EC60091: Machine Intelligence and Expert Systems

Term Project

Biometric authentication using keystroke dynamics

Authors
Joshua Peter Ebenezer (15EC10023)
Kaustav Brahma (15EC10026)
Rajdeep Biswas (15EC10043)
Pourush Sood (15EC35011)
Manan Khaneja (16PH20023)

Course Instructor
Prof. Sudipta Mukhopadhyay

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology Kharagpur

October 30, 2018

Contents

1 Introduction **1**

2 Description of the Problem **1**

3 Data Collection **1**

4 Feature Extraction **2**

5 Normalization and Pre-processing **3**

6 Brief theory of One-Class SVMs **3**

7 Cross Validation **4**

8 Results **5**
8.1 Effect of kernels . . . 5
8.1.1 Quadratic Kernel . . . 5
8.1.2 Linear Kernel . . . 5
8.1.3 RBF Kernel . . . 6
8.2 Effects of PCA and normalization . . . 6
8.2.1 Without applying PCA and without normalizing the feature vector . . . 6
8.2.2 Without applying PCA and normalizing the feature vector . . . 7
8.2.3 Applying PCA and without normalizing the feature vector . . . 7
8.2.4 Applying PCA and normalizing the feature vector . . . 8
8.3 Variation with γ and ν . . . 8
8.3.1 3D plot of accuracy for 15EC10058 (vs γ and ν) . . . 8
8.3.2 Variation with γ for $\nu = 10^{-3}$ . . . 9
8.3.3 Variation with ν for $\gamma = 0.1$ . . . 9
8.4 Average Precision . . . 10
8.5 Average Recall . . . 11

9 Conclusion **12**

List of Figures

1 Defining the hold times and latencies . . . 2
2 Flow of Feature Extraction . . . 2
3 An example of Feature Extraction . . . 3
4 Quadratic Kernel . . . 5
5 Linear Kernel . . . 5
6 RBF Kernel . . . 6
7 Without applying PCA and without normalizing the feature vector . . . 6
8 Without applying PCA and normalizing the feature vector . . . 7
9 Applying PCA and without normalizing the feature vector . . . 7
10 Applying PCA and normalizing the feature vector . . . 8
11 3D plot of accuracy for 15EC10058 (vs γ and ν) . . . 8
12 Variation with γ for $\nu = 10^{-3}$ . . . 9
13 Variation with ν for $\gamma = 0.1$ . . . 9
14 Average Precision vs ν for $\gamma = 0.1$ . . . 10
15 Average Precision vs γ for $\nu = 10^{-3}$ . . . 10
16 Average Recall vs ν for $\gamma = 0.1$ . . . 11
17 Average Recall vs γ for $\nu = 10^{-3}$ . . . 11

1 Introduction

As early as the beginning of the 20th century, psychologists, and mathematicians have experimented with human actions. Psychologists have demonstrated that human actions are predictable in the performance of repetitive, and routine tasks. Handwriting and typing are examples of distinct manual skills that have measurable characteristics that are unique to those who perform the task.

Since the beginning of civilization, humans are able to recognize the person coming into a room from the sound of steps of the individual. Clearly, each person has a unique way of walking. Similarly, telegraph operators were able to find out who was sending message by just listening to the characteristics of dots and dashes. Today, the telegraph keys have been replaced by other input/output devices such as keyboard and mouse.

It has been established that keyboard characteristics are rich in cognitive qualities and hold promise as an individual identifier. The keystroke dynamics is an important behavioral biometric which can be used to authenticate a user. Primarily, two features, i.e. hold time of an individual key and the latency of the consecutive keystrokes are used for authentication. The emotional state of the person (happy, sad or neutral) can affect the keyboard dynamics. So, analysis of mood of a person may prove to be very crucial while authentication using keystroke dynamics.

2 Description of the Problem

A Java based GUI was used to collect the keystroke dynamics for 15 users. The GUI asks the user to type in a displayed sentence. This keystroke dynamics (the latency of consecutive keystrokes and hold time of an individual key) corresponds to the dynamics for the neutral emotional state. The GUI then presents an emotional video (to make the user happy or sad) and subsequently the user is asked to type the same sentence again. This time the keystroke dynamics collected corresponds to the emotional state of the person.

These keystroke dynamics were used to extract the common features across all the 15 users. The alphabets and all pairs of alphabets which were typed in by every user. The corresponding hold time and latencies of the common features were extracted to create a feature vector for each of the 15 users.

These feature vectors were then fed to a 1-class SVM with 5-fold cross-validation to train the SVM and then test it and report its accuracy of prediction.

3 Data Collection

The keystroke dynamics corresponding to the 3 emotional states (happy, sad, neutral) have been collected from 15 users. The interface first asks the user to type in a displayed sentence. This corresponds to the keystroke dynamics of the neutral emotional state. The interface then presents an emotional video to the person (happy or sad).

After the user has watched the video, the user is asked to type in the same original sentence. This keystroke dynamics is for the happy or sad emotional state.

The data to be worked upon was collected using an interface in which users were asked to type a sentence, randomly selected from a group of sentences. The user was then shown a video, and asked to type again while registering an emotion. This was not of use to us since we did not consider any parameter related to the emotion of the individual in our feature vector. The data of each individual was divided week-wise and further sentence and emotion wise.

All the data of an individual (week-emotion-sentence) was first combined into a single file. Then, the features were extracted from these files.

4 Feature Extraction

Hold time of a keypress is the time between the press and release of the key. Other keys may or may not have been pressed in this period. Latency is the time between the release of a key and the press of another key. By inter-key, we mean that two keys were pressed successively.

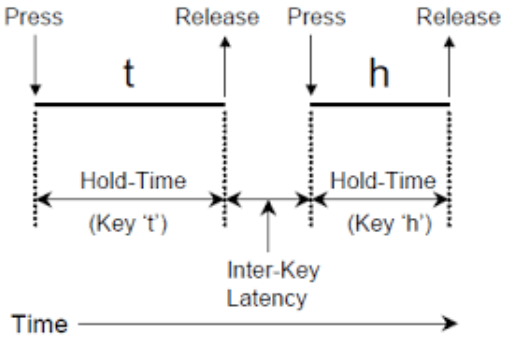

Figure 1: Defining the hold times and latencies

As mentioned in [1], Over many centuries, humans have relied on written signatures to verify the identity of an individual. It has been proven that human hand and its environment make written signatures difficult to forge. It has been shown that the same neurophysiological factors that make written signature unique are also exhibited in an individual typing pattern. Once a computer user types on the keyboard of a computer, he/she leaves a digital signature in the form of keystroke latencies (elapsed time between keystrokes and hold times). The combination of hold times and latencies has been used in [1] and very high identification rate has been observed with machine learning paradigms like neural networks.

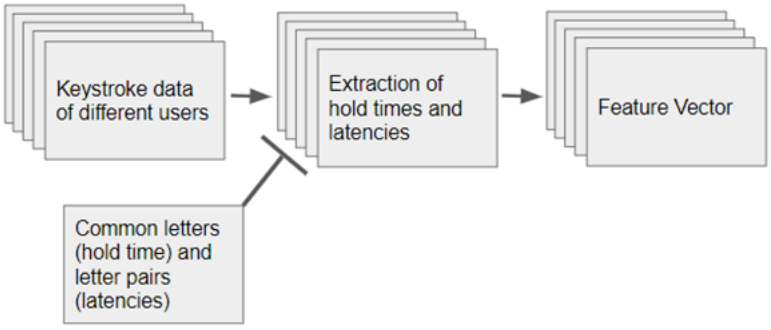

Figure 2: Flow of Feature Extraction

Thus the use of hold times and latencies as features is justified. The hold times and latencies were extracted from the data of each user. Though ideally all users should have written the same sentences, due to mismatch in the data between different groups, we had to extract common keypresses and latency pairs to make the feature vector. Thus finally the hold times and latencies of the same keys and key pairs respectively were taken for each user.

To illustrate how exactly we made the feature vector, let us take a toy example. There are only two users, and the keys pressed by User 1 were only a, p and c. The keys pressed by user 2 were a and p only. The successive key-pairs pressed by User 1 were ap and ac. For User 2, they were ap only. Since the common values are a, p and ap, only these will be taken to form the feature vector. Thus, for User 1, c and ac data will be discarded. Now, in the figure we can also see the contents of the files for User 1. We form a feature vector with header [a,p,ap] and take all possible combinations of these for User 1 to obtain an 256X25 matrix

which we take as the feature vector. The same is done for User 2. In the figure shown below a 4X3 matrix was formed for the sake of brevity.

The number of rows in each file (eg. a.txt), on an average, was found to be 25. If we obtain 25 common letters and letter pairs among all users, we will obtain 25^{25} feature vectors for a single user which is not feasible to train or even store efficiently. So, for each of the 25 common features obtained 256 values were sampled to train the SVM. This gives a total of 256 feature vectors per user.

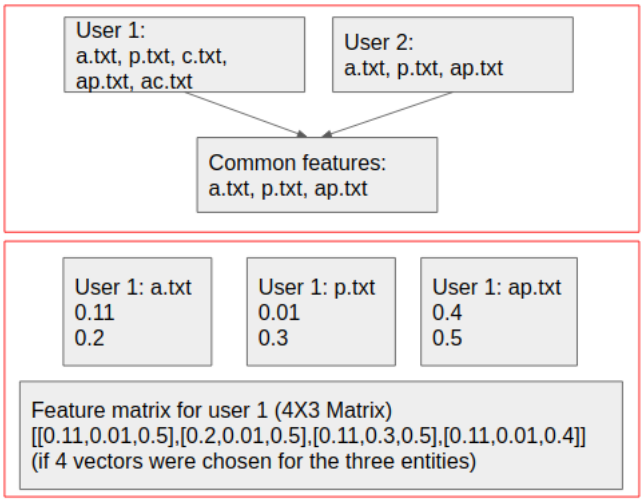

Figure 3: An example of Feature Extraction

[1] M. S. Obaidat, Keystroke Dynamics Based Authentication (Hold Times and Latencies)

5 Normalization and Pre-processing

The training data was normalized to have mean 0 and variance 1. The mean of the training data was subtracted from the testing data and the testing data was then divided by the variance of the training data. The results with and without normalization are presented.

Principal component analysis was performed and 8 principal components were selected from the 25. The results with and without PCA are also presented.

6 Brief theory of One-Class SVMs

The Support Vector Method For Novelty Detection by Schlkopf et al. separates all the data points from the origin (in feature space F) and maximizes the distance from this hyperplane to the origin. This results in a binary function which captures regions in the input space where the probability density of the data lives. It is trained on only the regular, authentic users data.

The following optimization problem is to be solved :

$$\min_{w,\xi_i,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho$$

subject to:

$$(w \cdot \phi(x_i)) \geq \rho - \xi_i \qquad \text{for all } i = 1, \ldots, n$$
$$\xi_i \geq 0 \qquad \text{for all } i = 1, \ldots, n$$

The solution for w and gives the plane and the decision is taken as :

$$f(x) = \text{sgn}((w \cdot \phi(x_i)) - \rho) = \text{sgn}(\sum_{i=1}^{n} \alpha_i K(x, x_i)$$

The parameter ν, also known as the margin of the One-Class SVM, corresponds to the probability of finding a new, but regular, observation outside the frontier. For the kernel $K(x, x)$,

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

where $\sigma \in R$ is a kernel parameter and $\|x - x'\|$ is the dissimilarity measure.

The dissimilarity is the L2 norm. We define $\gamma = 1/2\sigma^2$. Intuitively, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close.

We also show results of the SVM trained with linear and quadratic kernels.

7 Cross Validation

For user i among n users, data corresponding user i and the n-1 remaining users shuffled randomly. The data corresponding to user i is divided into 80:20 ratio, out of which 80% data of user i will go for training. Now a portion of data (totally 5% of size of user is data) from n-1 users is added to 20% data of user i for testing so that resulting ratio for testing becomes 80:20. (80% from user i and 20% from other users). The above operation is repeated 5 times.

8 Results

8.1 Effect of kernels

8.1.1 Quadratic Kernel

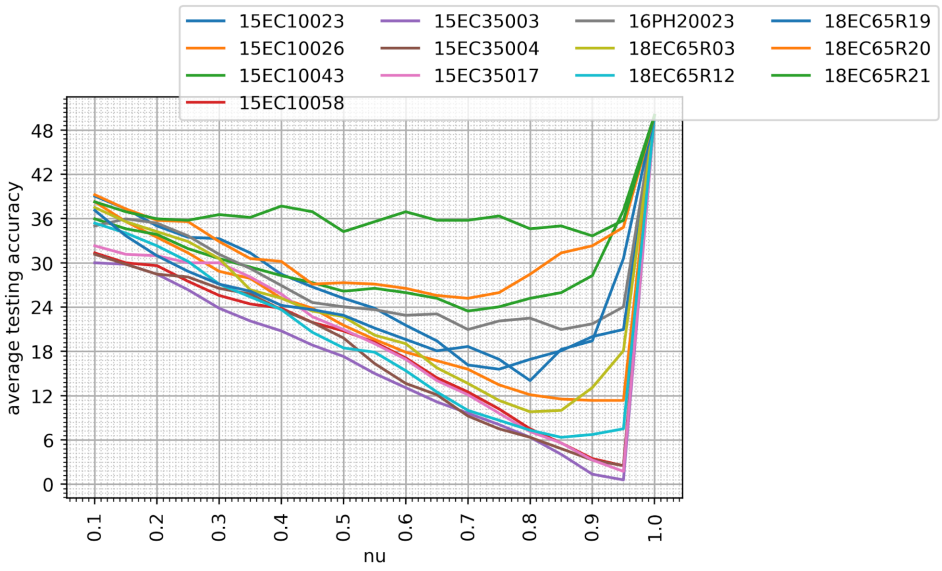

Figure 4: Quadratic Kernel

8.1.2 Linear Kernel

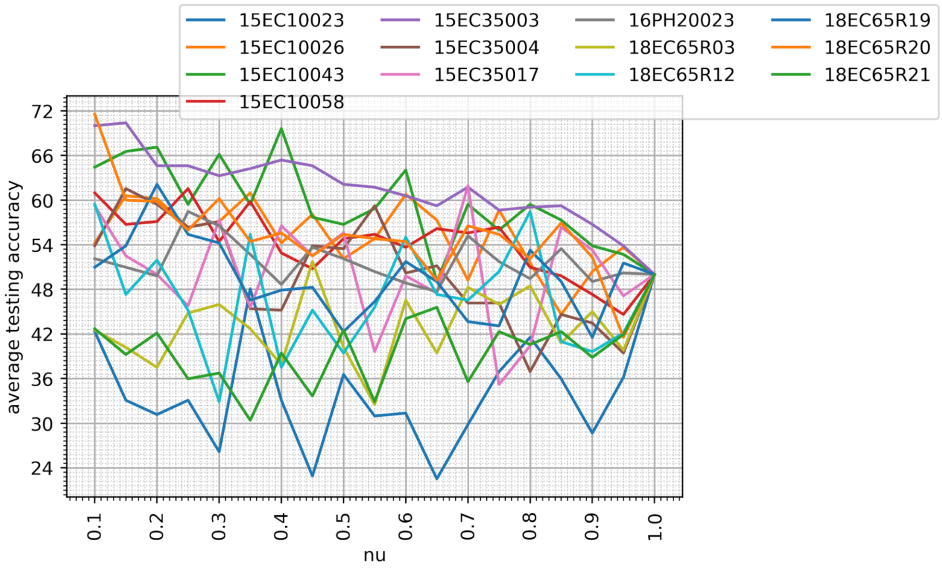

Figure 5: Linear Kernel

8.1.3 RBF Kernel

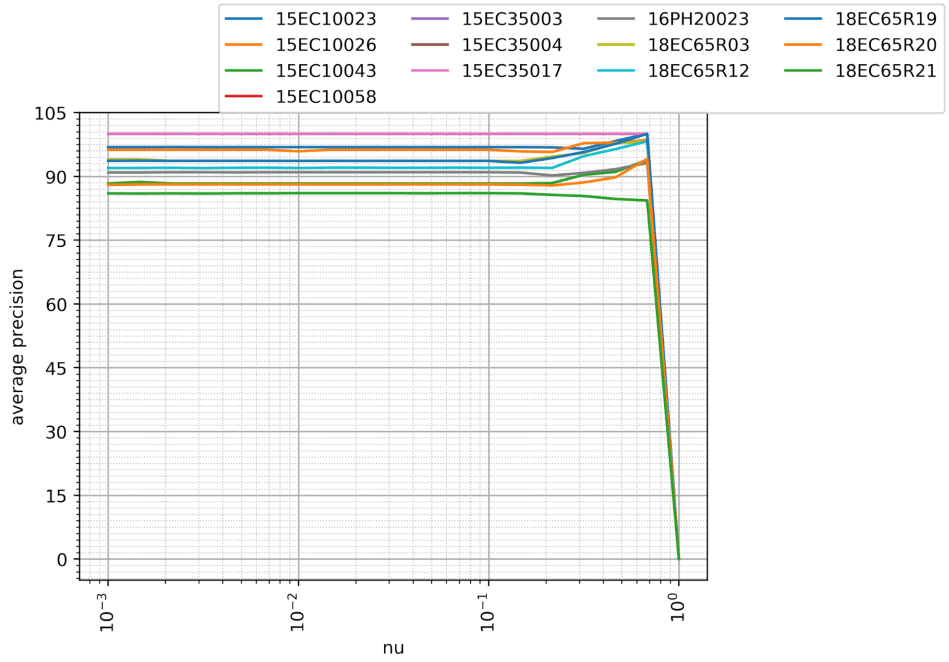

Figure 6: RBF Kernel

We selected the RBF kernel for the final analysis.

8.2 Effects of PCA and normalization

8.2.1 Without applying PCA and without normalizing the feature vector

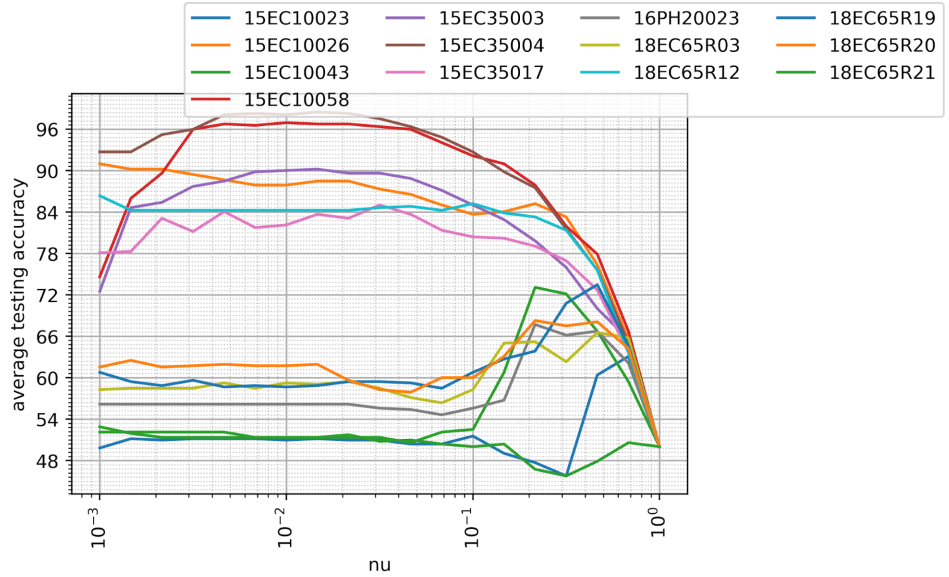

Figure 7: Without applying PCA and without normalizing the feature vector

8.2.2 Without applying PCA and normalizing the feature vector

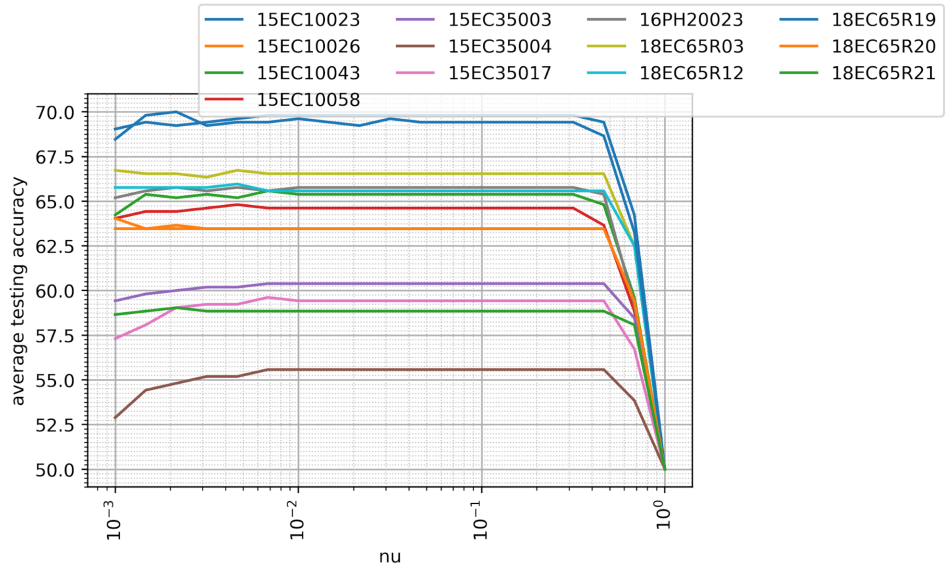

Figure 8: Without applying PCA and normalizing the feature vector

8.2.3 Applying PCA and without normalizing the feature vector

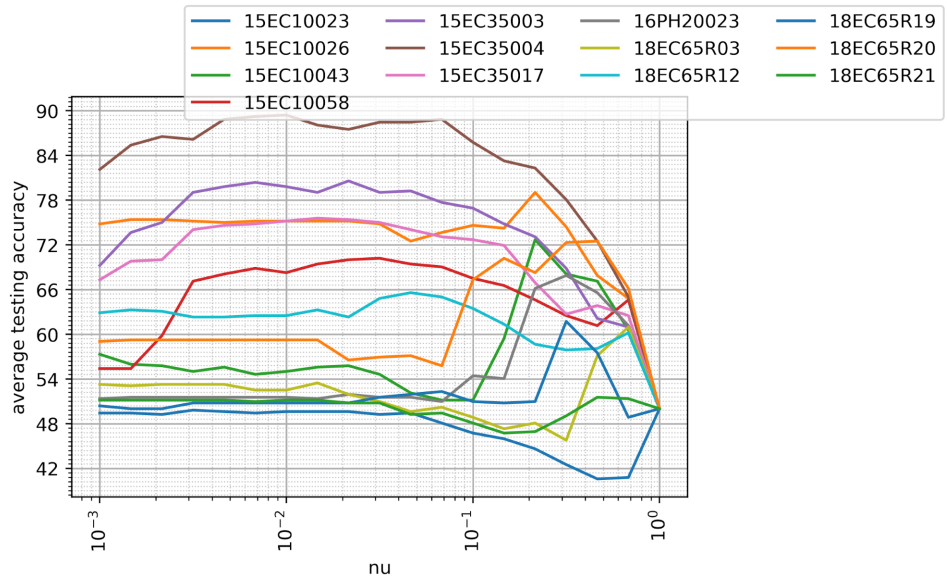

Figure 9: Applying PCA and without normalizing the feature vector

8.2.4 Applying PCA and normalizing the feature vector

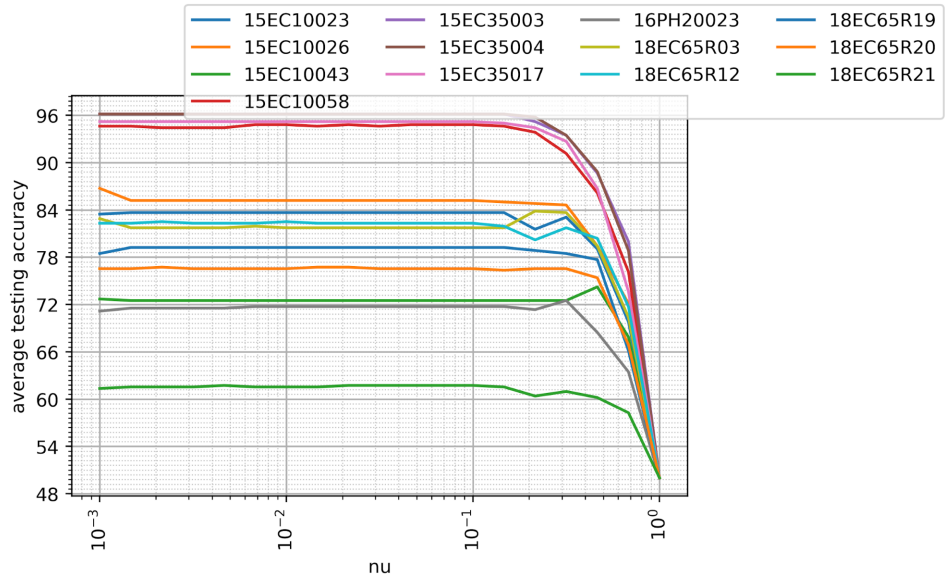

Figure 10: Applying PCA and normalizing the feature vector

8.3 Variation with γ and ν

8.3.1 3D plot of accuracy for 15EC10058 (vs γ and ν)

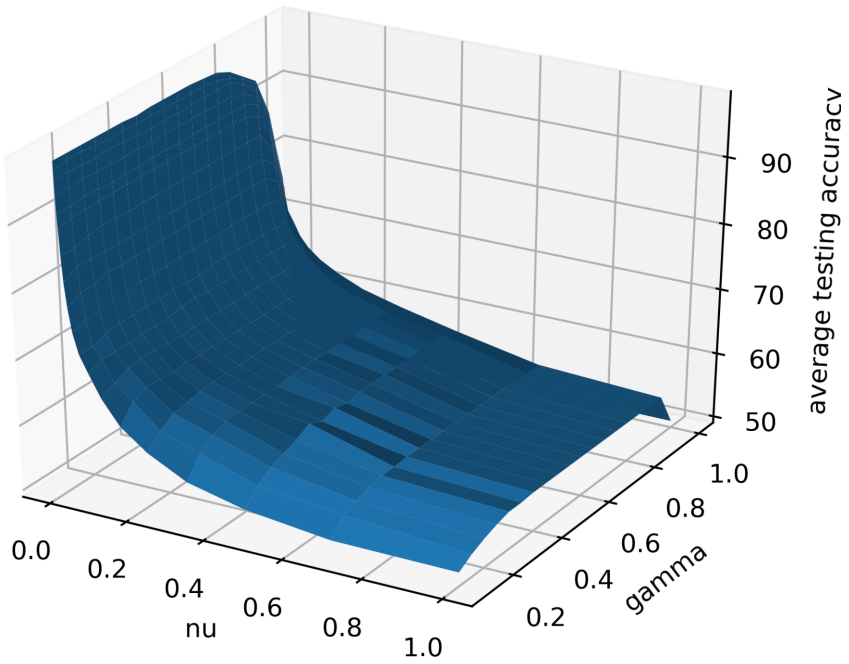

Figure 11: 3D plot of accuracy for 15EC10058 (vs γ and ν)

8.3.2 Variation with γ for $\nu = 10^{-3}$

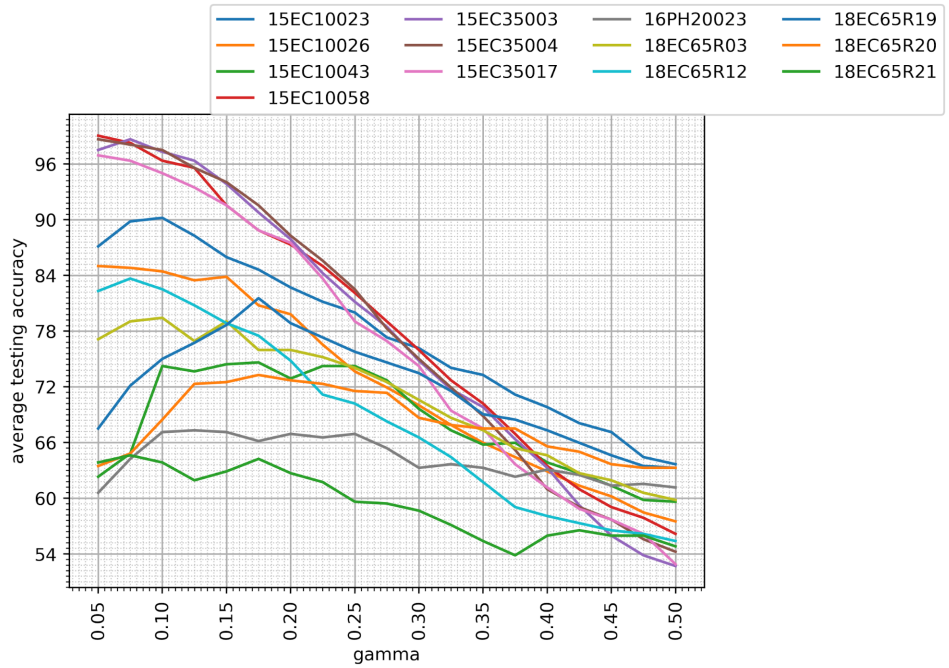

Figure 12: Variation with γ for $\nu = 10^{-3}$

8.3.3 Variation with ν for $\gamma = 0.1$

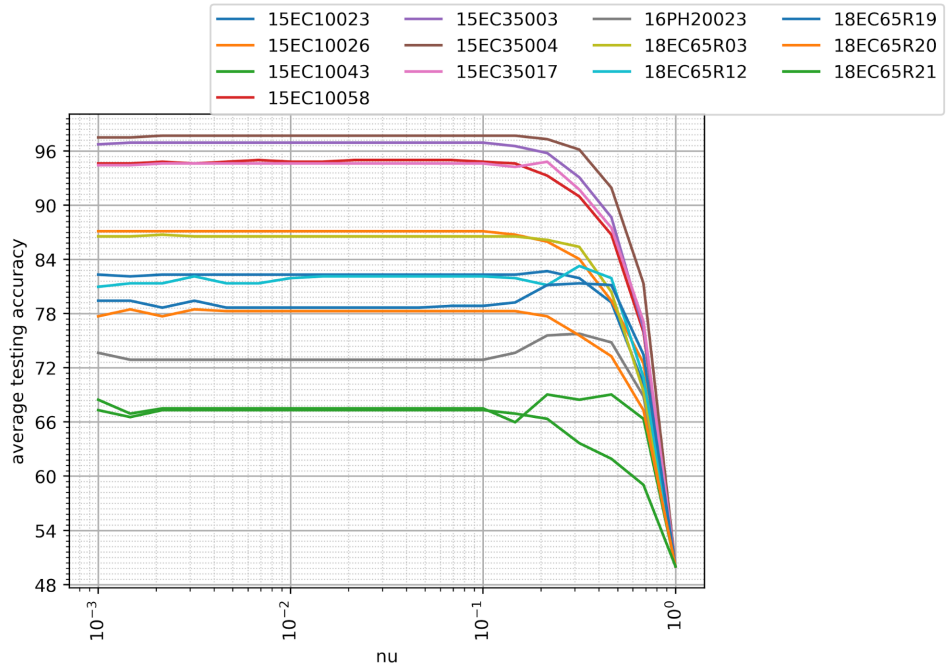

Figure 13: Variation with ν for $\gamma = 0.1$

8.4 Average Precision

Variation with γ and ν. First figure corresponds to Average Precision vs ν for $\gamma = 0.1$ while the second to Average Precision vs γ for $\nu = 10^{-3}$

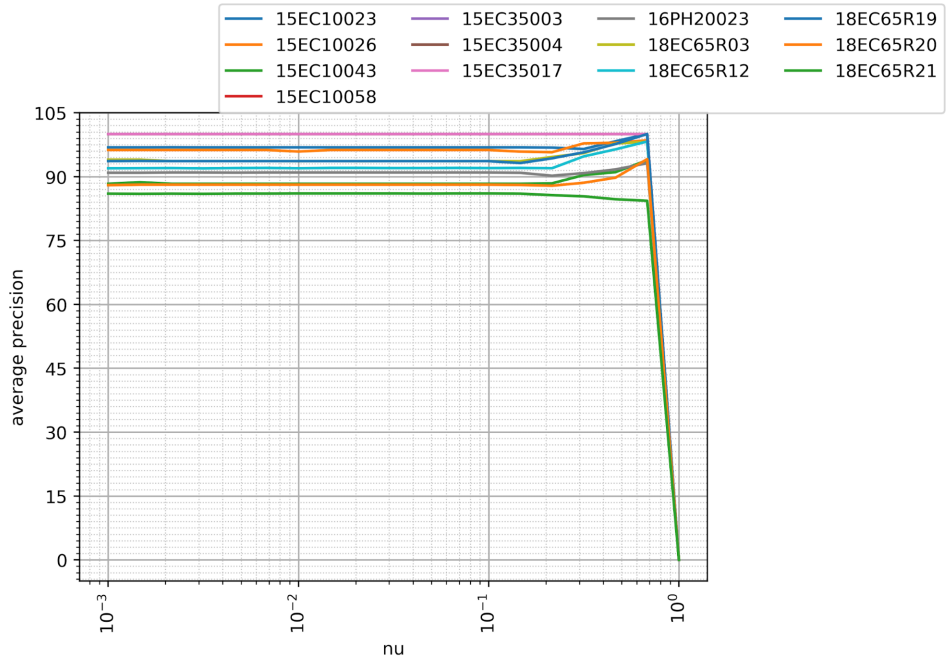

Figure 14: Average Precision vs ν for $\gamma = 0.1$

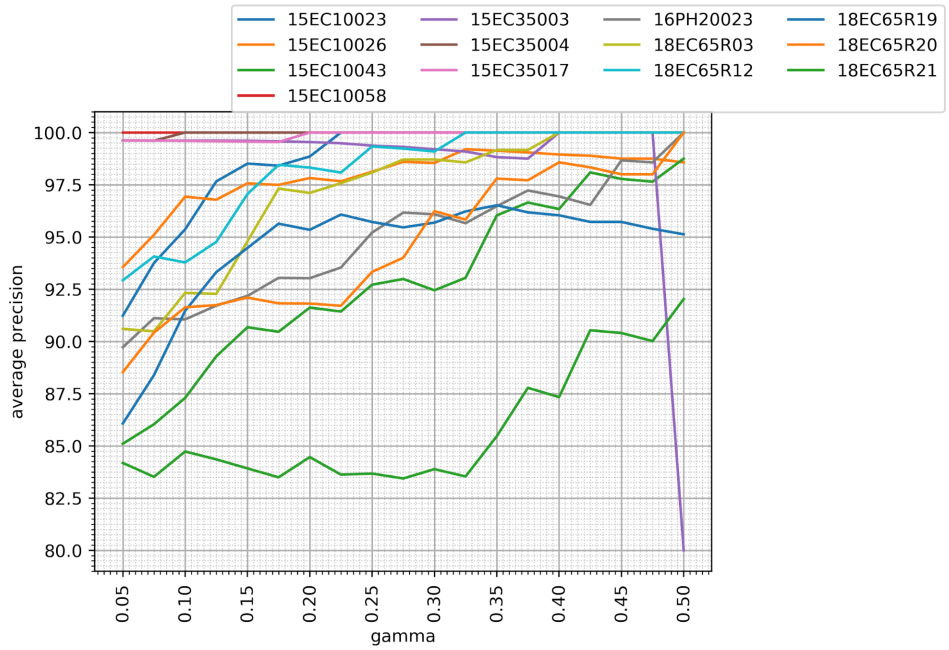

Figure 15: Average Precision vs γ for $\nu = 10^{-3}$

The precision fluctuates with gamma but decreases with nu.

8.5 Average Recall

Variation with γ and ν. First figure corresponds to Average Recall vs ν for $\gamma = 0.1$ while the second to Average Recall vs γ for $\nu = 10^{-3}$

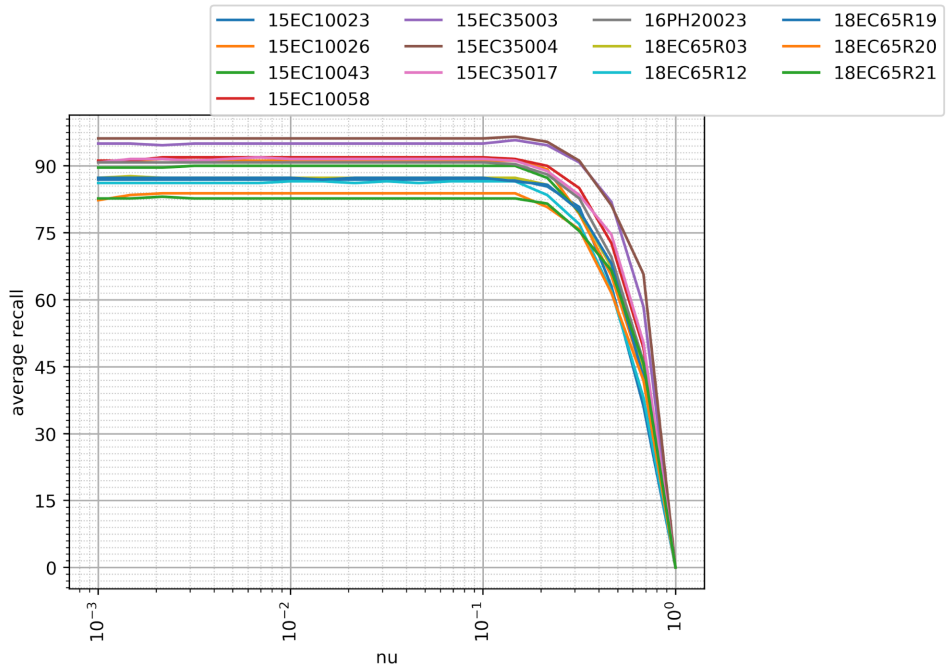

Figure 16: Average Recall vs ν for $\gamma = 0.1$

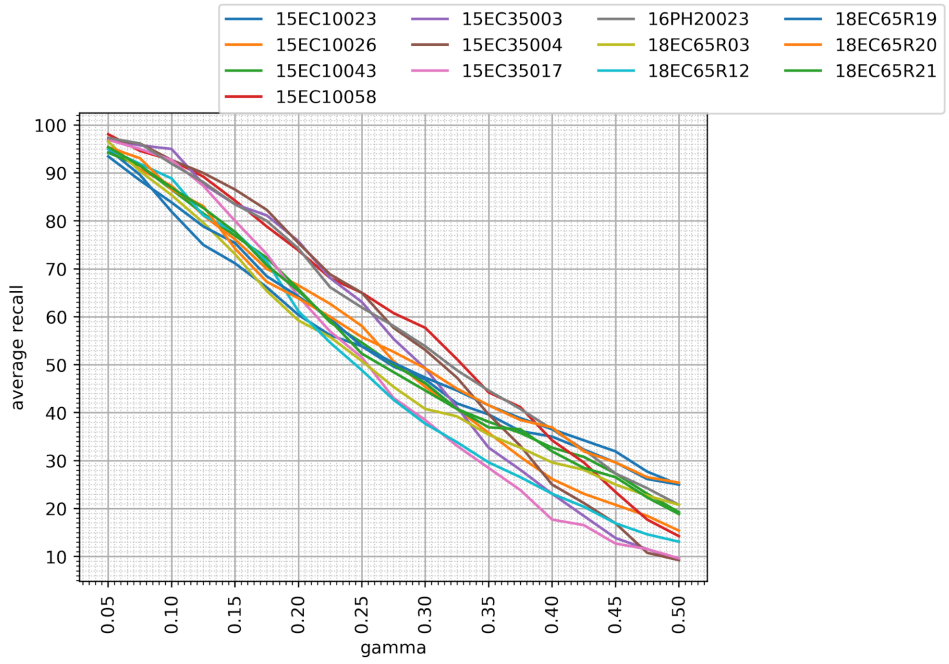

Figure 17: Average Recall vs γ for $\nu = 10^{-3}$

The precision fluctuates with γ but decreases with ν.

9 Conclusion

We achieved more than 60% accuracy for all 13 users whose data we used in this project, using our method of feature selection, feature processing, and the SVM. We obtained an accuracy of near 100% for 4 users, and more than 80% for 9 users. Our analysis also revealed that lower values of ν and γ give better results, and that the choice of parameters heavily influences the outcomes.