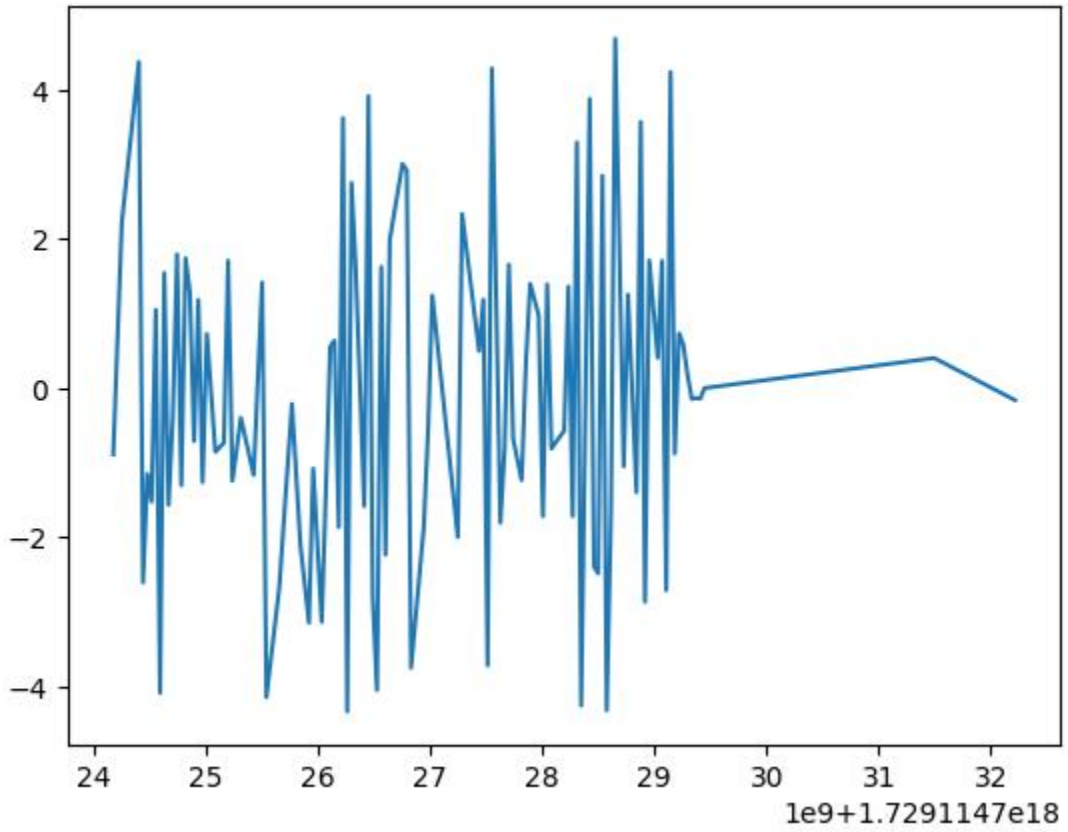
	time	seconds_elapsed	z	y	x	magnitude	active
0	1729114709796172300	0.052172	0.000000	0.000000	0.000000	0.000000	False
1	1729114709834106600	0.090107	-0.256169	0.037665	-0.009615	0.259102	False
2	1729114709872040700	0.128041	0.160883	-0.024792	0.024385	0.164598	False
3	1729114709909975000	0.165975	-0.163143	0.028428	-0.004211	0.165655	False
4	1729114709947909400	0.203909	0.098420	-0.015674	0.012540	0.100446	False
...	...	...	...	...	...	...	...
2973	1729114822575113700	112.831114	-0.000735	0.003801	-0.001142	0.004037	False
2974	1729114822613048000	112.869048	-0.003042	-0.002776	0.010107	0.010914	False
2975	1729114822650982400	112.906982	-0.002964	0.010719	0.002897	0.011493	False
2976	1729114822688916700	112.944917	1.575144	-0.176965	0.185672	1.595891	True
2977	1729114822726851000	112.982851	-0.666391	0.145894	0.161224	0.700968	False

2978 rows × 7 columns

Z Axis



This graph represents the one full movement:

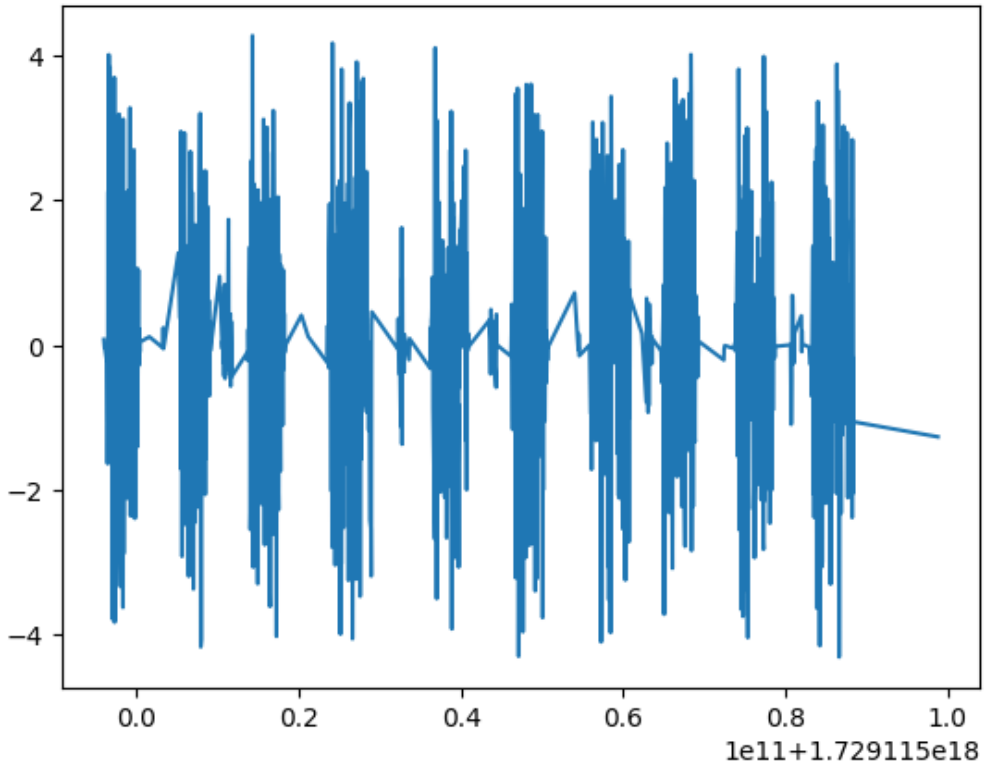


Cleaning Curve Fast Accelerometer

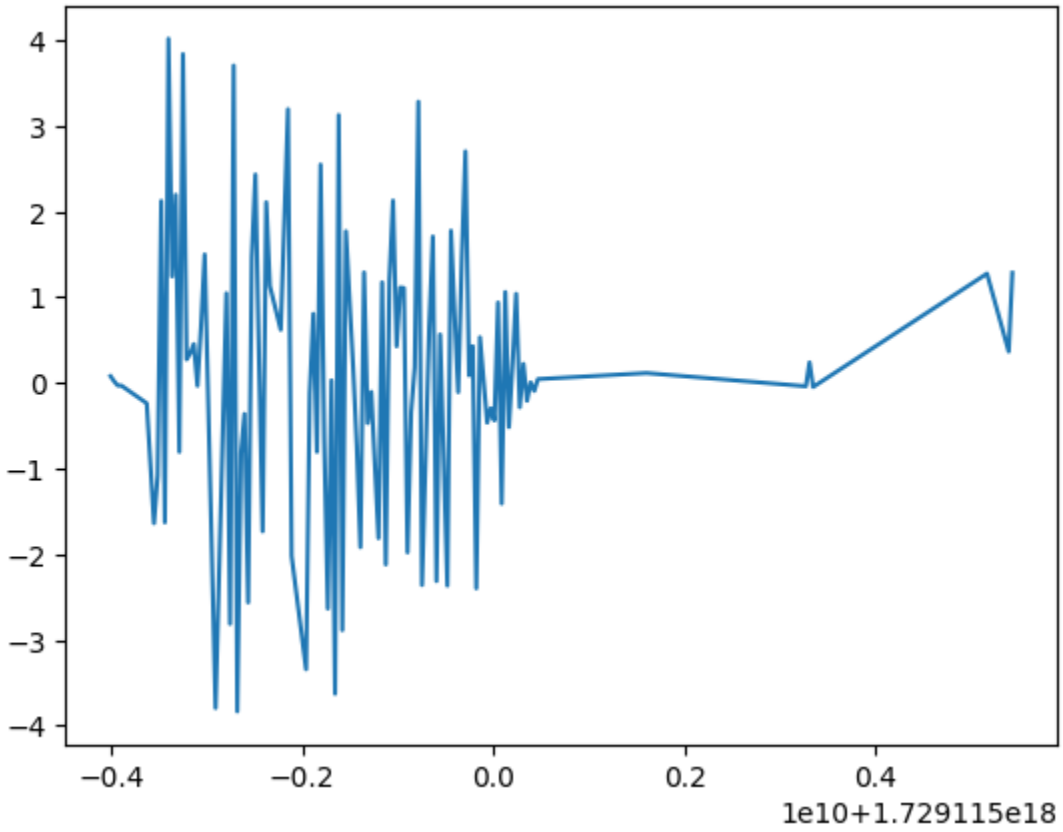
	time	seconds_elapsed	z	y	x	magnitude	active
0	1729114987950340400	0.028340	0.000000	0.000000	0.000000	0.000000	False
1	1729114987987895800	0.065896	0.001596	0.005874	-0.011253	0.012794	False
2	1729114988025830400	0.103830	0.044533	-0.006611	0.016885	0.048083	False
3	1729114988063764700	0.141765	-0.041039	-0.010302	-0.018762	0.046286	False
4	1729114988101699300	0.179699	0.039904	-0.030601	0.003223	0.050390	False
...	...	...	...	...	...	...	...
2920	1729115098718901500	110.796902	0.001064	-0.003592	-0.005816	0.006918	False
2921	1729115098756836000	110.834836	0.003590	-0.013272	-0.008080	0.015947	False
2922	1729115098794770400	110.872771	0.001181	0.001149	0.010580	0.010708	False
2923	1729115098832705000	110.910705	-1.269129	0.293492	-0.059588	1.303985	True
2924	1729115098870639600	110.948640	0.556180	-0.121369	-0.024277	0.569786	False

2925 rows × 7 columns

Z Axis



This graph represents the one full movement:



- **Data Flattening**

To flatten the data, we had to first separate data for each movement from one point to another. We set x amount of data should belong to one movement, and we were able to separate reach row of data this way as evidenced by the graphs shown below.

Flattening Straight Slow Data into Rows

	0	1	2	3	4	5	6	7	8	9	...	39	40	41	42	43	44	45	46	47	time
0	2.371576	2.122112	2.909207	2.896534	3.585578	3.562775	3.186043	3.018978	3.135702	4.709179	...	3.777659	4.879147	2.398383	3.105318	4.051899	2.096247	2.044812	2.530960	2.325784	4.514235
1	2.211903	2.085911	2.666916	2.343165	2.031213	2.204109	2.014863	2.078244	2.076389	2.082099	...	3.636804	2.356534	2.545400	2.239837	2.883761	3.001038	2.583434	2.014062	3.317880	10.356184
2	2.962727	2.035539	2.427897	3.053663	4.277031	4.884827	2.773665	3.098241	3.322927	2.629502	...	2.069469	2.261327	2.167470	2.449125	2.583810	2.581053	2.071538	2.408598	2.111740	9.976836
3	2.029873	2.960320	2.601397	2.766443	2.728132	2.006235	2.030639	2.203344	2.755221	2.347860	...	2.863282	2.218723	2.060986	2.023459	3.150228	2.192580	4.952481	2.025808	2.731471	11.645966
4	2.823050	2.034155	3.416577	2.283977	2.600332	2.550683	3.638719	3.265535	2.213674	2.490698	...	2.780290	2.140177	2.336947	2.031931	2.037253	2.515832	2.188437	2.252081	2.315338	9.900967
5	2.645902	2.202630	2.283878	2.231912	2.424722	2.562714	2.558824	2.324147	2.023054	2.238441	...	2.074128	3.621369	3.863420	2.285943	2.180671	2.162177	3.163396	2.069397	3.129875	9.483683
6	2.168176	2.444815	2.750544	2.242590	2.142738	3.772134	3.695413	3.922720	2.588071	2.322422	...	2.521286	2.767169	2.388169	2.214194	2.346595	2.436690	2.293517	2.366243	2.222404	6.638574
7	2.068471	2.826105	4.494189	3.062824	2.370137	2.559930	3.992951	3.089477	4.630917	2.481509	...	3.646228	2.242434	3.404361	2.387664	2.101436	2.225410	2.244414	3.550783	2.665907	6.031615
8	2.595993	2.280732	2.111242	2.047253	2.002173	2.142381	2.999113	2.091458	3.865743	2.605269	...	2.416793	2.482181	3.160907	3.562111	2.703941	4.241735	4.476408	4.636997	3.140430	9.483667
9	2.053901	2.524367	2.363959	2.016465	2.883691	3.332334	2.290435	2.059972	3.652162	3.051133	...	2.941197	3.257938	2.369738	2.442004	2.208651	2.483198	2.382410	2.325380	2.092158	9.711267

10 rows × 49 columns

Flattening Straight Fast Data into Rows

	0	1	2	3	4	5	6	7	8	9	...	35	36	37	38	39	40	41	42	43	time
0	2.715793	3.451921	2.847930	2.520625	2.295387	2.184215	2.212477	2.541820	3.727414	3.271509	...	2.722196	3.013193	2.261778	2.698033	3.300384	2.430632	2.002075	2.501843	2.004883	3.414077
1	2.349730	4.045139	2.017969	2.821946	2.929542	2.210115	2.811800	3.588242	2.222862	2.928176	...	2.024767	2.572843	2.205079	2.034230	2.333482	2.423693	2.323870	2.238367	2.324386	8.497260
2	2.798992	3.440837	2.271195	2.083719	4.399298	4.345245	2.481314	2.289419	2.957538	2.123119	...	2.916371	2.731337	2.058413	2.111066	2.169513	2.377412	2.298850	2.272302	2.030336	6.297076
3	2.449133	2.016092	3.902702	3.255774	2.947191	2.475685	2.631105	2.472141	2.006035	3.888945	...	3.202871	2.333157	3.011344	3.371351	2.750013	3.062585	2.716737	4.096598	2.643325	3.452011
4	2.328725	2.358303	2.189483	2.072514	2.254124	3.242123	2.849933	2.123119	2.792120	2.423267	...	3.139261	3.088831	2.575867	2.093811	2.502501	2.999478	2.163286	2.960040	3.017243	7.966179
5	2.637776	2.513295	2.134894	2.559395	2.022107	2.484892	2.435474	2.383720	2.405925	2.792618	...	3.291481	2.919464	3.199975	2.111689	2.419192	2.656623	2.316345	2.760768	3.078082	8.004111
6	2.257003	3.332752	2.014133	2.870286	3.839150	3.209215	2.575845	2.677402	2.263200	2.046029	...	3.434262	3.106005	2.162082	2.141244	2.884161	2.193285	2.790366	2.634635	2.142737	7.624767
7	2.310868	4.405917	3.498773	2.536722	2.217377	2.199716	2.582549	2.286486	2.441769	2.005275	...	2.241766	4.375402	2.427945	2.539033	2.810792	2.151366	3.275512	3.553868	2.253181	8.383450
8	2.930541	2.249209	2.628964	2.014306	2.175389	2.007261	2.197676	2.288425	2.289228	4.191482	...	2.050803	2.363993	3.569679	2.329145	3.986133	3.540307	4.051314	3.450003	2.201831	8.876594
9	3.586923	2.586111	2.281530	2.482879	2.372329	2.379500	3.529151	2.241529	2.806015	4.145238	...	3.064042	2.124452	4.144578	3.729103	3.074080	3.558168	2.719381	2.120660	2.698640	8.990393

10 rows × 45 columns

Flattening Curve Slow Data into Rows

	0	1	2	3	4	5	6	7	8	9	...	94	95	96	97	98	99	100	101	102	time
0	1.742149	2.250546	4.473449	2.757855	1.530576	2.478263	1.574833	4.440044	3.201811	2.041243	...	1.911624	2.309244	1.881595	2.176479	1.870181	1.694478	1.537589	1.617366	1.741895	8.042094
1	2.327219	1.846475	1.670599	4.269923	4.705505	3.191439	1.977099	2.739042	3.231184	3.339578	...	1.936396	3.763141	2.919715	3.196006	3.874643	2.127255	2.089808	2.021786	2.092321	7.511017
2	1.964121	1.790412	2.085649	1.694083	1.527661	2.039664	1.756165	1.509168	1.700939	1.534991	...	1.706918	2.019088	2.518099	2.059457	1.965532	1.921734	2.045268	1.621498	2.663930	12.025218
3	2.594402	2.208658	2.076134	2.493269	1.585359	3.280989	4.944858	2.016732	1.914693	2.182724	...	2.020645	1.580629	1.911453	2.372028	2.848579	1.813822	1.606995	2.252181	3.506575	10.356106
4	2.088739	2.935154	3.116831	2.281686	1.528577	2.458338	2.341353	1.854815	3.475939	1.746751	...	4.537835	1.927740	2.625963	2.481021	3.872049	4.874452	3.174485	2.100265	2.200769	9.331873
5	1.574156	1.988651	2.653880	2.504308	2.006297	1.759331	1.696649	2.750349	2.212959	2.557824	...	2.306130	1.659927	1.543225	2.322661	1.717913	1.870514	1.790029	4.848476	3.353515	8.345573
6	2.404828	1.769350	2.385149	3.993062	1.777328	3.757428	4.436377	3.326628	2.316716	2.606313	...	2.916913	2.824895	2.590325	2.308333	2.242285	1.986486	1.914897	2.124553	1.633840	4.855604
7	1.563565	2.059806	3.291723	2.982909	2.197968	1.637974	1.646945	1.770220	1.820618	1.847344	...	3.957118	2.939762	2.520168	2.110575	2.003295	2.319862	1.839612	1.910261	1.872235	7.624812
8	1.730797	1.532725	2.561157	3.190676	2.446366	2.477188	1.989493	2.779836	2.935106	2.269661	...	1.735405	1.706418	1.781868	1.605481	1.562872	1.718263	2.925080	1.861342	2.300557	13.049418
9	2.360549	2.936340	2.027521	2.120626	3.345866	2.478141	2.432153	2.037672	1.583724	2.140768	...	2.540565	2.325057	2.578920	2.362373	1.959983	1.788503	1.906917	2.028031	2.012269	8.042076

10 rows × 104 columns

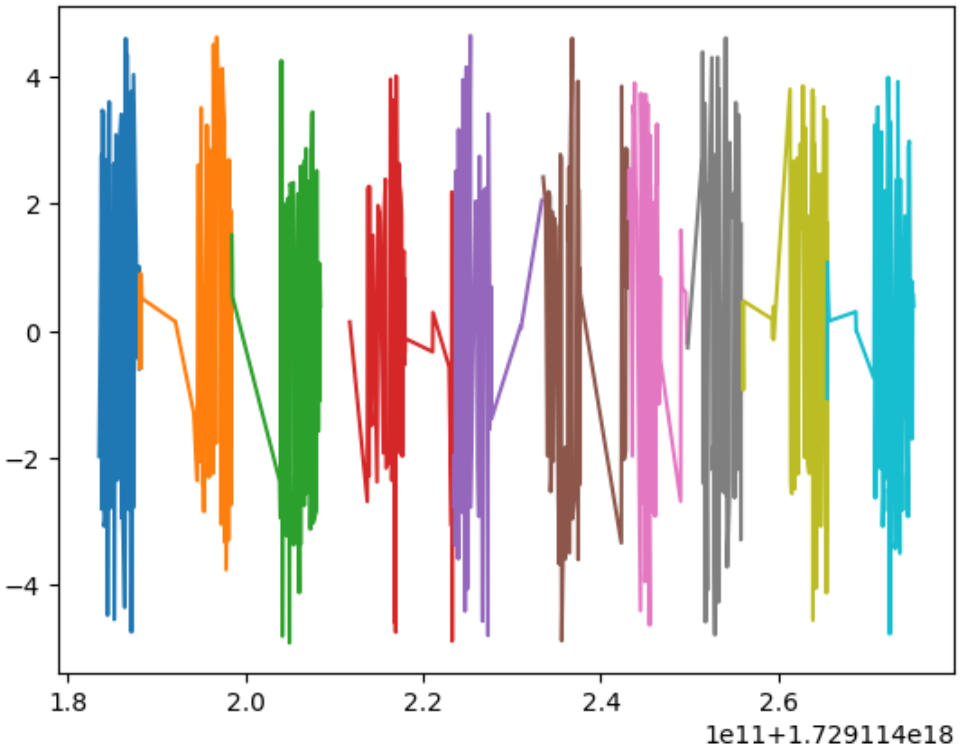
REPORT: ASSIGNMENT 4
COURSE: CPSC-5616EL-01: Machine Learning and Deep Learning
GROUP: 3

Flattening Curve Fast Data into Rows

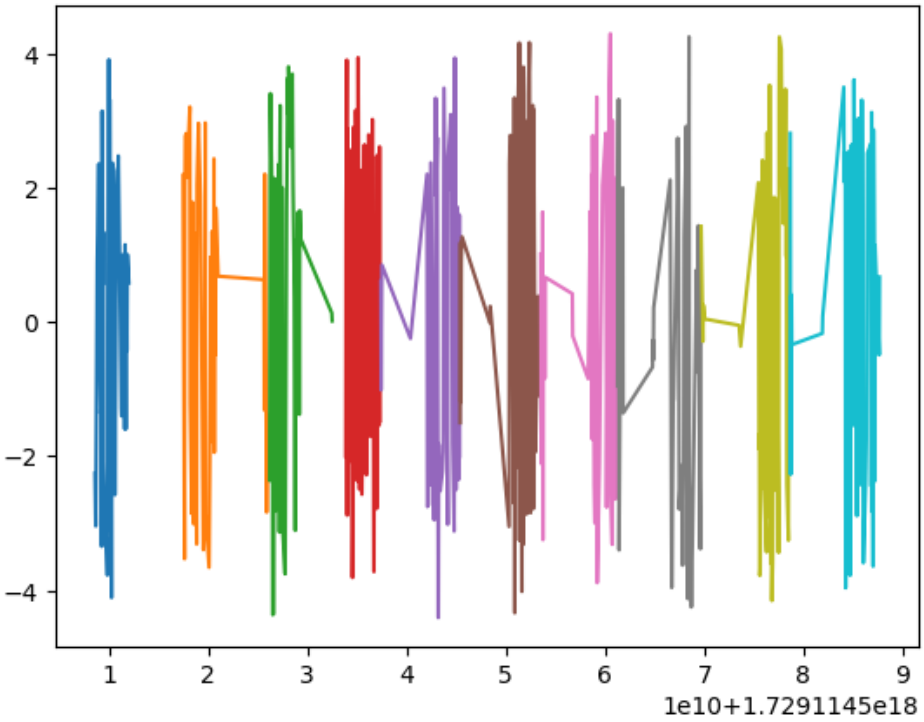
	0	1	2	3	4	5	6	7	8	9	...	96	97	98	99	100	101	102	103	104	time
0	1.220110	1.343753	1.354199	1.216459	1.304769	2.071664	1.442196	2.206189	1.988942	4.055076	...	1.656810	1.355869	1.415636	1.305623	2.528849	1.799735	1.327568	1.276882	1.581912	9.445710
1	3.051739	2.255097	1.998337	1.656062	3.288846	1.893564	1.851700	2.210736	3.079208	3.493309	...	1.349674	1.396661	1.626213	1.249138	1.285455	2.748619	1.324817	2.635682	3.654437	6.107469
2	3.239227	2.221109	1.731353	1.492881	1.341988	1.682938	1.800111	1.642308	1.821374	1.599375	...	2.076098	2.004019	2.389815	1.903246	2.252983	2.434065	1.732759	1.591875	1.904805	6.562687
3	1.887755	1.833757	1.963298	1.587033	1.262227	1.436754	1.437267	1.408782	1.370303	1.434380	...	2.350719	2.758191	3.513099	3.043766	2.812755	2.880048	2.678758	2.653626	1.967182	10.469953
4	1.985602	3.066473	3.986447	4.181601	3.766433	1.313260	1.401146	1.307262	1.458217	1.337971	...	2.560722	2.633999	3.485254	3.263770	3.971340	3.163800	3.118856	3.381946	2.108350	11.532120
5	1.225862	3.125379	1.648601	2.458467	2.129480	3.335846	4.007523	3.889955	2.381442	2.411256	...	2.018503	4.359888	3.736259	2.386863	3.258493	4.444154	2.177213	2.088615	2.917924	9.331910
6	3.438521	3.723679	4.114057	3.067999	2.466214	3.350672	2.473622	4.050294	2.839703	2.195408	...	1.883103	3.065772	3.678363	1.822582	3.410408	3.771820	3.834950	4.322636	3.267707	10.583747
7	2.964102	1.896989	2.299545	2.650292	2.270512	1.727640	1.797195	1.780051	3.531001	4.448920	...	3.446445	3.294972	2.580501	3.333736	1.731026	2.008566	2.113351	3.444547	3.789592	8.573209
8	2.536646	1.977105	1.963545	1.925460	1.996514	2.231821	2.045534	2.018144	1.679843	1.425702	...	2.932296	3.412209	3.395904	3.206670	4.439973	2.833359	3.218233	3.122814	2.805369	9.407761
9	3.323856	1.812396	2.134920	1.778407	1.842351	2.066995	1.787191	1.597515	1.260286	1.453516	...	1.554557	3.821898	3.584435	2.700145	4.353761	2.219982	4.108464	3.085589	3.387396	9.293952

10 rows × 106 columns

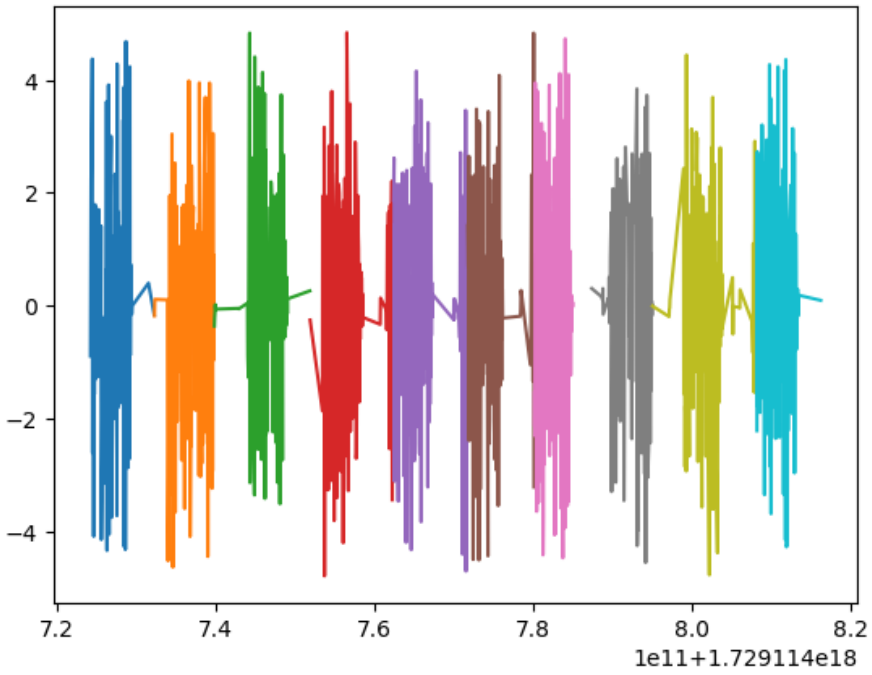
Visualizing each movement in the Straight Slow graph



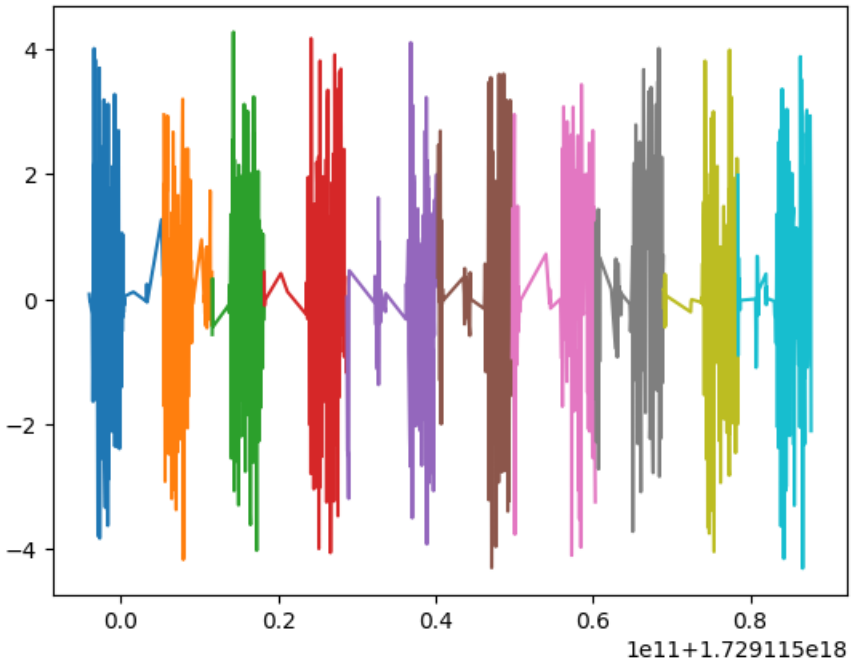
Visualizing each movement in the Straight Fast graph



Visualizing each movement in the Curve Slow graph



Visualizing each movement in the Curve Fast graph



Feature Engineering

We used feature engineering to created time and distance data that was computed by taking the maximum and minimum for each movement from start to finish and subtracting the time elapsed.

Straight Slow

46	47	time	distance
2.530960	2.325784	4.514235	4.884643
2.014062	3.317880	10.356184	4.196215
2.408598	2.111740	9.976836	4.737690
2.025808	2.731471	11.645966	4.218465
2.252081	2.315338	9.900967	4.679288
2.069397	3.129875	9.483683	4.277348
2.366243	2.222404	6.638574	4.762841
3.550783	2.665907	6.031615	4.976420
4.636997	3.140430	9.483667	4.309653
2.325380	2.092158	9.711267	4.446066

Straight Fast

42	43	time	distance
2.501843	2.004883	3.414077	4.674456
3.238367	2.324386	8.497260	4.683516
2.272302	2.030336	6.297076	4.914314
4.096598	2.643325	3.452011	4.652506
2.960040	3.017243	7.966179	4.471981
2.760768	3.078082	8.004111	4.895914
2.634635	2.142737	7.624767	4.445302
3.553868	2.253181	8.383450	4.466935
3.450003	2.201831	8.876594	4.627185
2.120660	2.698640	8.990393	4.679034

Curve Slow

101	102	time	distance
1.617366	1.741895	8.042094	5.494333
2.021786	2.092321	7.511017	5.480213
1.621498	2.663930	12.025218	4.759760
2.252181	3.506575	10.356106	5.264345
2.100265	2.200769	9.331873	5.439272
4.848476	3.353515	8.345573	5.420741
2.124553	1.633840	4.855604	5.882628
1.910261	1.872235	7.624812	4.897848
1.861342	2.300557	13.049418	4.982348
2.028031	2.012269	8.042076	5.606347

Curve Fast

103	104	time	distance
1.276882	1.581912	9.445710	5.098008
2.635682	3.654437	6.107469	5.355785
1.591875	1.904805	6.562687	4.932158
2.653626	1.967182	10.469953	5.323754
3.381946	2.108350	11.532120	5.352571
2.088615	2.917924	9.331910	4.995910
4.322636	3.267707	10.583747	5.415371
3.444547	3.789592	8.573209	4.717542
3.122814	2.805369	9.407761	5.362686
3.085589	3.387396	9.293952	4.956392

We are building a regression model and our target will be the distance. We used numerical methods to compute the distance. We took the accelerometer data and first integrated it, this gives us the velocity data. Then we integrated it again and this gives us the distance data.

	0	1	2	3	4	5	6	7	8	9	...	97	98	99	100	101	102	103	104	time	distance
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	4.879147	2.398383	3.105318	4.051899	2.096247	2.044812	2.530960	2.325784	4.514235	4.884643
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.356534	2.545400	2.239837	2.883761	3.001038	2.583434	2.014062	3.317880	10.356184	4.196215
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.261327	2.167470	2.449125	2.583810	2.581053	2.071538	2.408598	2.111740	9.976836	4.737690
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.218723	2.060986	2.023459	3.150228	2.192580	4.952481	2.025808	2.731471	11.645966	4.218465
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.140177	2.336947	2.031931	2.037253	2.515832	2.188437	2.252081	2.315338	9.900967	4.679288
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	3.621369	3.863420	2.285943	2.180671	2.162177	3.163396	2.069397	3.129875	9.483683	4.277348
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.767169	2.388169	2.214194	2.346595	2.436690	2.293517	2.366243	2.222404	6.638574	4.762841
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.242434	3.404361	2.387664	2.101436	2.225410	2.244414	3.550783	2.665907	6.031615	4.976420
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.482181	3.160907	3.562111	2.703941	4.241735	4.476408	4.636997	3.140430	9.483667	4.309653
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	3.257938	2.369738	2.442004	2.208651	2.483198	2.382410	2.325380	2.092158	9.711267	4.446066
10	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	3.013193	2.261778	2.698033	3.300384	2.430632	2.002075	2.501843	2.004883	3.414077	4.674456
11	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.572843	2.205079	2.034230	2.333482	2.423693	2.323870	3.238367	2.324386	8.497260	4.683516
12	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.731337	2.058413	2.111066	2.169513	2.377412	2.298850	2.272302	2.030336	6.297076	4.914314
13	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.333157	3.011344	3.371351	2.750013	3.062585	2.716737	4.096598	2.643325	3.452011	4.652506
14	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	3.088831	2.575867	2.093811	2.502501	2.999478	2.163286	2.960040	3.017243	7.966179	4.471981
15	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.919464	3.199975	2.111689	2.419192	2.656623	2.316345	2.760768	3.078082	8.004111	4.895914
16	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	3.106005	2.162082	2.141244	2.884161	2.193285	2.790366	2.634635	2.142737	7.624767	4.445302
17	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	4.375402	2.427945	2.539033	2.810792	2.151366	3.275512	3.553868	2.253181	8.383450	4.466935
18	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.363993	3.569679	2.329145	3.986133	3.540307	4.051314	3.450003	2.201831	8.876594	4.627185
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	2.124452	4.144578	3.729103	3.074080	3.558168	2.719381	2.120660	2.698640	8.990393	4.679034
20	0.000000	0.000000	1.742149	2.250546	4.473449	2.757855	1.530576	2.478263	1.574833	4.440044	...	2.309244	1.881595	2.176479	1.870181	1.694478	1.537589	1.617366	1.741895	8.042094	5.494333
21	0.000000	0.000000	2.327219	1.846475	1.670599	4.269923	4.705505	3.191439	1.977099	2.739042	...	3.763141	2.919715	3.196006	3.874643	2.127255	2.089808	2.021786	2.092321	7.511017	5.480213
22	0.000000	0.000000	1.964121	1.790412	2.085649	1.694083	1.527661	2.039664	1.756165	1.509168	...	2.019088	2.518099	2.059457	1.965532	1.921734	2.045268	1.621498	2.663930	12.025218	4.759760
23	0.000000	0.000000	2.594402	2.208658	2.076134	2.493269	1.585359	3.280989	4.944858	2.016732	...	1.580629	1.911453	2.372028	2.848579	1.813822	1.606995	2.252181	3.506575	10.356106	5.264345
24	0.000000	0.000000	2.088739	2.935154	3.116831	2.281686	1.528577	2.458338	2.341353	1.854815	...	1.927740	2.625963	2.481021	3.872049	4.874452	3.174485	2.100265	2.200769	9.331873	5.439272
25	0.000000	0.000000	1.574156	1.988651	2.653880	2.504308	2.006297	1.759331	1.696649	2.750349	...	1.659927	1.543225	2.322661	1.717913	1.870514	1.790029	4.848476	3.353515	8.345573	5.420741
26	0.000000	0.000000	2.404828	1.769350	2.385149	3.993062	1.777328	3.757428	4.436377	3.326628	...	2.824895	2.590325	2.308333	2.242285	1.986486	1.914897	2.124553	1.633840	4.855604	5.882628
27	0.000000	0.000000	1.563565	2.059806	3.291723	2.982909	2.197968	1.637974	1.646945	1.770220	...	2.939762	2.520168	2.110575	2.003295	2.319862	1.839612	1.910261	1.872235	7.624812	4.897848
28	0.000000	0.000000	1.730797	1.532725	2.561157	3.190676	2.446366	2.477188	1.989493	2.779836	...	1.706418	1.781868	1.605481	1.562872	1.718263	2.925080	1.861342	2.300557	13.049418	4.982348
29	0.000000	0.000000	2.360549	2.936340	2.027521	2.120626	3.345866	2.478141	2.432153	2.037672	...	2.325057	2.578920	2.362373	1.959983	1.788503	1.906917	2.028031	2.012269	8.042076	5.606347
30	1.220110	1.343753	1.354199	1.216459	1.304769	2.071664	1.442196	2.206189	1.988942	4.055076	...	1.355869	1.415636	1.305623	2.528849	1.799735	1.327568	1.276882	1.581912	9.445710	5.098008
31	3.051739	2.255097	1.998337	1.656062	3.288846	1.893564	1.851700	2.210736	3.079208	3.493309	...	1.396661	1.626213	1.249138	1.285455	2.748619	1.324817	2.635682	3.654437	6.107469	5.355785
32	3.239227	2.221109	1.731353	1.492881	1.341988	1.682938	1.800111	1.642308	1.821374	1.599375	...	2.004019	2.389815	1.903246	2.252983	2.434065	1.732759	1.591875	1.904805	6.562687	4.932158
33	1.887755	1.833757	1.963298	1.587033	1.262227	1.436754	1.437267	1.408782	1.370303	1.434380	...	2.758191	3.513099	3.043766	2.812755	2.880048	2.678758	2.653626	1.967182	10.469953	5.323754
34	1.985602	3.066473	3.986447	4.181601	3.766433	1.313260	1.401146	1.307262	1.458217	1.337971	...	2.633999	3.485254	3.263770	3.971340	3.163800	3.118856	3.381946	2.108350	11.532120	5.352571
35	1.225862	3.125379	1.648601	2.458467	2.129480	3.335846	4.007523	3.889955	2.381442	2.411256	...	4.359888	3.736259	2.386863	3.258493	4.444154	2.177213	2.088615	2.917924	9.331910	4.995910
36	3.438521	3.723679	4.114057	3.067999	2.466214	3.350672	2.473622	4.050294	2.839703	2.195408	...	3.065772	3.678363	1.822582	3.410408	3.771820	3.834950	4.322636	3.267707	10.583747	5.415371
37	2.964102	1.896989	2.299545	2.650292	2.270512	1.727640	1.797195	1.780051	3.531001	4.448920	...	3.294972	2.580501	3.333736	1.731026	2.008566	2.113351	3.444547	3.789592	8.573209	4.717542
38	2.536646	1.977105	1.963545	1.925460	1.996514	2.231821	2.045534	2.018144	1.679843	1.425702	...	3.412209	3.395904	3.206670	4.439973	2.833359	3.218233	3.122814	2.805369	9.407761	5.362686
39	3.323856	1.812396	2.134920	1.778407	1.842351	2.066995	1.787191	1.597515	1.260286	1.453516	...	3.821898	3.584435	2.700145	4.353761	2.219982	4.108464	3.085589	3.387396	9.293952	4.956392

Preparing the data for training

The flattened data straight slow, straight fast, curve slow and curve fast have different shapes, so we used pre-padding to make them equal shapes for merging the four datasets into one single dataset**.** Then we split the data into training and testing by a 80-20 split respectively. We used pre-padding which added 0s at the start of the rows to make all of the data have the same width.

3. Results and Discussion

The Models

Random Forest Regression (RFR) predicts values by combining multiple decision trees. Each tree makes a prediction, and the final result is the average of all trees' predictions. This approach reduces errors and improves accuracy.

Support Vector Regression (SVR) predicts values by finding a line (or curve) that best fits the data within a margin of error. It aims to keep most predictions within a certain distance from this line, minimizing error while balancing complexity, which makes it effective for handling noise and outliers in data.

1D Convolutional Neural Network (CNN1D) is a variation of CNN that processes data in either a sequential or time-series format rather than the 2D grid structure, like images. In 1D CNN, the convolution filters move only along one dimension of the data. It will, therefore, have the capability of learning such local patterns in sequences-such as a trend or dependencies between adjacent points. These are applied to tasks such as signal processing, natural language processing, audio analysis, and sensor data modeling. Key layers in a 1D CNN include convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification or regression tasks. They perform very well when the sequence of data is important.

Gated Recurrent Unit (GRU) is a form of recurrent neural network that is designed to deal with sequence data and to capture the long-term dependencies. It contains two gates: a reset gate to forget the irrelevant past information and an update gate to determine how much new information to retain. GRUs are less complex and faster than LSTMs, yet they still effectively handle the problem of vanishing gradients, hence useful in time-series forecasting and natural language processing.

CNN-GRU

HybridCNN_GRU is a neural network, which combines the convolutional and recurrent layers for the processing of sequential data. This model consists of a 1D convolutional layer, which captures the local spatial features from input sequences. Each convolutional layer is followed by a ReLU activation function and a MaxPool1d for downsampling. The pre-processed features are fed into the Gated Recurrent Unit network to learn the temporal dependencies in the sequence data. This reduces the output of the GRU at the last time step and feeds into a fully connected layer, which generates a single scalar prediction. It initializes convolutional and GRU layers dynamically, hence depending on the size of features fed to it, these can be initialized accordingly. More importantly, this model is in a hybrid design, applying CNN to capture the patterns in a localized manner, while the GRU tries to model sequential dependencies from them. This is one of the best models available for such applications as Time Series Analysis, Speech Processing, and Sensor Data Modeling.

Results for Random Forest Regression

This regression model results include three important performance metrics, which are RMSE, MAE, and the coefficient of determination. Here is how each explains the performance of your model:

Root Mean Squared Error (RMSE) = 0.3501

The RMSE calculates the standard deviation of the residuals, which are the prediction errors. Residuals measure how far from the regression line the individual data values are; the RMSE is a kind of measure regarding how spread out those residuals are. In other words, we can learn from it how diffuse the data is around the line of best fit. The lower the value of the RMSE, the better the fit. An RMSE of 0.3501 would give the interpretation that our model's prediction, on average, is 0.3501 units away from its actual values.

Mean Absolute Error (MAE) = 0.2858

MAE gives the average magnitude of errors in a set of predictions without considering the direction. It's the average over the test sample of the absolute differences between prediction and actual observation, with equal weight given to all individual differences. If it is 0.2858, the implication is that your average prediction error is 0.2858 units, which informs you about the accuracy of your model in the units of your target variable.

R^2 (Coefficient of Determination) = 0.8983

R^2 is the statistical measure indicating the proportion of variance of a dependent variable by an independent variable or variables explained within a regression model. It provides an indication of the goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model. R^2 of 0.8983 means that about 89.83% of the observed variation is accounted for by the regressors of this model. Generally speaking, it is a high value and this suggests that your model fits well with the data.

Overall, based on these metrics, your model does quite well, supported by a very strong explanatory power of the high R^2 value. By the same token, the relatively low values of RMSE and MAE do certainly indicate good predictive accuracy and reliability.

RMSE	**0.3501134778625813**
MAE	**0.28581227703502976**
R2	**0.8983421031735459**

Results For SVM

Here is the interpretation of performance metrics for our Support Vector Regression or SVR model:

Root Mean Squared Error (RMSE) = 0.3951

This metric means that, on average, the model's predictions are 0.3951 units away from the actual data points. Where RMSE is a little higher than our best Random Forest Regression model so far; this means it is somewhat less precise in the forecast of the data.

Mean Absolute Error (MAE) = 0.3013

MAE informs us about the average magnitude of an error on the same scale as that of our target variable. Each error value is given an equal weight. If MAE is 0.3013, it means our model is off from the actual value by about 0.3013 units. This MAE is higher than it was for the Random Forest model, but still pretty decent.

R^2 (Coefficient of Determination) = 0.8706

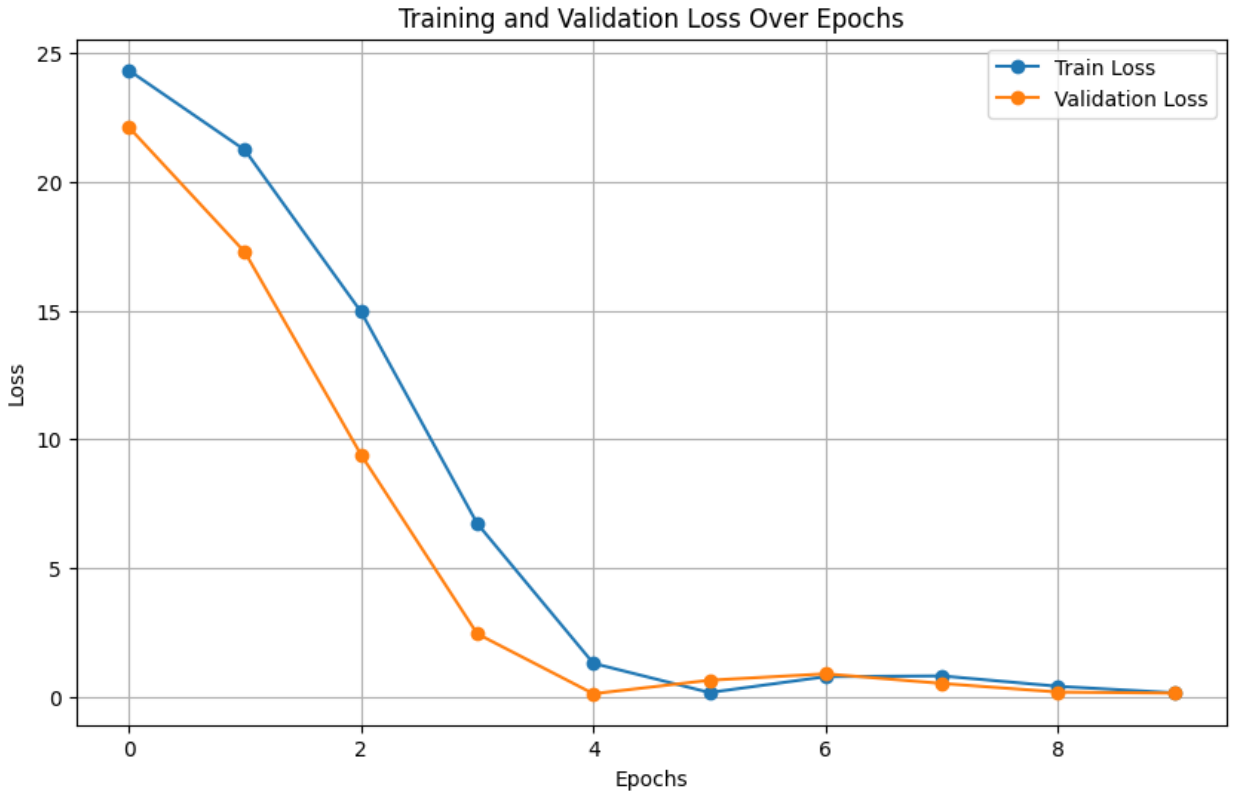
The R^2 value of 0.8706 infers that about 87.06% of the variation in the dependent variable is explained by the independent variables of the model. It's a high value but a bit lower than the one obtained for the Random Forest model, whose R^2 value was 0.8983.

RMSE	**0.39507935400452143**
MAE	**0.3012801839362118**
R2	**0.8705529578500941**

Result From Hybrid CNN-GRU Model

*Mean Squared Error (*MSE)

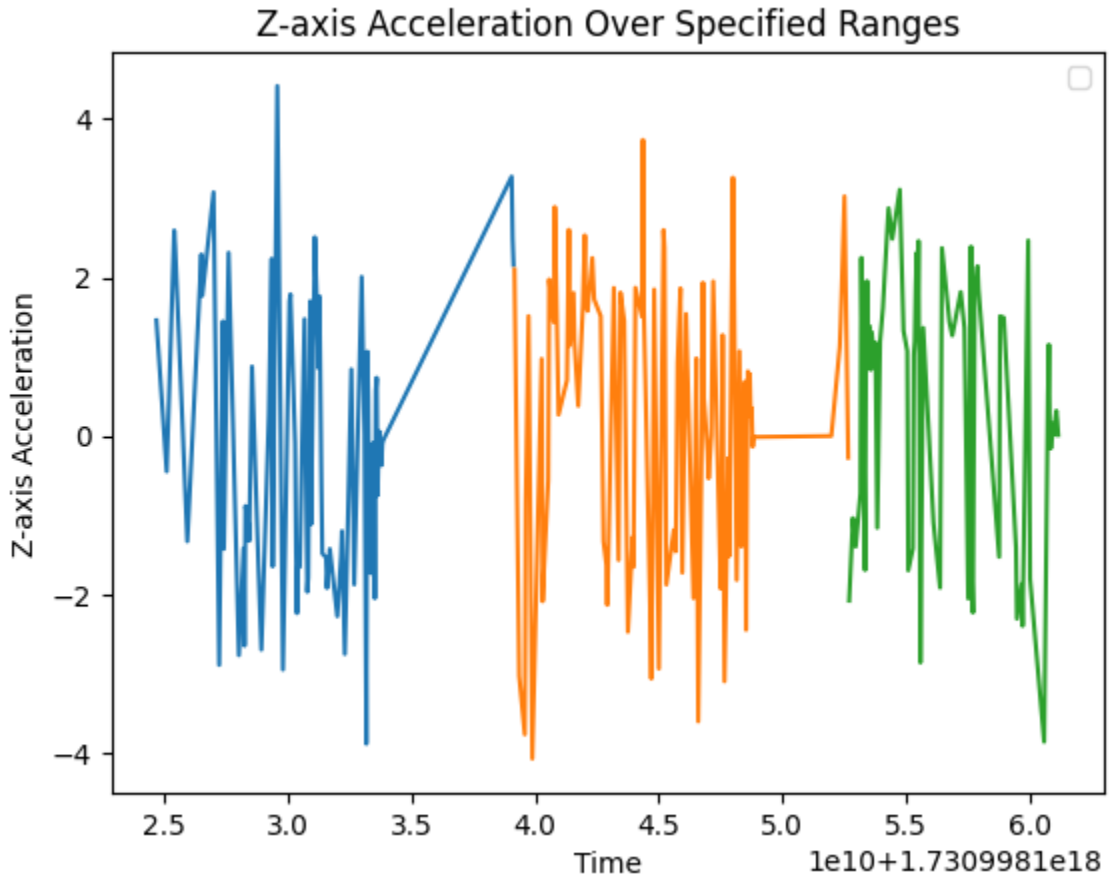
It had a training loss of 0.1570, while its validation loss is 0.1453—following that, this model performs modestly better on new and unseen data. The overall MSE Loss was higher than our Random Forest's best model—so this model makes the predictions less efficiently.



This graph shows how a model, while learning over time, decreases its error for both training data in blue and new, unseen data in orange. Initially, it drops quickly, but by the fourth time it sees the data, it doesn't improve much after that. When both lines come closer and flatten, that is a good indication that the model has learned well and isn't just memorizing the data.

Results from the new data

The new dataset provided was used for testing all the models for validating performance on unknown data. The data contained a mix of straight and curved accelerometer movement data which was cleaned and transformed the same way as the main dataset used for training the models.



Results From CNN-GRU: `Evaluation Loss on New Data: 5.3383`

- Evaluation Loss: 5.3383. The lower, the less on average will be the difference in values between what it had predicted versus the real result from that data. This gives some indication regarding predictive accuracy when encountering data never seen before.

Results From Random Forest:

```
RMSE 2.1908632874040395  MAE 1.5995380790642002  R2 -0.3887164333154096
```

RMSE: 2.1909. This measures the mean magnitude of the error for a unit like data. A lower value is better; with that, it means a more accurate model.
MAE, or Mean Absolute Error: 1.5995. It also shows the average error but in a way that it does not consider the direction, whether plus or minus. Similar to RMSE, the smaller the value of MAE, the higher is the accuracy.
R2: -0.3887. It always measures the goodness of fit for the data to a statistical model. The negative R2 indicates that the model does not fit very well to the data.

Results From SVR:

```
RMSE 2.281100240449355  MAE 1.6134487667750426  R2 -0.5054688003743055
```

RMSE: 2.2811. Much alike the Random Forest, only a little higher, this may indicate it is not quite as accurate.
MAE: 1.6134. Also slightly higher than the Random Forest, suggesting less accuracy.
R2 = -0.5055. This is worse than the R2 for the Random Forest, which indicates an even worse fit to the data.

Comparing the ML and DL models:

Performance with Old Data:

When the models are trained and validated with old data, it showed a varying degree of accuracy. RFR showed some promise in this regard with an RMSE value of 0.3501, MAE of 0.2858, and R^2 of 0.8983, reflecting that this model has satisfactorily captured the variation in the training data. Also, the SVR model did well with an RMSE of 0.3951, MAE of 0.3013, and R^2 of 0.8706. The Hybrid CNN-GRU, as a model tailored for sequential data, had a relatively low training loss of 0.1570 and validation loss of 0.1453, showing good fit and generalization on the old data.

New Data Performance:

All models showed a big decrease in performance when evaluated on new test data. The RMSE was higher in the Random Forest model, going as high as 2.1909, the MAE also went up to 1.5995, and the R^2 drastically decreased to -0.3887, which is considered a very bad fit. On the SVR model, it was equally terrible; the RMSE of 2.2811 and MAE of 1.6134, while R^2 was even lower, at -0.5055. Whereas the Hybrid CNN-GRU model was specifically designed to run on sequential data, the results obtained on new data were highly contrasting; its evaluation loss of 5.3383 stressed great difficulties to generalize outside the standard dataset which was split into train and val splits. Further confirmation that all these models, though powerful, had been able to learn how to understand and estimate the behavior of well-known patterns within their training datasets—but not easily hold performance to that same high level for those patterns when the data was unfamiliar—was at best suspect for issues regarding model overfitting. The increase in the Loss for all the models on the new raw data could be as a result of how the data was collected which could indicate a difference in the levels of noise, speed and movement pattern of the robot in the main data and the new given test data as when the main data was split into testing and training splits, the models performed well on the test data.

Conclusion

The most important result of this project is that it considerably expanded our knowledge about robotic distance estimation based on IMU data. We have established a broad framework that integrates classic machine learning models with advanced deep learning approaches, effectively combining Random Forest Regression-RFR and Support Vector Regression-SVR with a hybrid CNN-GRU model. Such a progression from traditional models to a complex neural network has enabled us to dissect intricate temporal patterns in sensor data more proficiently. Though the models were promising regarding accuracy on known data, their performance on new, unseen datasets was indicative of challenges in generalization, thus pointing to the need for further refinement in model training and data preprocessing.These results are helpful in demonstrating the potential of hybrid models in robotic systems but also in pointing out the necessary methodologies that must be adopted to avoid overfitting and enhance model adaptability. This work contributes both to the development of robotic navigation systems and sets a base in this area for further studies that will improve the reliability and accuracy of autonomous robotic tasks.