

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR



**DEPARTMENT OF ELECTRONICS & ELECTRICAL
COMMUNICATION**

M.Tech FIRST YEAR

VIPES

A Report on

**USER AUTHENTICATION BASED ON KEYSTROKE DATA
USING ONE-CLASS SUPPORT VECTOR MACHINE**

Date: 8th nov 2020

Soumyadeep Kal (20EC65R01)

Krishnendu Ghosh (20EC65R02)

Dishant Satuley (20EC65R03)

Navin Tiwari (20EC65R08)

Chaitanya Hegde (20EC65R23)

Table of Contents

| | |
|-------------------------|----|
| 1. PROBLEM STATEMENT | 3 |
| 2. INTRODUCTION | 3 |
| 3. METHODOLOGY | 4 |
| 4. RESULTS AND ANALYSIS | 10 |
| 5. CONCLUSION | 10 |
| 6. REFERENCES | 11 |

1. PROBLEM STATEMENT

Authenticate the user by training and testing One Class Support Vector Machine classifier by keystroke dynamics of the neutral, happy and sad mood data.

Use five-fold validation to report results.

2. INTRODUCTION

Keystroke dynamics is the study of the typing patterns of different people. These unique typing patterns of each person can be used to distinguish them from one another. Each user has a certain unique feature of typing - for example, for how long a user presses the keys, how much time difference between different consecutive key presses, etc. These features are used to authenticate a person uniquely.

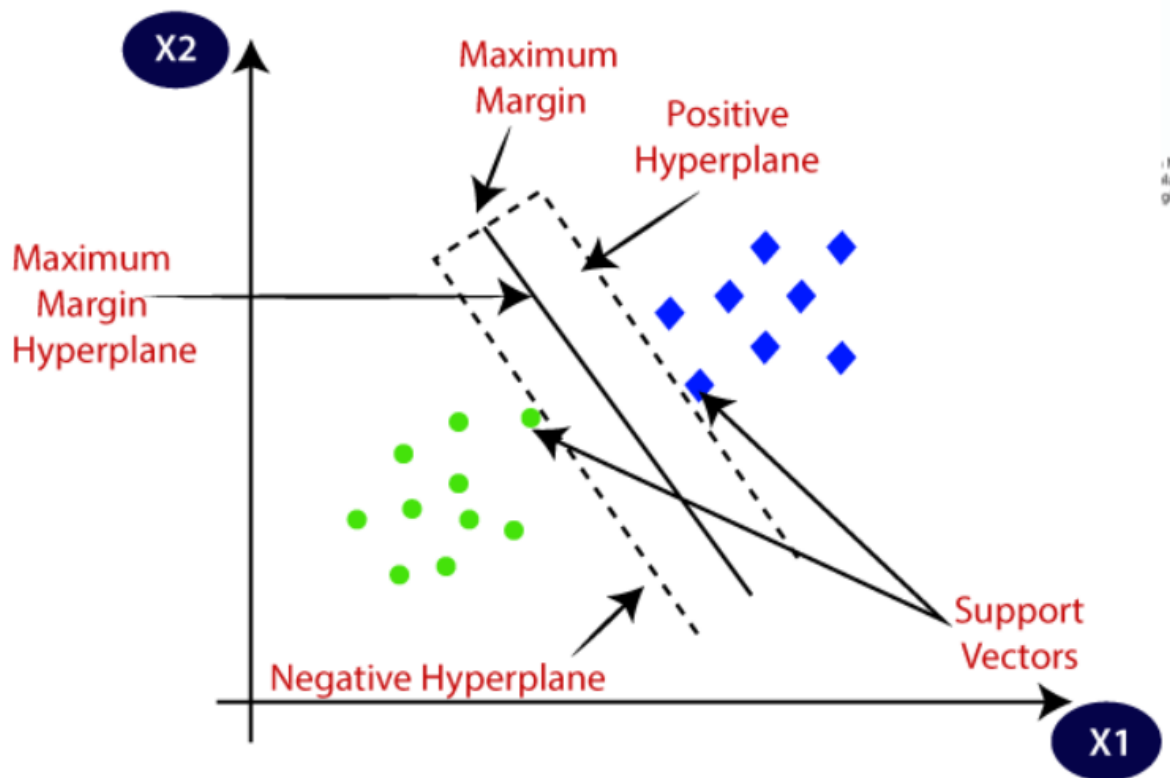
Keystrokes data make a vast area of research in biometrics because of its practical applications in everyday life . One famous application is in the **Coursera** website using a typing test as a method to verify the students. There is no additional hardware required to collect the keystrokes; only a keyboard is enough .

In our project , we verify the identity of different users based on their keystroke data. A classifier model is first trained by providing it with the typing patterns of the users to be enrolled, multiple patterns per user. The model is then provided with test patterns for verification. The particular classifier we use here is **Support Vector Machine** or **SVM**.

3. METHODOLOGY

Support Vector machines are widely used as a classifier because of its ease of implementation and its ability to classify multidimensional data points distinctly.

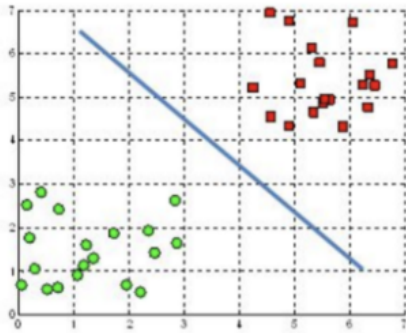
The main objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data.



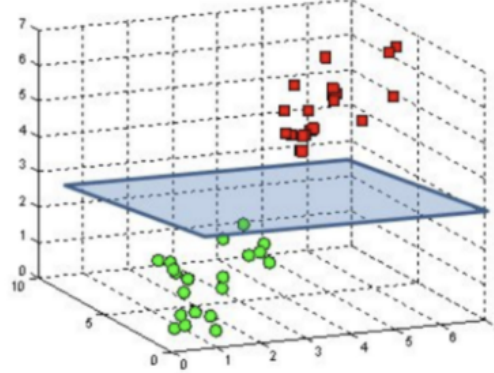
Hyperplane:

A hyperplane is a decision plane which separates between a set of objects having different class memberships.

A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane



Support Vectors:

Support vectors are the data points, which are closest to the hyperplane.

Margin:

A margin is a gap between the two lines on the closest class points.

This algorithm looks to select a hyperplane with maximum possible margin between the support vectors of the dataset.

For our classification purpose we train our one class SVM classifier using the following features -

- Hold time
- Latencies

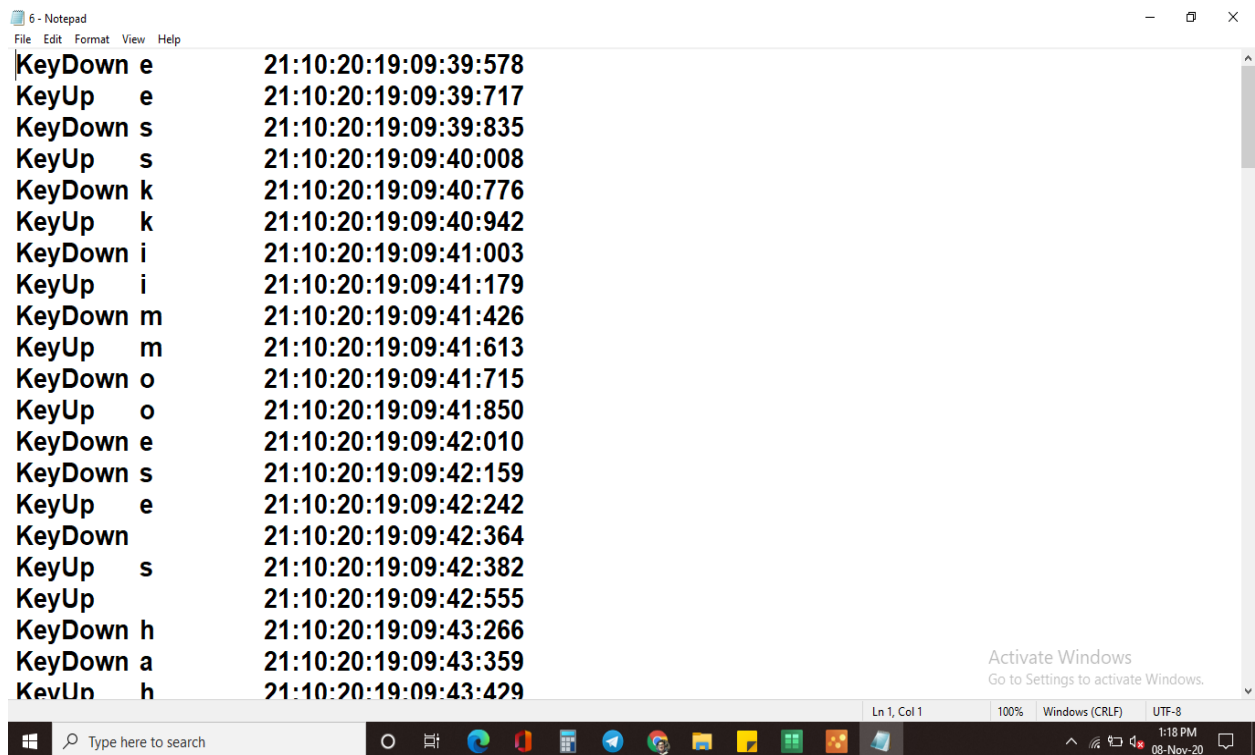
Target attribute is the username.

Data Acquisition

Keystroke data was collected considering the Happy, Sad or Neutral mood of the user. First step was to collect the data keystroke data. A software was installed in the computer of each user. Then each user was requested to participate in a data acquisition procedure on alternative days of the week. So data was collected in different sessions.

A total of 7 users participated in the data acquisition part and a total of 12 days data was collected in 4 weeks.

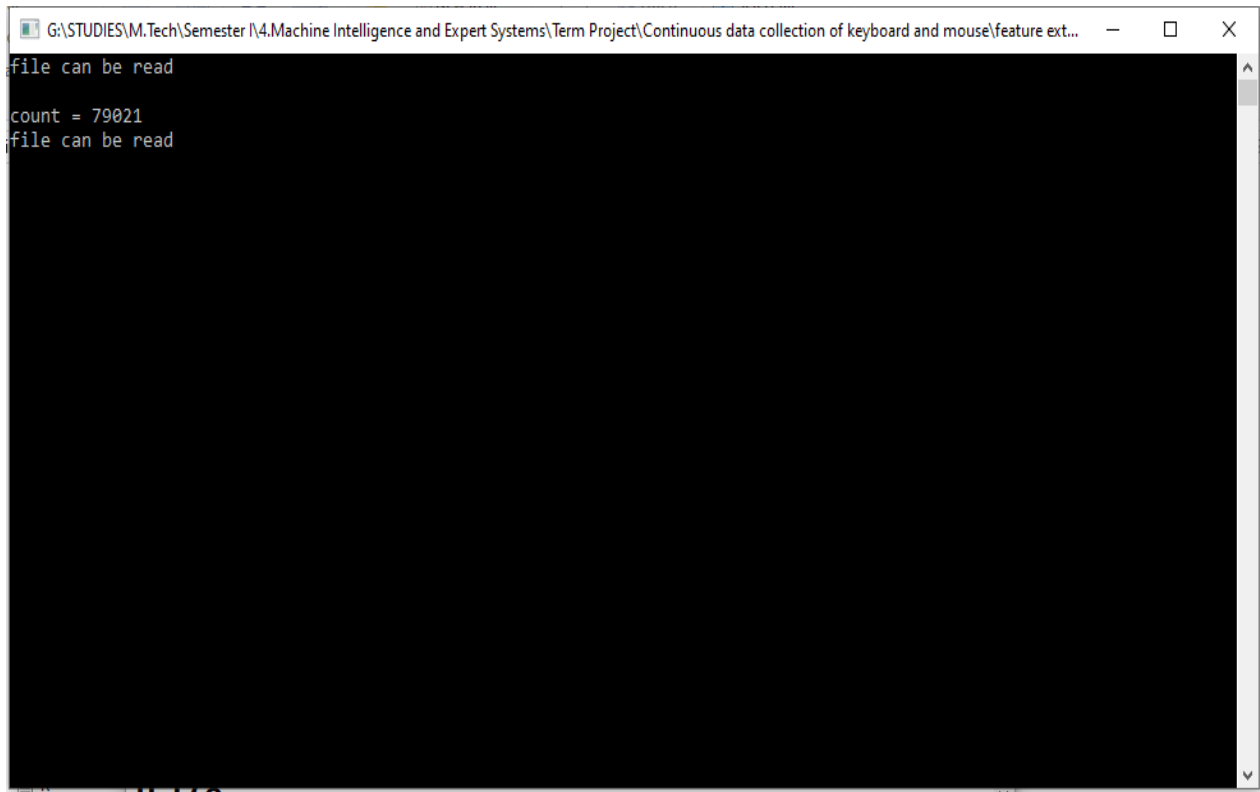
Next step was to preprocess this data. By running a file feature_extractor.exe, two features hold time and latencies for different key combinations were extracted from the keystroke data. This data was stored in a CSV file after preprocessing it.



The screenshot shows a Notepad window titled "6 - Notepad" with a menu bar (File, Edit, Format, View, Help). The text inside the window is a list of keyboard events, each followed by a timestamp. The events are: KeyDown e, KeyUp e, KeyDown s, KeyUp s, KeyDown k, KeyUp k, KeyDown i, KeyUp i, KeyDown m, KeyUp m, KeyDown o, KeyUp o, KeyDown e, KeyDown s, KeyUp e, KeyDown, KeyUp s, KeyUp, KeyDown h, KeyDown a, and KeyUp h. The timestamps range from 21:10:20:19:09:39:578 to 21:10:20:19:09:43:429. The status bar at the bottom shows "Ln 1, Col 1", "100%", "Windows (CRLF)", "UTF-8", and the system clock "1:18 PM 08-Nov-20".

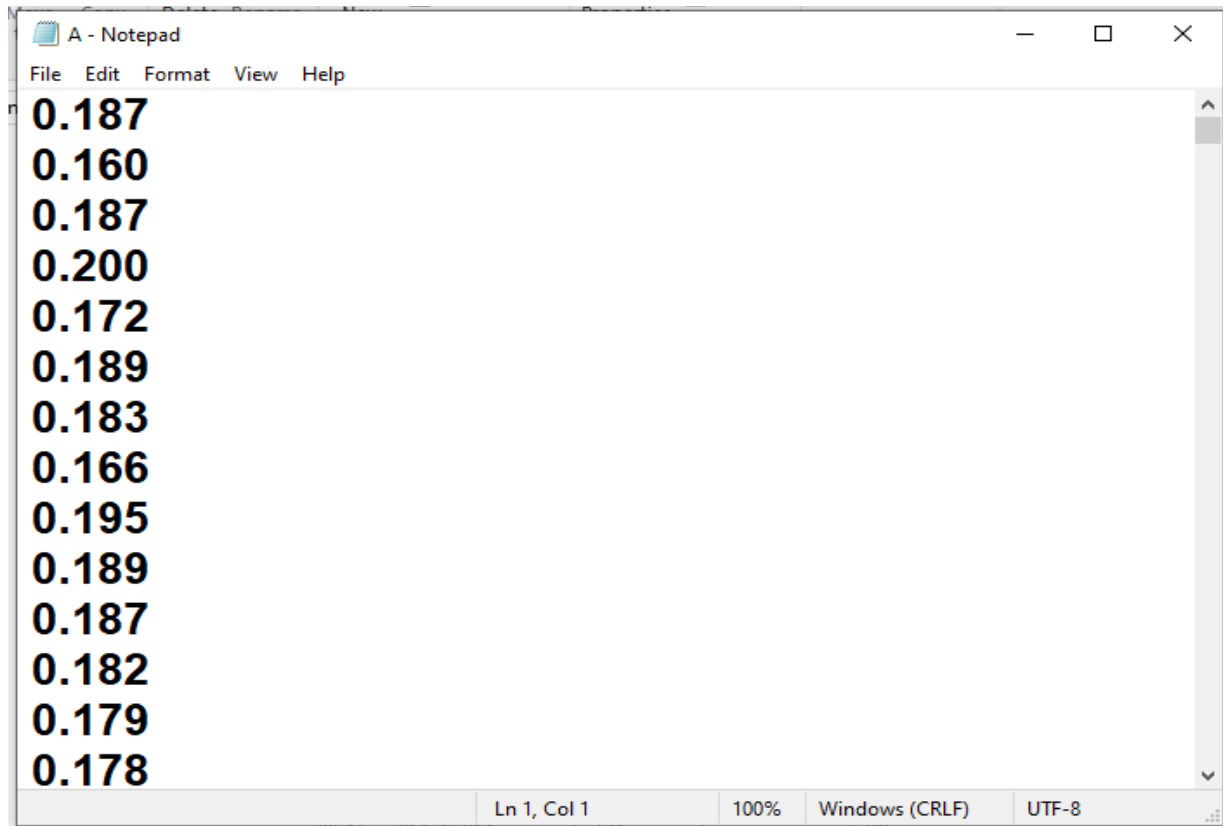
| Event | Timestamp |
|-----------|-----------------------|
| KeyDown e | 21:10:20:19:09:39:578 |
| KeyUp e | 21:10:20:19:09:39:717 |
| KeyDown s | 21:10:20:19:09:39:835 |
| KeyUp s | 21:10:20:19:09:40:008 |
| KeyDown k | 21:10:20:19:09:40:776 |
| KeyUp k | 21:10:20:19:09:40:942 |
| KeyDown i | 21:10:20:19:09:41:003 |
| KeyUp i | 21:10:20:19:09:41:179 |
| KeyDown m | 21:10:20:19:09:41:426 |
| KeyUp m | 21:10:20:19:09:41:613 |
| KeyDown o | 21:10:20:19:09:41:715 |
| KeyUp o | 21:10:20:19:09:41:850 |
| KeyDown e | 21:10:20:19:09:42:010 |
| KeyDown s | 21:10:20:19:09:42:159 |
| KeyUp e | 21:10:20:19:09:42:242 |
| KeyDown | 21:10:20:19:09:42:364 |
| KeyUp s | 21:10:20:19:09:42:382 |
| KeyUp | 21:10:20:19:09:42:555 |
| KeyDown h | 21:10:20:19:09:43:266 |
| KeyDown a | 21:10:20:19:09:43:359 |
| KeyUp h | 21:10:20:19:09:43:429 |

A snapshot of data before processing it.

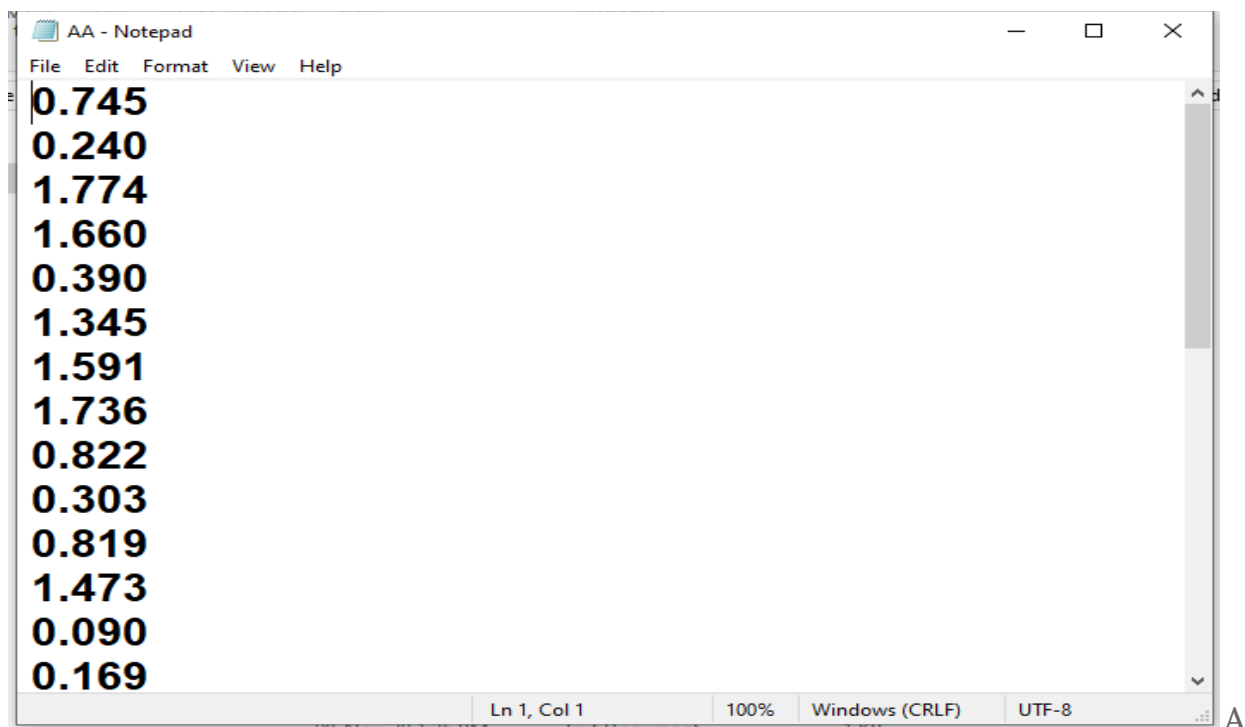


```
G:\STUDIES\M.Tech\Semester I\4.Machine Intelligence and Expert Systems\Term Project\Continuous data collection of keyboard and mouse\feature ext...
file can be read
count = 79021
file can be read
```

A snapshot of Feature Extractor running program



A snapshot of hold time file extracted from keystroke data



snapshot of latency time extracted from keystroke data

The screenshot displays an Excel spreadsheet with the following data (approximate values from the visible portion):

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|----|------------|-----------------|-----------------|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | Use | 72_Latenc2_Hold | 25_Latenc2_Hold | 5_Latenc5_Hold | 0.07 | 1.084 | 0.09 | NaN | 0.08 | 0.223 | 0.075 | 0.115 | 0.103 | 0.734 | 0.109 | 0.245 | 0.13 | 0.427 | 0.12 | 0.155 |
| 2 | navin | NaN | NaN | NaN | 0.089 | 0.621 | 0.059 | 1.174 | 0.024 | 0.184 | 0.059 | 0.274 | 0.103 | 0.577 | 0.064 | 0.11 | 0.055 | 0.141 | 0.111 | 0.405 |
| 3 | sourav | NaN | 0.576 | 0.481 | 0.157 | 0.214 | 0.089 | 0.712 | 0.073 | 0.021 | 0.124 | 0.112 | 0.162 | 0.25 | 0.142 | 0.105 | 0.118 | 0.099 | 0.14 | 0.103 |
| 4 | krishnendi | NaN | 0.297 | NaN | 0.159 | 0.422 | 0.175 | NaN | 0.112 | 0.2 | 0.152 | 0.193 | 0.143 | 0.36 | 0.112 | 0.096 | 0.152 | 0.24 | 0.192 | 0.297 |
| 5 | kumarsun | NaN | NaN | NaN | 0.155 | 0.151 | 0.097 | 0.764 | 0.119 | 0.157 | 0.114 | 0.093 | 0.115 | 0.303 | 0.099 | 0.283 | 0.119 | 0.15 | 0.183 | 0.134 |
| 6 | sourav | NaN | 0.404 | NaN | 0.089 | 0.621 | 0.059 | 1.47 | 0.06 | 0.205 | 0.101 | 0.167 | 0.047 | 0.333 | 0.057 | 0.135 | 0.069 | 0.207 | 0.093 | 0.099 |
| 7 | sourav | NaN | 0.142 | 0.481 | 0.089 | 0.621 | 0.059 | 1.47 | 0.06 | 0.205 | 0.101 | 0.167 | 0.047 | 0.333 | 0.057 | 0.135 | 0.069 | 0.207 | 0.093 | 0.099 |
| 8 | navin | NaN | NaN | NaN | 0.07 | 1.363 | 0.07 | NaN | 0.089 | 0.309 | 0.059 | 0.155 | 0.069 | 0.925 | 0.13 | 0.245 | 0.073 | 0.252 | 0.09 | 0.119 |
| 9 | dishant | NaN | 9.467 | NaN | NaN | NaN | NaN | NaN | 0.1 | 0.228 | 0.2 | 3.413 | 0.14 | 0.709 | 0.155 | NaN | 0.164 | 0.261 | 0.124 | 0.131 |
| 10 | chaitanya | 2.007 | 0.113 | 0.176 | 0.174 | 1.315 | 0.016 | 0.294 | 0.087 | 0.235 | 0.088 | 0.579 | 0.083 | 0.442 | 0.102 | 0.158 | 0.089 | 0.75 | 0.152 | 0.11 |
| 11 | navin | NaN | NaN | NaN | 0.07 | 1.363 | 0.07 | NaN | 0.086 | 0.295 | 0.025 | 0.205 | 0.065 | 0.925 | 0.104 | 0.245 | 0.073 | 0.225 | 0.11 | 0.119 |
| 12 | chaitanya | 2.007 | 0.108 | 0.191 | 0.129 | 3.96 | 0.063 | 0.343 | 0.097 | 0.253 | 0.104 | 0.914 | 0.102 | 0.454 | 0.093 | 0.147 | 0.11 | 0.157 | 0.106 | 0.142 |
| 13 | dishant | NaN | 73.116 | NaN | NaN | NaN | NaN | NaN | 0.102 | 0.349 | 0.151 | 0.252 | 0.169 | 0.421 | 0.138 | NaN | 0.192 | 0.301 | 0.173 | 0.133 |
| 14 | sourav | NaN | 0.308 | 0.364 | 0.045 | 0.457 | 0.109 | 1.178 | 0.08 | 0.23 | 0.09 | 0.163 | 0.054 | 1.099 | 0.064 | 0.127 | 0.074 | 0.298 | 0.125 | 0.103 |
| 15 | chaitanya | 2.007 | 0.108 | 0.176 | 0.174 | 1.315 | 0.016 | 0.294 | 0.087 | 0.227 | 0.09 | 0.232 | 0.125 | 0.47 | 0.083 | 0.164 | 0.074 | 0.305 | 0.116 | 0.11 |
| 16 | chaitanya | 2.007 | 0.108 | 0.176 | 0.174 | 1.315 | 0.016 | 0.294 | 0.087 | 0.227 | 0.09 | 0.232 | 0.125 | 0.47 | 0.083 | 0.164 | 0.074 | 0.305 | 0.116 | 0.11 |
| 17 | chaitanya | 2.007 | 0.108 | 0.364 | 0.135 | 1.088 | 0.03 | 0.358 | 0.103 | 0.206 | 0.11 | 0.188 | 0.081 | 0.983 | 0.101 | 0.16 | 0.108 | 0.212 | 0.116 | 0.151 |
| 18 | navin | NaN | NaN | NaN | 0.091 | 0.625 | 0.06 | | | | | | | | | | | | | |

Once the Data Acquisition part is done and the CSV file is generated, load the csv file. Then putting the NaN in the empty slots in all the columns. Then the data is divided into to train and test where the train contains 80% composition and the testing with remains. So One class SVM is trained with 80% data. The kernel used is 'rbf' and the gamma '0.005'. Then the 5 - fold validation is used. Model is supplied with unseen examples. The fit() function was used to train One class SVM objects. The decision_function() is used in calculating similarities scores for the test samples .

4. RESULTS AND ANALYSIS

The model was tested with different users. The accuracy is recorded for each user. If the EER is low, the user is valid else i.e if the EER is low then the accuracy for right validation is high. In this way, for each user the EER value is recorded. Then average is taken of all i.e, in our case there are totally 7 user, so sum of all EER's and divide by 7. This gives the accuracy of the whole model. The average EER obtained is 0.0377. The EER for each user too is obtained, that is in the range of 0 and 0.25. When the average of all the EER is taken EER obtained is 0.0377.

5. CONCLUSION

From the results obtained above we can conclude that the Support vector machine (SVM) is efficient to successfully classify the users based on their unique typing pattern.

In addition, keystroke dynamics is one of the most efficient and inexpensive behavioral biometrics that can be used to authenticate a user. So SVM , along with keystroke dynamics can be utilized as the classification engine of users with high efficacy due to its high recognition rate and efficient processing.

As a future work, the mood of an user also can be analyzed from the collected data if we segregate the data based on the mood of an user and train them .For that we may need more data with different emotional states.

6. REFERENCES

- [1] Y. Sang, H. Shen, and P. Fan. Novel Impostors Detection in Keystroke Dynamics by Support Vector Machine. Lecture Notes in Computer Science, 3320:666– 669, 2004.
- [2] G. Azevedo, G. Cavalcanti, and C. Filho. An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting. 2007 IEEE Congress on Evolutionary Computation,
- [3] <http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf> -
- [4] <https://www.ugpti.org/smartse/resources/downloads/support-vector-machines.pdf>
- [5] https://en.wikipedia.org/wiki/Support_vector_machine