ML Project

User Verification based on Mouse Dynamics

Aaditya Gupta - 2020552
Harjeet Singh Yadav - 2020561
Rohan Kulkarni - 2020537
Hemang Dahiya - 2020435

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Motivation

The modern world is extrapolating technology to a certain level that is exploitative, and the shady elements of the cyber world are trying to encroach on customers' privacy by the minute. The surge of cyber fraud and theft of login details to access customers’ financing accounts is appalling. In the last year almost 30,000 websites were hacked daily and according to a research cyberattacks increased 50% year-over-year.

Motivation

According to Cornell University's research, there is a unique way in which every person uses their mouse pointer while visiting any website. Taking inspiration from this, we are developing and training our model on the cursor movements data of registered users. Then, by leveraging the classification process of our model, we would be able to identify fraudulent user activity and hence, prevent that particular user from performing any similar activity in future.

Literature Survey

- **Intrusion Detection Using Mouse Dynamics** by Margit Antal, Elod Egyed-Zsigmond talks about this exact problem and proposes an intrusion detection model using a variety of features like ***'elapsed time', 'distance traveled', average velocity in x', 'average velocity in y'*** etc. They have also published a collection of datasets namely "Balabit-Chaoshen-DFL" datasets, and we are using the same for our project.

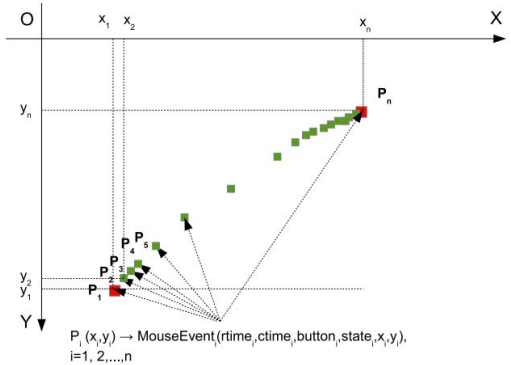


Fig. 1: A mouse action: a sequence of consecutive mouse events which describes the movement of the mouse between two screen positions.

Literature Survey

- **CLUSTERING WEB USERS BY MOUSE MOVEMENT TO DETECT BOTS AND BOTNET ATTACKS** by Justin Morgan: The objective of this project is to present an unsupervised approach for website administrators to detect web bots. The thesis presents an approach to cluster users and then tag them as being actual registered users, or as bots. K means clustering was used to cluster malicious traffic flows.

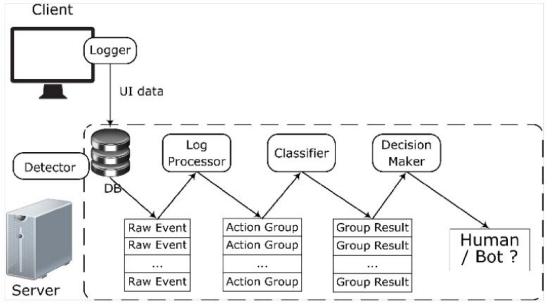
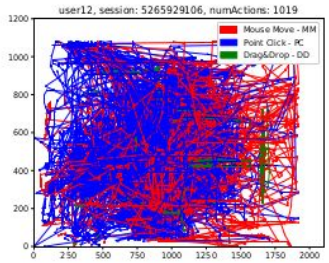
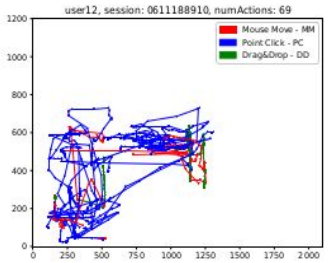


Figure 2.6: Architecture of the client-side Logger and server-side Detector
The client-side Logger and server-side Detector setup above, as shown in [8], closely resemble that of this thesis work's system design.

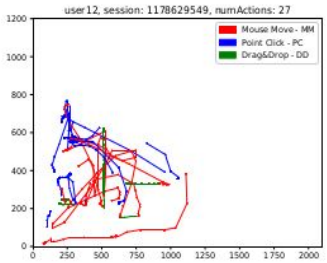
Dataset Description



(a) training session 5265929106



(b) test session 0611188910



(c) test session 1178629549

We have used :-

- The Balabit Mouse Challenge data set (Link)
- Chao Shen's Mouse Dynamics data set (Link)
- DFL Mouse Dynamics data set (Link)

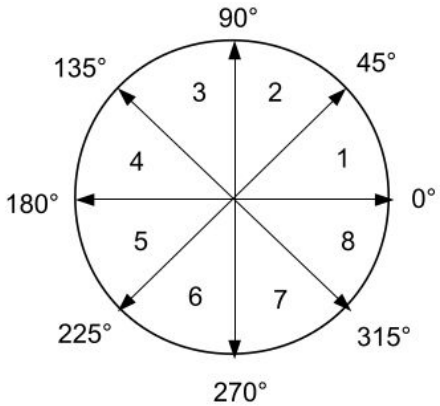
Different Attributes

The data-set consists of 39 attributes:-

- The type of action (MM: Mouse movement, CC: Point click, DD: Drag & Drop)
- Traveled Distance, (total distance by mouse)
- Curvature (average, std, min, max curvature)
- Velocity (average, std, min, max velocity of the mouse)
- Omega/Angular velocity (average, std, min, max omega)
- Deviation (largest deviation while using the mouse)
- Jerk (average, std, min, max jerk)
- Sum of angles (summation of angles of all the trajectories)

Different Attributes

- number of points, which means the number of mouse events contained in an action
- a_beg_time, which is acceleration time at the beginning of a mouse movement
- Directions (dir of end to end line), there are 8 directions defined as follows:-



Different Attributes

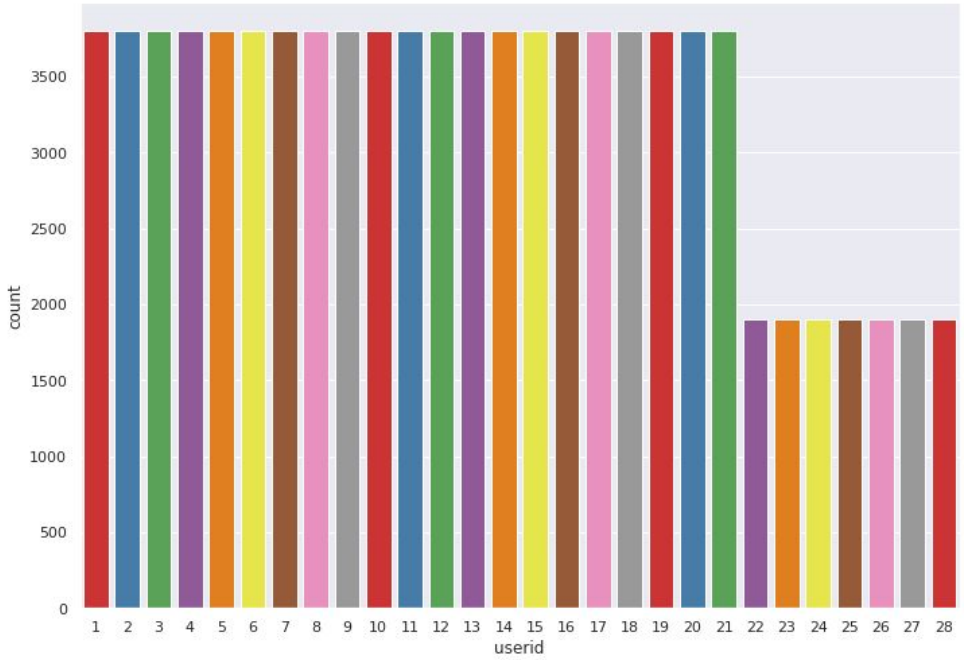
Name	Description	#features
v_x	mean, std, min, max	4
v_y	mean, std, min, max	4
v	mean, std, min, max	4
a	mean, std, min, max	4
j	mean, std, min, max	4
ω	mean, std, min, max	4
c	mean, std, min, max	4
type	MM, PC, DD	1
elapsed time	$t_n - t_1$	1
trajectory length	s_n	1
dist_end_to_end	$\|P_1P_n\|$	1
direction	see Fig 6	1
straightness	$\frac{\|P_1P_n\|}{s_n}$	1
num_points	n	1
sum_of_angles	$\sum_{i=1}^{n} \Theta_i$	1
largest_deviation	$max_i\{d(P_i, \|P_1P_n\|)\}$	1
sharp angles	$\#\{\Theta_i \| \Theta_i < TH\}$	1
a_beg_time	accel. time at the beginning	1
Total		39

Detailed description of features

Vx = velocity in x-direction
Vy = velocity in y-direction
V = linear velocity
a = acceleration
j = jerk
w = angular velocity

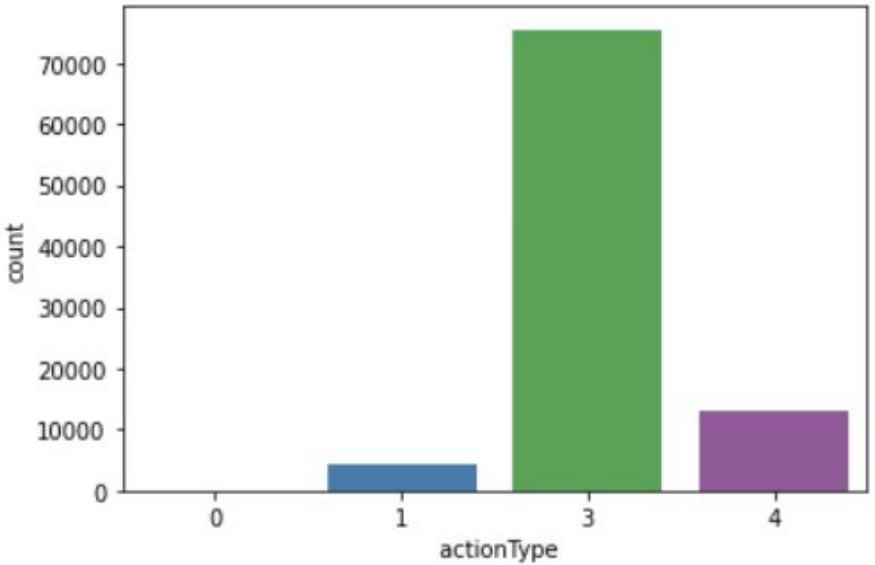
std: standard deviation

Visualisation



The graph shown here depicts the number of data points we have in our dataset corresponding to each user.

Visualisation

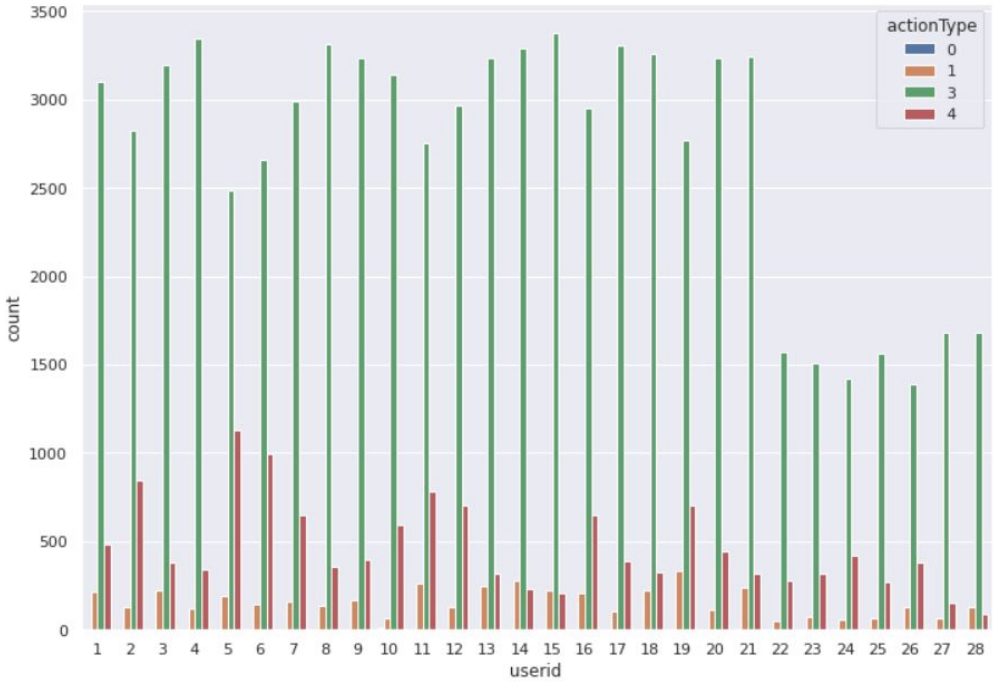


The graph here depicts the number of data points we have for each action type :

(i.e MM, PC, DD)

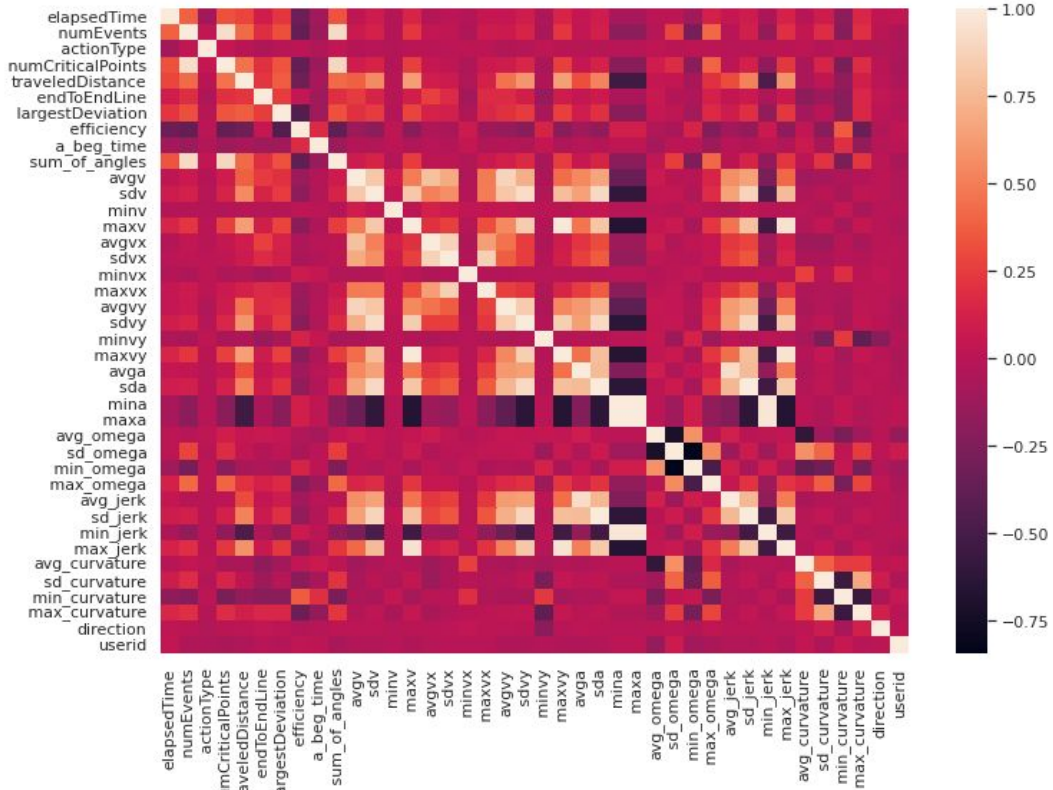
(1: MM, 3: PC, 4: DD)

Visualisation



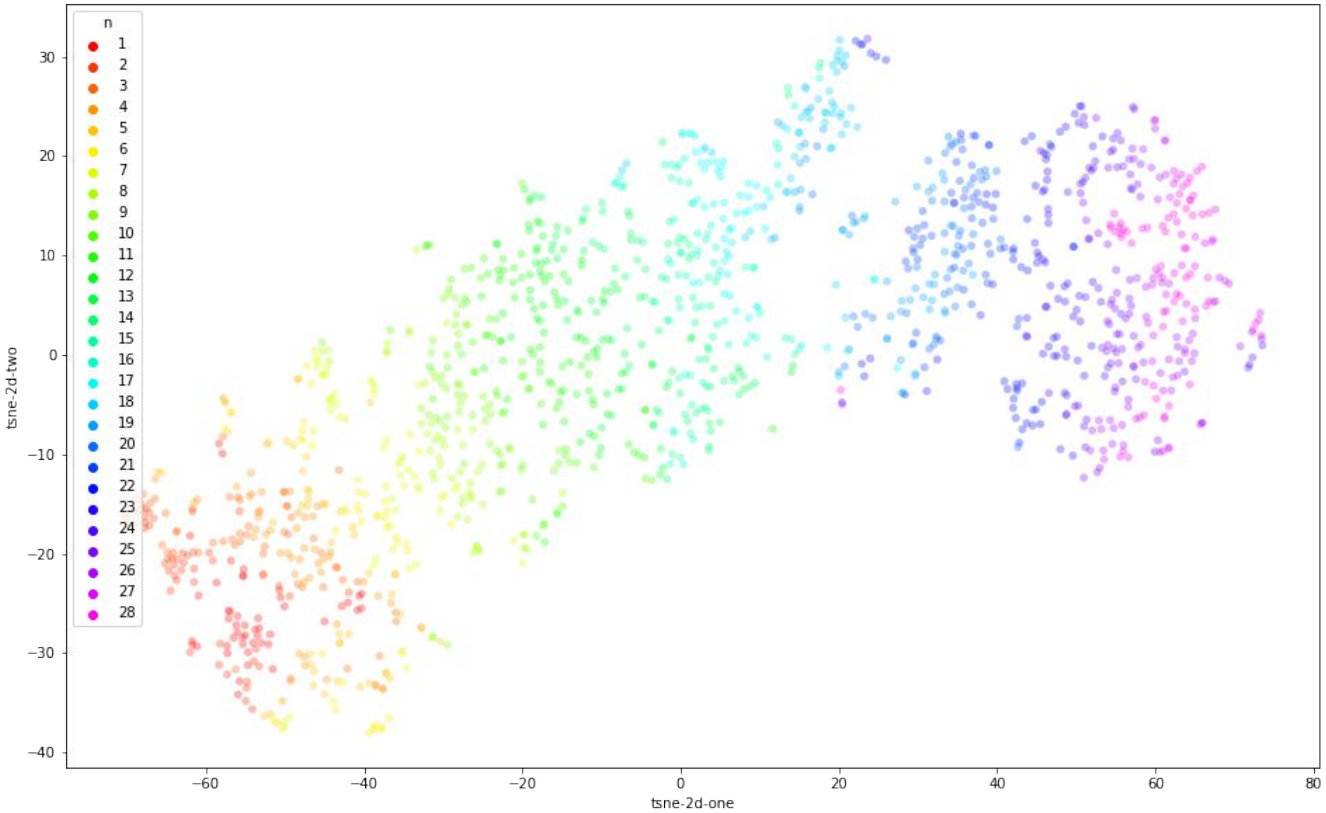
The number of times each user performed each action type.

Visualisation



This heat-map shows the correlation between the features in the dataset.

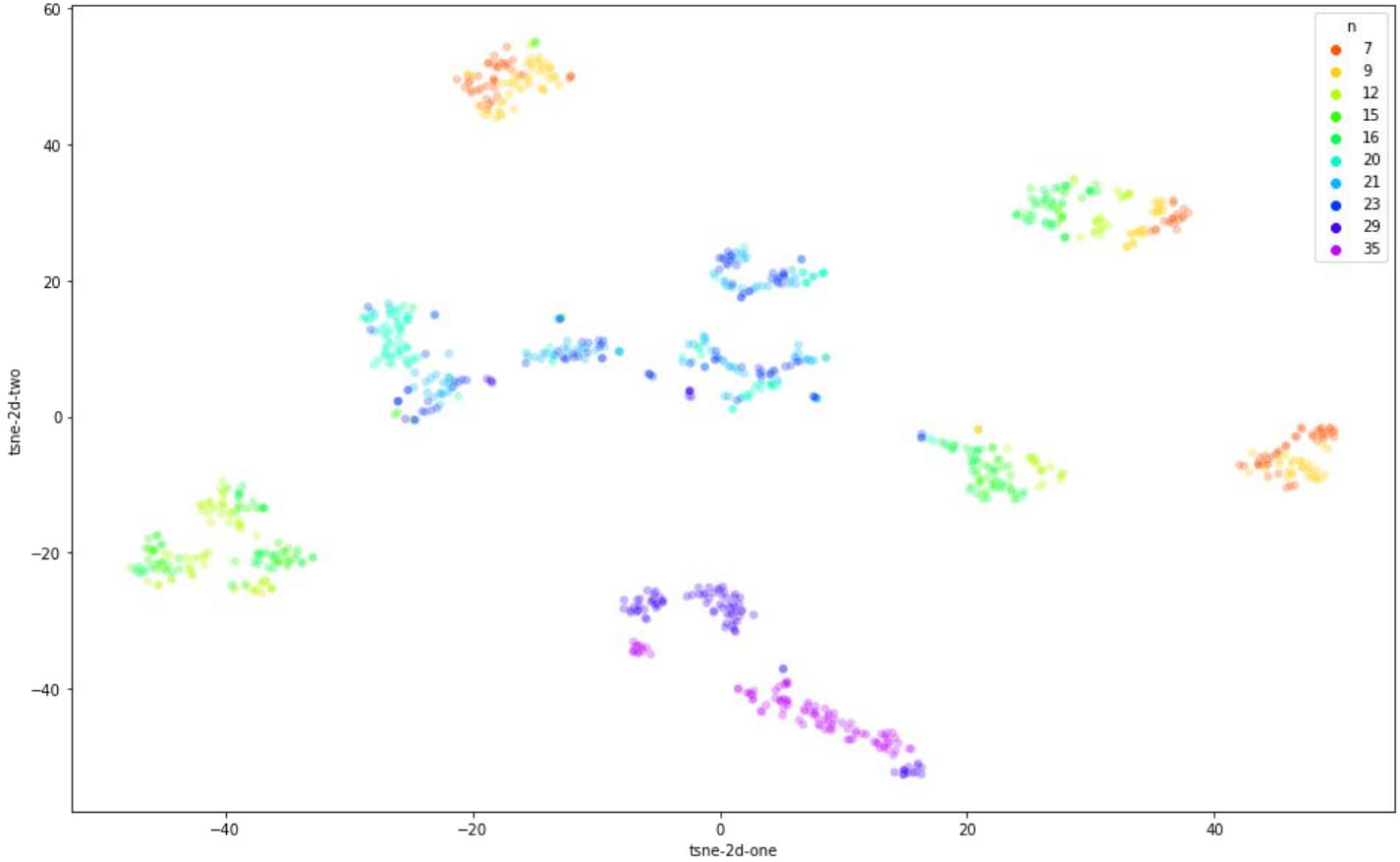
t-SNE on Chao-shen's Dataset



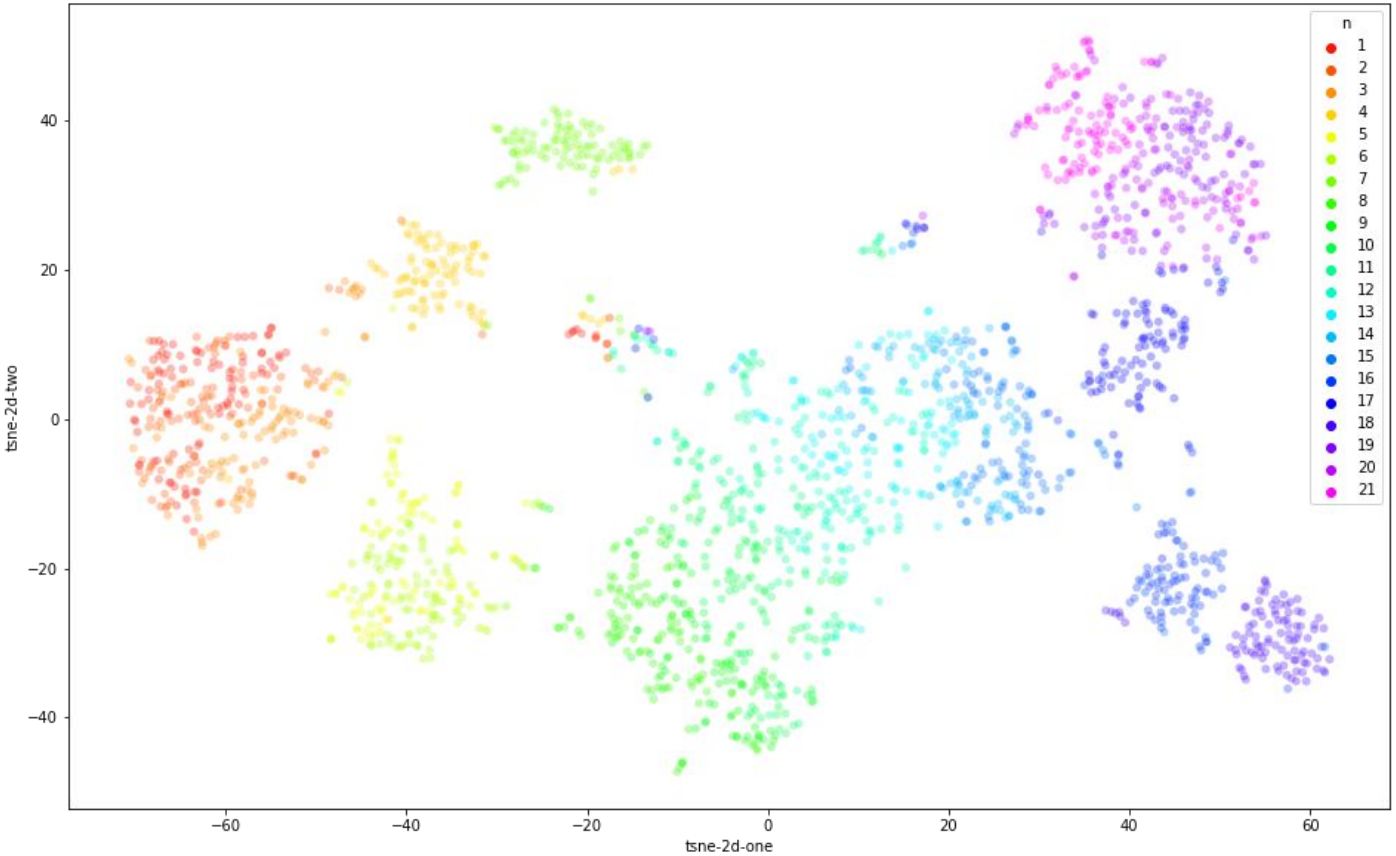
An interesting observation came from the datasets individually.
When we applied t-SNE on the data, with perplexity = 30, the data somewhat split into clusters corresponding to the number of users in the data.

This signifies the presence of separability in the dataset.

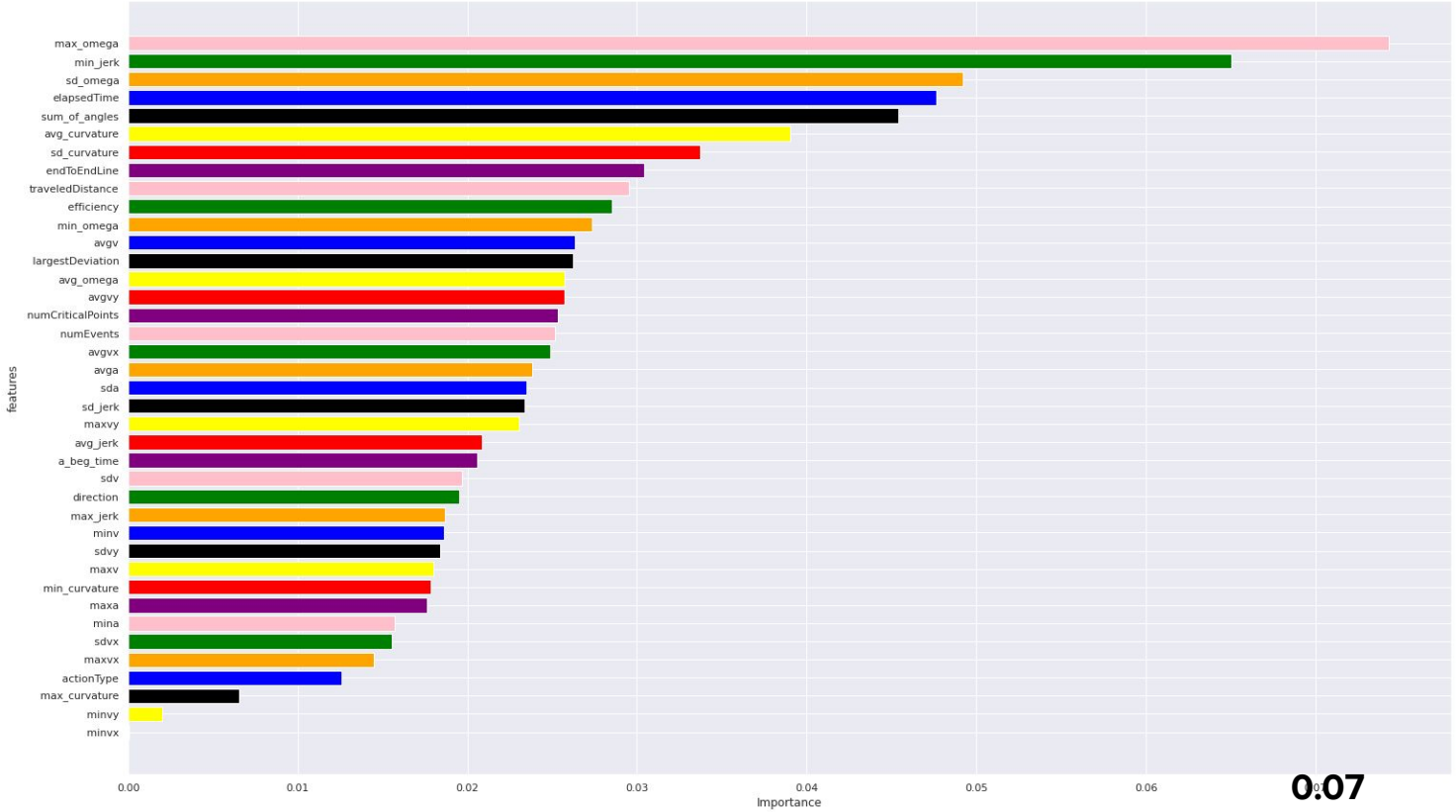
t-SNE on Balabit Dataset



t-SNE on DFL Dataset



Importance Score of Each Feature



0.07

Importance Score of Each Feature

The plot shows the feature importance of each feature in the dataset:-

- max_omega has the highest importance among all the features
- min_jerk has the second highest
- min_vy has the lowest feature importance score
- Min_vx has the zero feature importance
- The plot can help us to drop the columns or features that are not much of importance to the classification model

Pre-processing

1. **Shuffling the dataset**: The dataset had session records for users in a continuous fashion. We shuffle the data to increase the efficiency of our model.
2. **Filling the null values**: Incidentally there were no null values in the dataset.
3. **Normalization**: The dataset given to us had varying values for different attributes, and hence normalizing the dataset was a necessity.
4. **Feature Importance:** The dataset has 39 features so, it important to calculate importance score and drop some features that don't contribute much to the classification model.

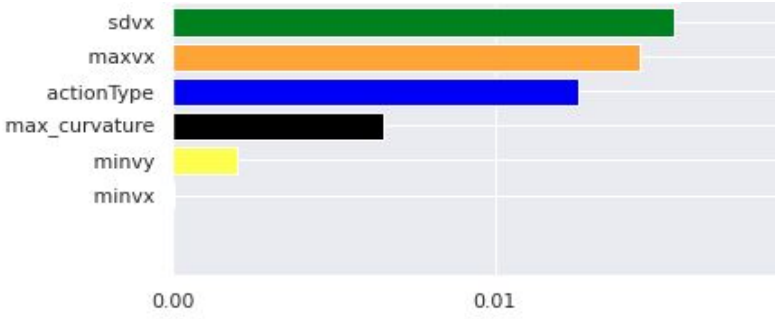
Pre-processing

There are columns that have 0 or very less feature importance

e.g 1) minvx : 0

2) minvy : 0.003

3) max_curvature : 0.007



We have dropped these three features form the dataset as they have less feature importance and are not contributing much to the model. After removing these features performance of the models were increased slightly.

Methodology

Decision Tree Classifier: Decision Tree Classifier is a tree based model that has hyperparameters including max depth, max features used for splitting

Random forest Classifier: Type of ensemble learning, which uses a number of weak classifiers(dtrees) and then gives the label based on majority voting.

(RandomizedSearchCV for hyperparameter tuning)

ADAboost Classifier: Also a type of ensemble learning, where trees are not independent of each other, and grow on top of each other.

Methodology

GBDT: Combining decision trees with a technique called boosting. Thus, GBDT is also an ensemble method. Boosting means combining a learning algorithm in series to achieve a strong learner from many sequentially connected weak learners.

LightGBM: Light Gradient Boosting Machine is tree based learning algorithm. It uses two types of techniques which are gradient Based on side sampling or GOSS and Exclusive Feature bundling or EFB

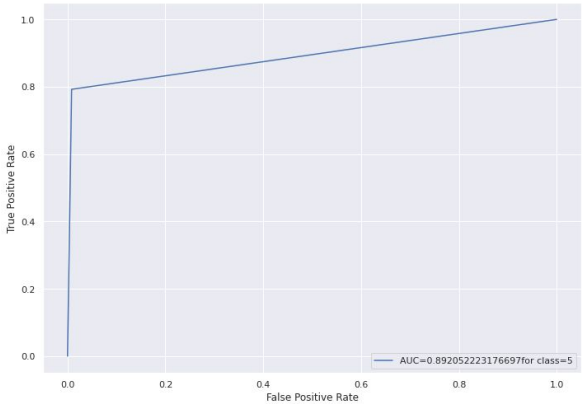
MLP : We tried a multilayer perceptron using 3 hidden layers namely 256,128,128,64 with sgd.

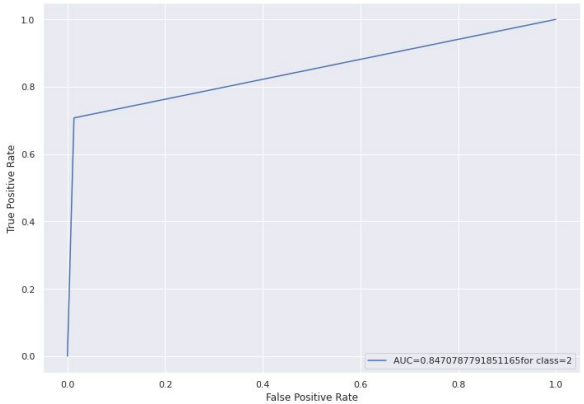
Result and Analysis

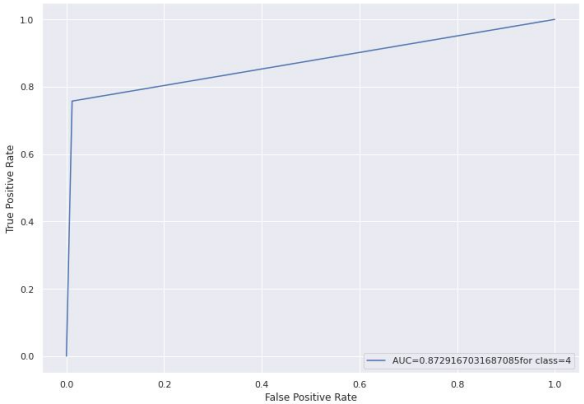
	Training Data			Test Data		
Technique	**Precision**	**Accuracy**	**F1 Score**	**Precision**	**Accuracy**	**F1 Score**
MLP	0.85	0.86	0.86	0.64	0.64	0.64
Decision Tree	0.99	0.99	0.99	0.72	0.71	0.72
Random Forest	0.99	0.99	0.98	0.76	0.77	0.77
Random Forest (threshold = 0.2)	0.98	0.99	0.99	0.88	0.88	0.88
AdaBoost	0.76	0.77	0.77	0.77	0.77	0.77
AdaBoost(threshold = 0.7)	0.98	0.99	0.99	0.89	0.89	0.89
GBDT	0.49	0.48	0.49	0.43	0.42	0.41
LightGBM	0.79	0.79	0.78	0.62	0.61	0.61

Results & Analysis

The ROC curves using Decision Tree Classifier for some of the classes:







Timeline

Week	Tasks
1	Reading related research papers and exploring recent works on the topic
2	Collecting data from online resources
3	Analyzing the given data and finding relevant features
4	Studying about various ML algorithms in detail.
5	Running various ML models to determine error and accuracy and hence find the best model amongst these.
6	Tuning hyperparameters to get best accuracy and performance.

Contribution

Aaditya Gupta:- Methodology, Results

Harjeet Singh Yadav:- Dataset Collection, pre-processing, and description.

Rohan Kulkarni:- Dataset Analysis & visualization

Hemang Dahiya:- Motivation, Survey