Assignment 3 Report

Data Mining

CSE 572

Spring 2018

Submitted to:

Professor Ayan Banerjee

Ira A. Fulton School of Engineering

Arizona State University

Submitted by:

Ananta Soneji (1213090617, asoneji@asu.edu)

Malav Shah (1213199778, mpshah5@asu.edu)

Sarthak Khanna (1211255640, sarthak.khanna@asu.edu)

Shachi Shah (1213185244, skshah11@asu.edu)

Shaishavkumar Jogani (1212392985, sjogani@asu.edu)

April 12, 2018

Table of Contents

I. INTRODUCTION

In Phase 2, we extracted the features from the users' hand signs and reducing the feature space and selected only those features which gave maximum distance between two class. We use the results of phase 2 to further investigate user dependence on hand actions by training on three different classifiers: Decision tree, Support Vector Machines, and Neural Network. Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each instance in the data. For this phase, we will train binary classifiers on the classes "user action vs non-action" for all users and for all actions. We overcome the class unbalanced problem by not considering accuracy of classifiers but considering their precision, recall and F1 score. In this report, first, we will discuss about how we processed the data to train and test the classifiers followed by description of the classification algorithms. Later, we will compare the results of all three models.

II. PREPROCESSING

For Phase 2, we had selected 8 users to perform feature extraction techniques. We had kept the criteria that if the number of data points for an instance of an action is not between 35 to 55 (i.e. 15 Hz frequency), we will not consider that entire instance file. Hence, there were many actions in a user which had count zero. So for Phase 3, we have taken only seven users. This phase requires us to perform classification techniques to predict the actions of each user. We can do this by dividing the user data into two classes "action" and "non-action". After getting the new feature matrix from performing PCA on the results of Phase 2, we have 10 new csv files, each for one gesture (naming convention "GesturenameNewFeatureMatrix.csv"). As all the user data is stored together in these 10 gesture files, we had to separate data for each-user-each-gesture such that the new data has feature instances which can be further classified as "action" and "non-action". For example, let us say we want to make the class file for user1 and gesture-about, so we would take the feature instances of only user1 from "AboutNew FeatureMatrix.csv", whose class will be 1 ("ABOUT") and all the other gestures' data for user1, whose class will be 0 ("NON-ABOUT"). This NON-ABOUT class includes the feature instances of a particular user for all the gestures apart from ABOUT. This is how we use the previously generated files for Phase 3. Now, once we have each-user-each-gesture data, we need to just separate it into training and testing data with 60% being the training data and the rest for testing.

III. ALGORITHMS

A) Decision Tree

- Decision tree is a type of graph that uses branching to reach to every possible value in the outcome. It is a rule based classifier and it generates rules to classify the data.

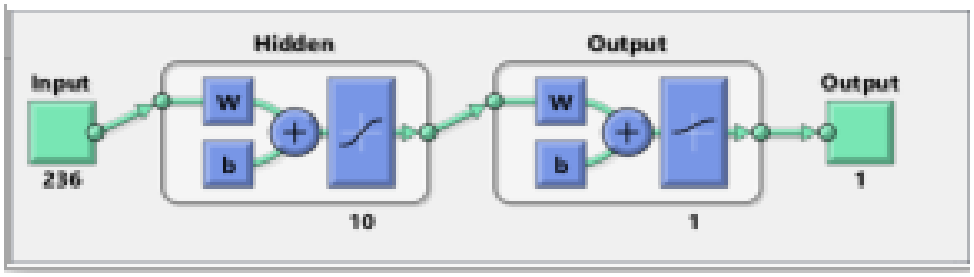
- Decision tree has one root node, from where it starts the branching on the bases of basis of features.
- Each node in a tree acts as a rule and checks the corresponding constraints and passes the data to the corresponding child node.
- The process continues till it reaches the leaf node.

B) Support Vector Machine

- SVMs are a supervised learning model.
- It is maximum margin classifier. It tries to find a classifier that maximizes the margin for the given data.
- SVM is a linear classifier but with some modifications we make it work as a nonlinear classifier.
- While generating a decision boundary for a nonlinear classification task it converts the data to a higher dimension where data is linearly separable and hence find a linear boundary in that dimension. This decision boundary works as a nonlinear decision boundary in the original dimension.
- To find an optimal classifier it tries to maximize the distance between the two nearest points if the opposite classes.

C) Neural Network

- Feedforward neural network are also known as deep feedforward network or multilayer perceptrons.
- Feedforward neural network's goal is to approximate the function by adding some parameters to the original function.
- It is important to have a limited number of training data as it tends to overfit the training data.
- Feedforward neural network has a hidden layer in between the input and the output. Here, we have 236 inputs and one output and in between them, we have one hidden layer with 10 perceptrons in it.
- We used patternnet function of Matlab to feedforward neural network.



Feedforward Neural Network Outline

Of the data collected for training, we used 80% of the data to train the neural network and 20% for validating the results of the designed net.

IV. RESULTS

For results, we have calculated Precision, Recall and Formula 1 score for 3 different models for all 10 gestures. To understand these term, we can classify model results with taking 'ABOUT' gestures' data. We can classify obtained result into 4 categories.

True Positive (TP): The model classifies the data as 'ABOUT' and had in fact 'ABOUT' data.
False Negative (FN): The model classifies the data as 'NOT ABOUT' and had in fact 'ABOUT' data.
True Negative (TN): The model classifies the data as 'NOT ABOUT' and had in fact 'NOT ABOUT' data.
False Positive (FP): The model classifies the data as 'NOT ABOUT' and had in fact 'NOT ABOUT' data.

Precision (P): It is the ratio of all the data which is predicted correctly 'ABOUT' data to Total data which is predicted 'ABOUT' data by the model.

$$\text{Precision} = \text{True Positive (TP)} \;/\; \text{True Positive (TP)} + \text{False Positive (FP)}$$

Recall (R): It is the ratio of all the data which is predicted correctly 'ABOUT' data to Total data 'ABOUT' given to model.

$$\text{Recall} = \text{True Positive (TP)} \;/\; \text{True Positive (TP)} + \text{False Negative (FN)}$$

Formula 1 Score (F1): It is the harmonic mean of Precision (P) and Recall (R).

$$\text{F1 Score} = 2 * \text{Precision (P)} * \text{Recall (R)} \;/\; (\text{Precision (P)} + \text{Recall (R)})$$

Or

$$\text{F1 Score} = 2*TP \;/\; (2*TP + FP + FN)$$

Precision, Recall and F1 score were calculated for all the test users and their values / plots are as follows. In the following plots, blue bars represent decision trees, orange bar represents SVM and yellow bar stands for Neural Networks. In the tables DT - Decision Trees, SVM - Support Vector Machine, NN - Neural Network.

1. ‘ABOUT’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	0	0.66667	0.2	0.66667	NaN	0.36842	0.33333
SVM	0.66667	0.18182	0.2	0.88889	1	0.90909	0.5
NN	0.44444	1	0.2	1	1	0.83333	0.4

Table 1. Precision values for ‘ABOUT’ Gesture

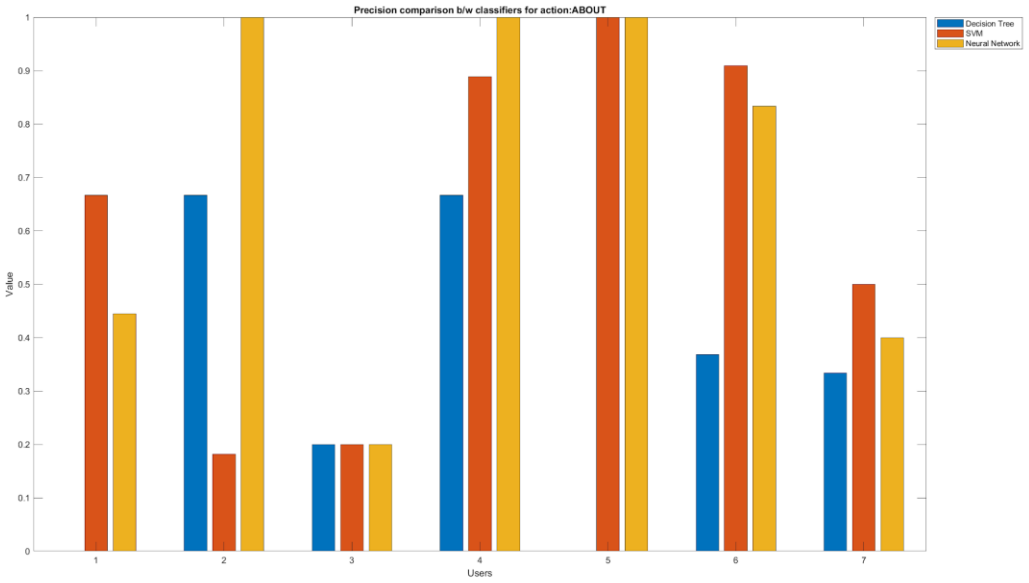


Figure 1. Precision values for ‘ABOUT’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	0	0.25	0.125	0.25	0	0.7	0.25
SVM	0.25	0.25	0.125	1	0.375	1	0.875
NN	0.5	0.25	0.125	0.875	0.625	1	0.25

Table 2. Recall values for 'ABOUT' Gesture

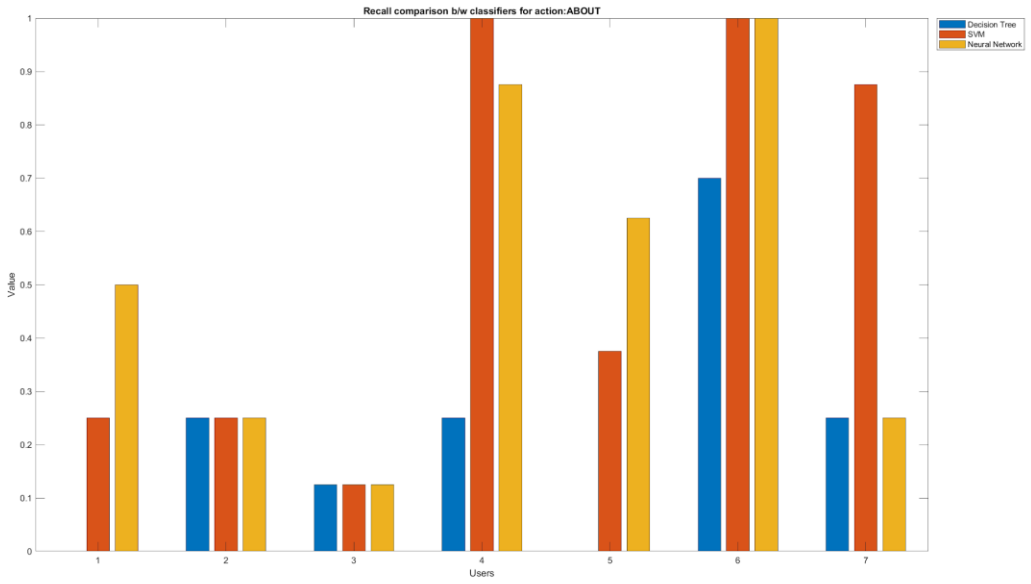


Figure 2. Recall values for 'ABOUT' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0.36364	0.15385	0.36364	NaN	0.48276	0.28571
SVM	0.36364	0.21053	0.15385	0.94118	0.54545	0.95238	0.63636
NN	0.47059	0.4	0.15385	0.93333	0.76923	0.90909	0.30769

Table 3. F1 values for 'ABOUT' Gesture

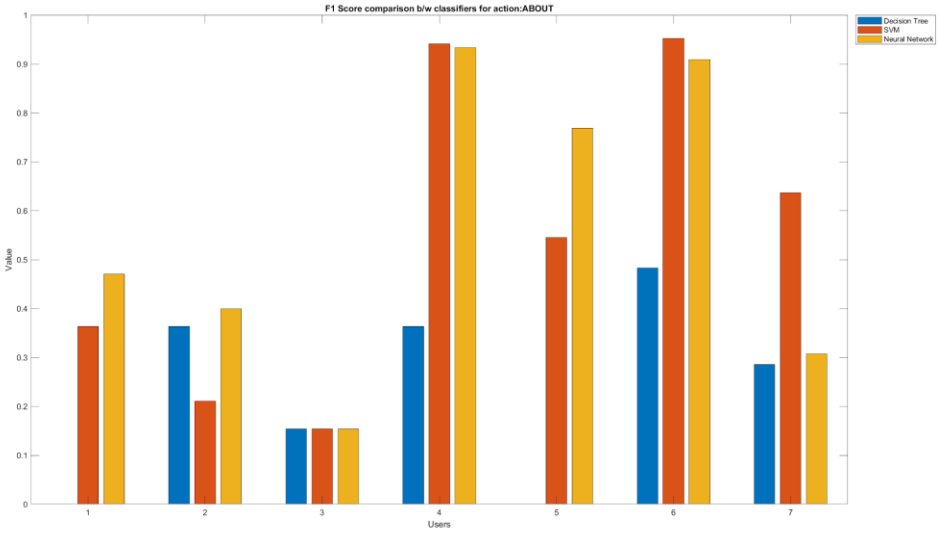


Figure 3. F1 values for 'ABOUT' Gesture

2. ‘AND’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	0.75	0.41667	1	0.5	1	1	1
SVM	0.875	0.53333	1	0.875	0.63636	0.88889	0.5
NN	0.5	0.57143	0.8	0.66667	0.875	0.875	NaN

Table 4. Precision values for ‘AND’ Gesture

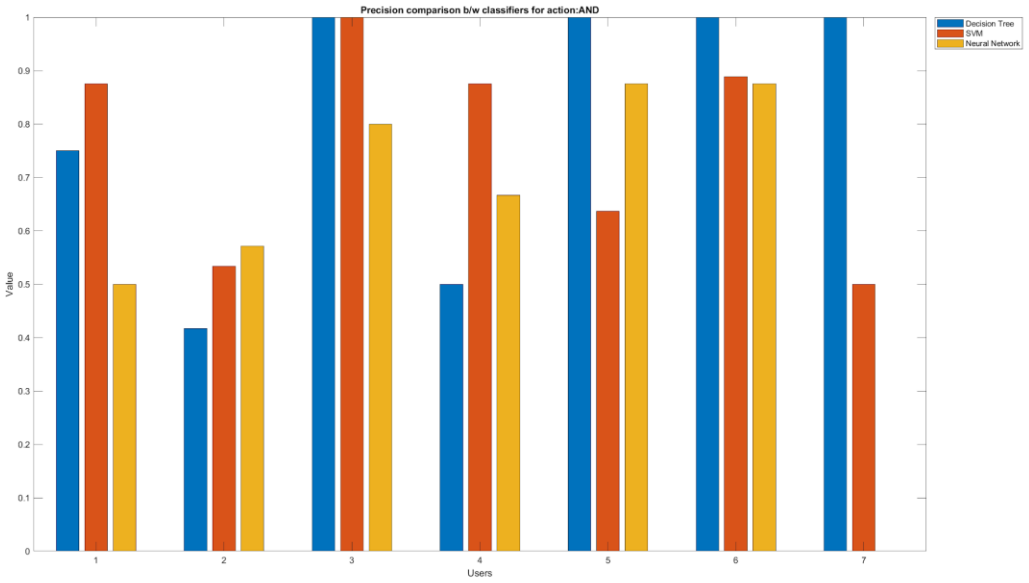


Figure 4. Precision values for ‘AND’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	0.75	0.625	0.75	0.625	0.875	1	0.28571
SVM	0.875	1	1	0.875	0.875	0.88889	0.14286
NN	0.625	1	0.5	0.75	0.875	0.77778	0

Table 5. Recall values for 'AND' Gesture

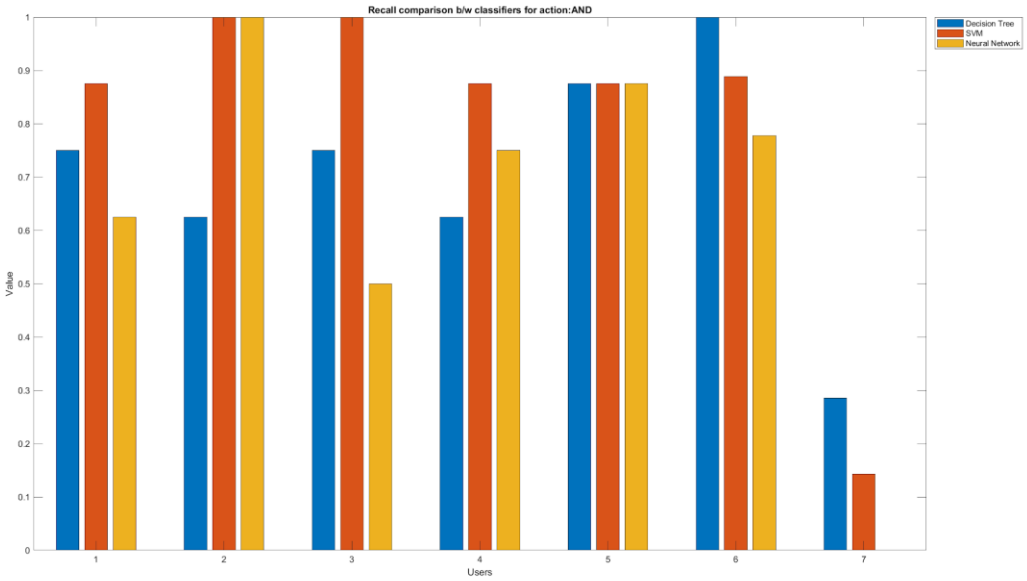


Figure 5. Recall values for 'AND' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	0.75	0.5	0.85714	0.55556	0.93333	1	0.44444
SVM	0.875	0.69565	1	0.875	0.73684	0.88889	0.22222
NN	0.55556	0.72727	0.61538	0.70588	0.875	0.82353	NaN

Table 6. F1 values for 'AND' Gesture

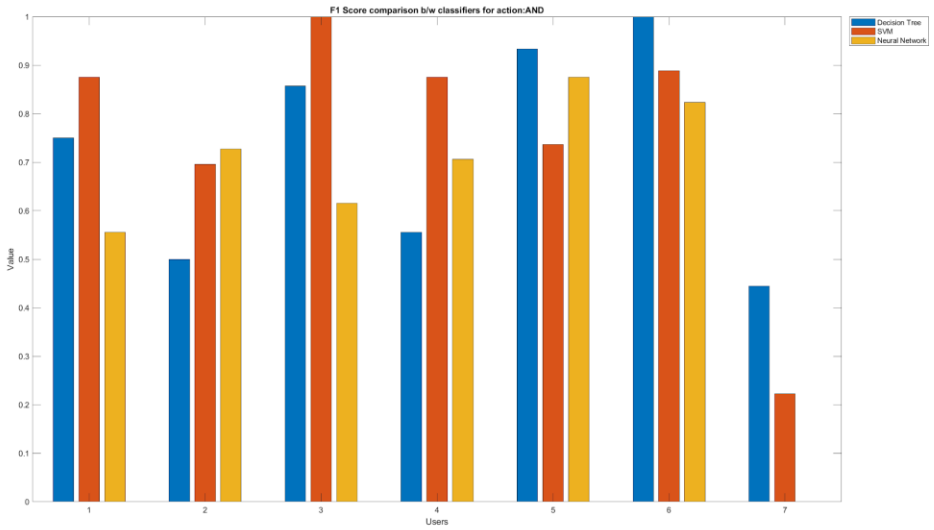


Figure 6. F1 values for 'AND' Gesture

3. ‘CAN’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	0	1	0.66667	1	0.875	NaN	0.42857
SVM	0	1	1	0.8	0.83333	NaN	1
NN	NaN	0.44444	0	0.88889	0.88889	NaN	1

Table 7. Precision values for ‘CAN’ Gesture

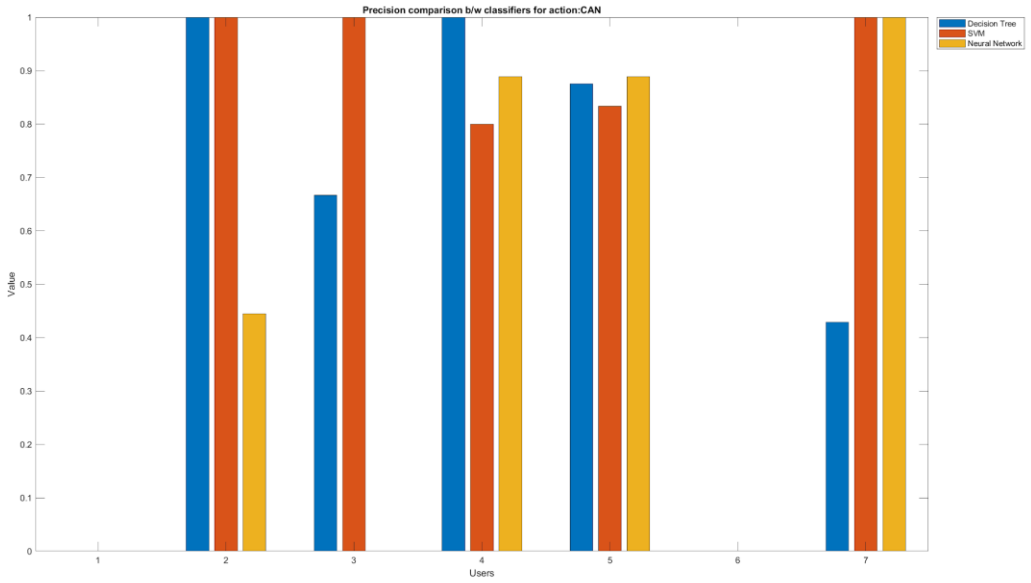


Figure 7. Precision values for ‘CAN’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0.75	0.22222	0.875	0.875	0	0.375
SVM	NaN	0.75	0.77778	1	0.625	0	0.625
NN	NaN	1	0	1	1	0	1

Table 8. Recall values for 'CAN' Gesture

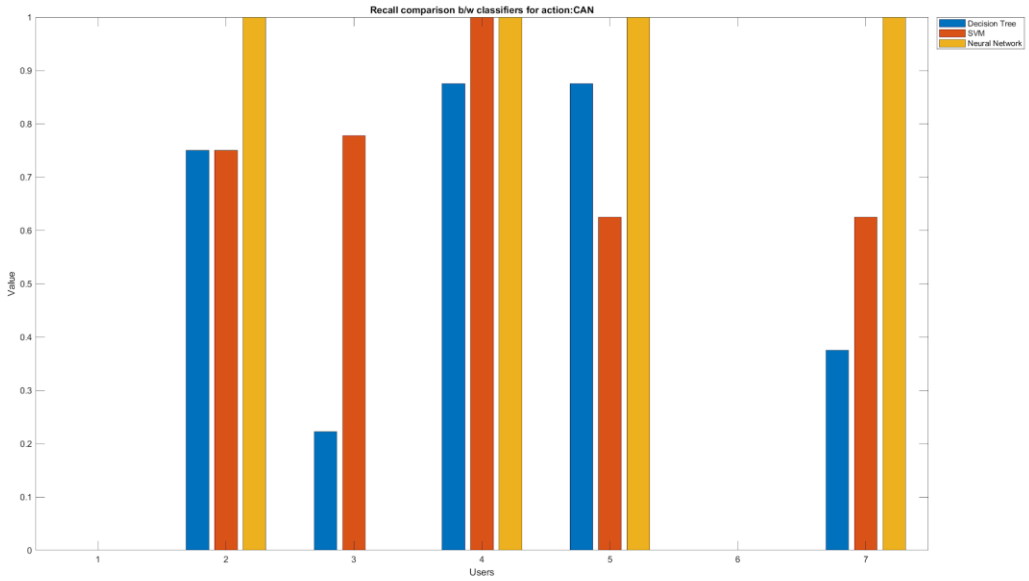


Figure 8. Recall values for 'CAN' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0.85714	0.33333	0.93333	0.875	NaN	0.4
SVM	NaN	0.85714	0.875	0.88889	0.71429	NaN	0.76923
NN	NaN	0.61538	NaN	0.94118	0.94118	NaN	1

Table 9. F1 values for 'CAN' Gesture

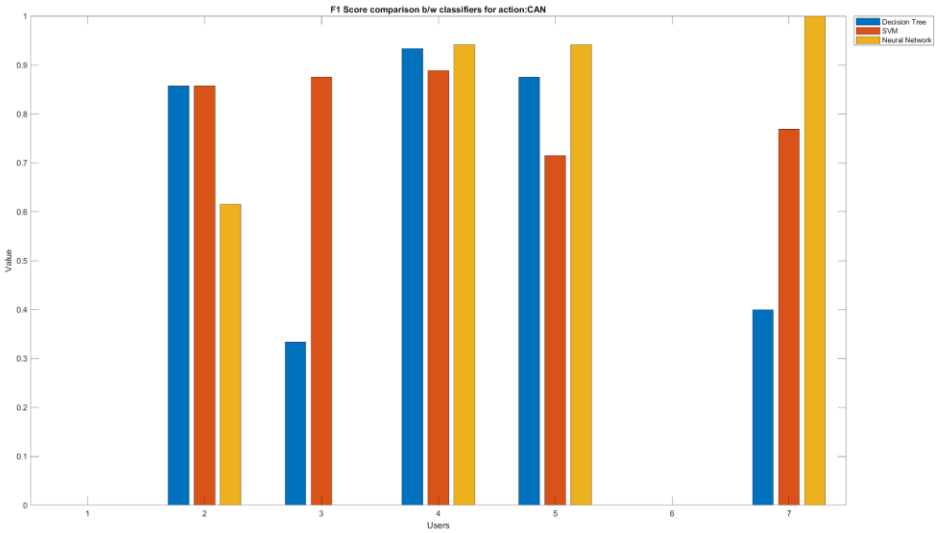


Figure 9. F1 values for 'CAN' Gesture

4. ‘COP’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0	NaN	0	0	0.33333	0.77778
SVM	0	0.33333	0	1	0.1	0.44444	1
NN	0	0.16667	0	0.75	0	0.44444	0.875

Table 10. Precision values for ‘COP’ Gesture

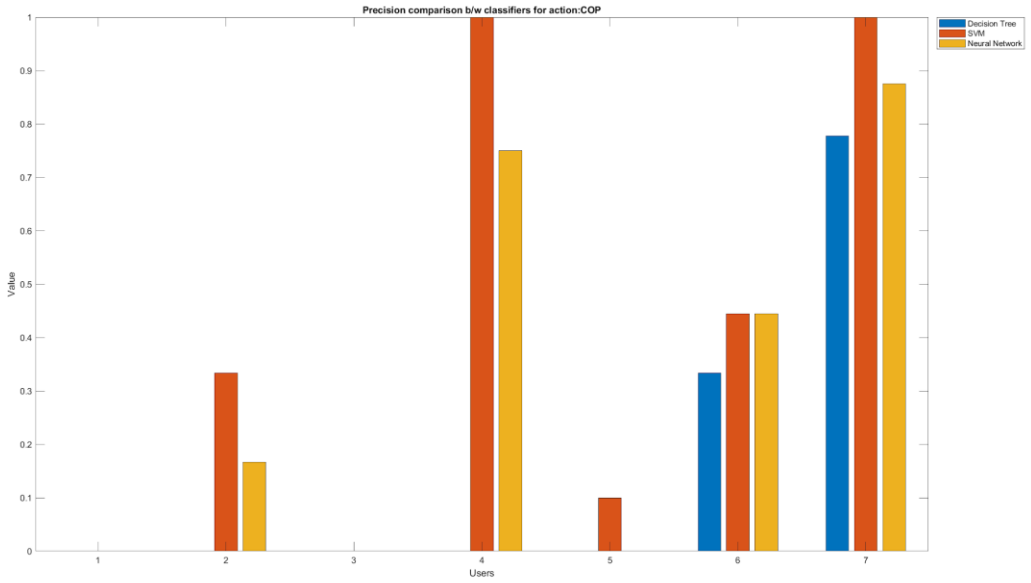


Figure 10. Precision values for ‘COP’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0	0	0	0	0.625	1
SVM	NaN	0.75	0	0.5	0.125	1	1
NN	NaN	0.375	0	0.375	0	1	1

Table 11. Recall values for 'COP' Gesture

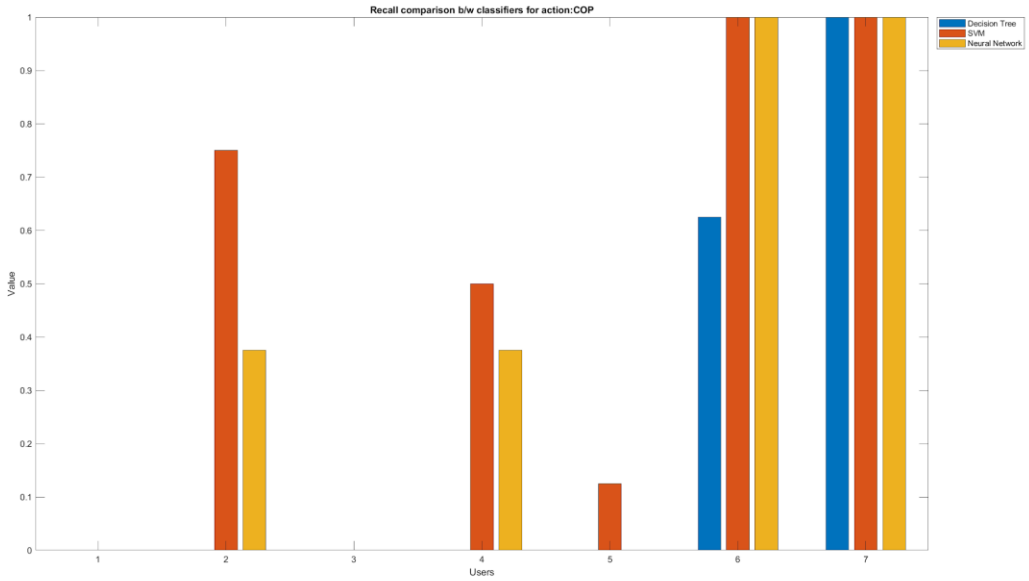


Figure 11. Recall values for 'COP' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	NaN	NaN	NaN	NaN	0.43478	0.875
SVM	NaN	0.46154	NaN	0.66667	0.11111	0.61538	1
NN	NaN	0.23077	NaN	0.5	NaN	0.61538	0.93333

Table 12. F1 values for 'COP' Gesture

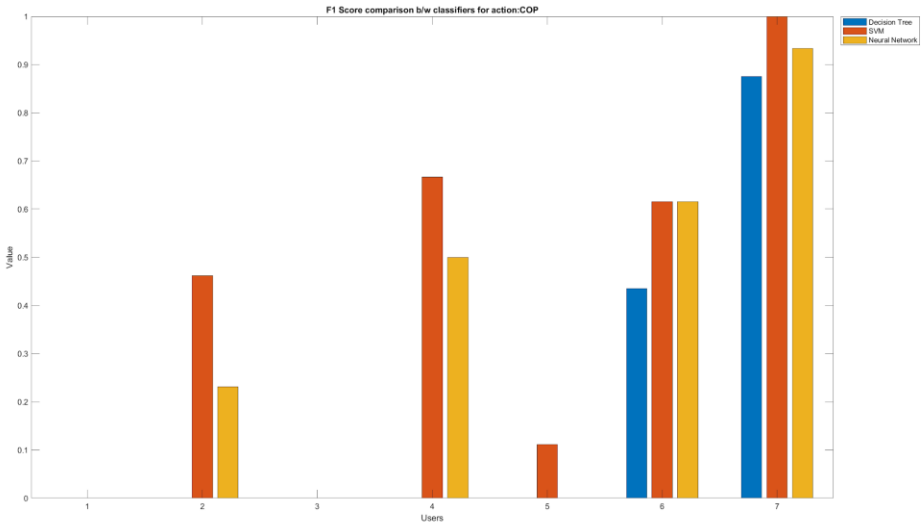


Figure 12. F1 values for 'COP' Gesture

5. ‘DEAF’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	0	0.30435	0.11111	0.33333	1	1	0.77778
SVM	NaN	0.72727	0.72727	1	0.53333	1	0.875
NN	0	0.8	0	0.5	0.7	1	1

Table 13. Precision values for ‘DEAF’ Gesture

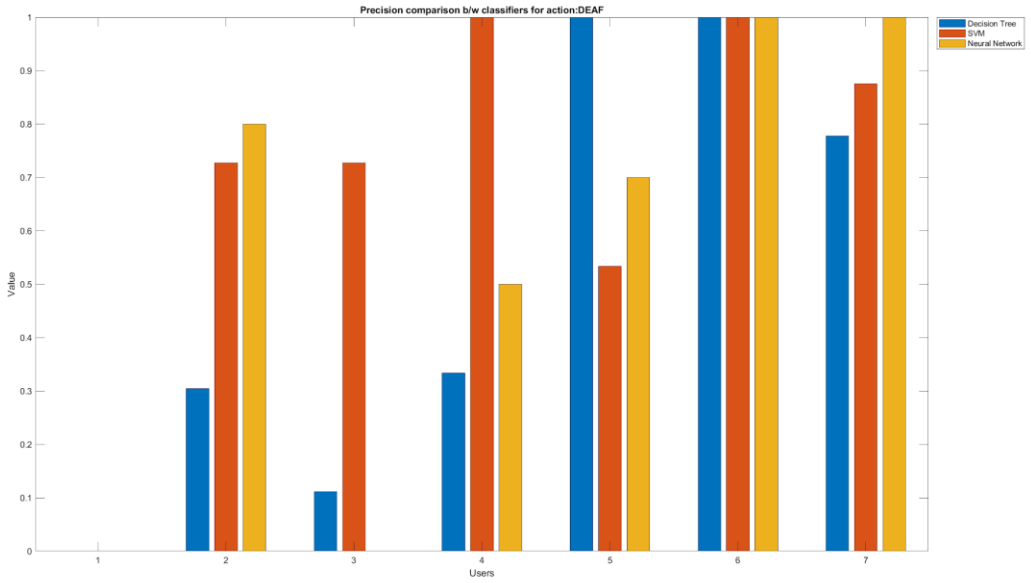


Figure 13. Precision values for ‘DEAF’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0.875	0.125	0.25	0.625	0.625	0.875
SVM	NaN	1	1	0.25	1	1	0.875
NN	NaN	1	0	0.25	0.875	0.875	0.75

Table 14. Recall values for 'DEAF' Gesture

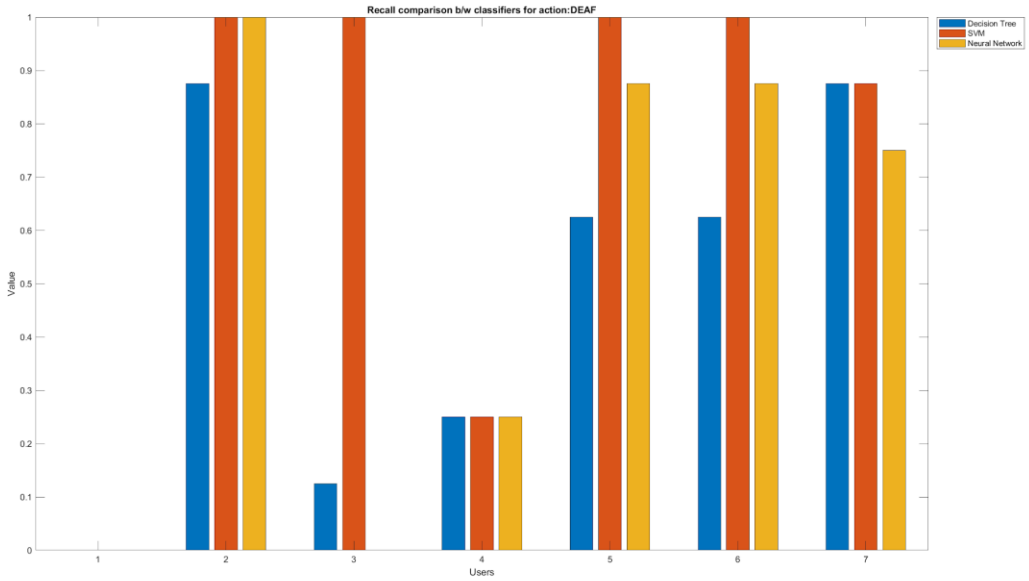


Figure 14. Recall values for 'DEAF' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0.45161	0.11765	0.28571	0.76923	0.76923	0.82353
SVM	NaN	0.84211	0.84211	0.4	0.69565	1	0.875
NN	NaN	0.88889	NaN	0.33333	0.77778	0.93333	0.85714

Table 15. F1 values for 'DEAF' Gesture

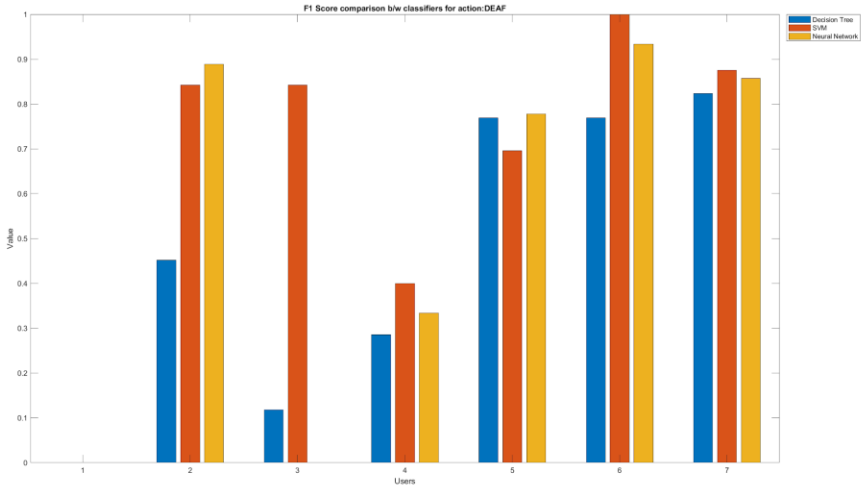


Figure 15. F1 values for 'DEAF' Gesture

6. ‘DECIDE’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0.63636	0.66667	0.71429	0.66667	0.28571	0.66667
SVM	1	1	0.42857	0.8	0.875	0.75	0.8
NN	0	0.85714	1	0.61538	1	0.75	0.66667

Table 16. Precision values for ‘DECIDE’ Gesture

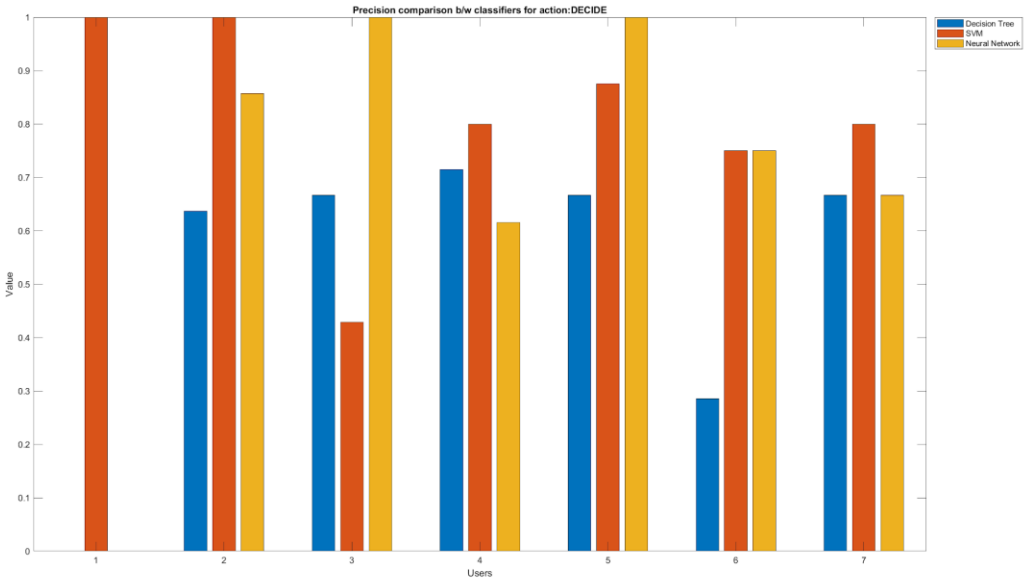


Figure 16. Precision values for ‘DECIDE’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	0	0.875	0.28571	0.625	0.5	0.44444	0.25
SVM	0.5	1	0.42857	1	0.875	0.66667	0.5
NN	0	0.75	0.28571	1	0.75	0.33333	0.25

Table 17. Recall values for 'DECIDE' Gesture

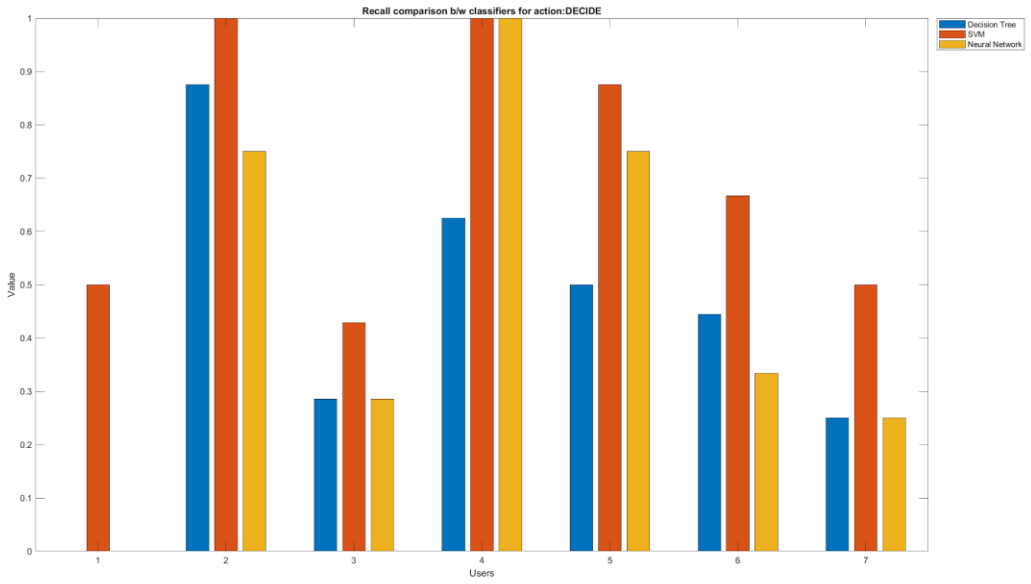


Figure 17. Recall values for 'DECIDE' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0.73684	0.4	0.66667	0.57143	0.34783	0.36364
SVM	0.66667	1	0.42857	0.88889	0.875	0.70588	0.61538
NN	NaN	0.8	0.44444	0.7619	0.85714	0.46154	0.36364

Table 18. F1 values for 'DECIDE' Gesture

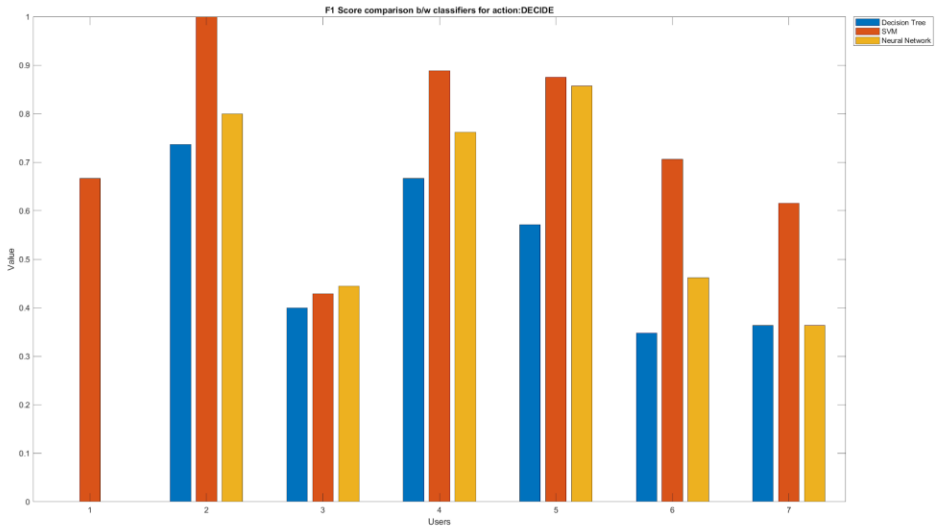


Figure 18. F1 values for 'DECIDE' Gesture

7. ‘FATHER’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	1	0.5	0.77778	0.63636	0.625	0.33333	1
SVM	0.46667	0.875	0.53846	0.63636	0.83333	0.5	1
NN	0.46667	0.42857	0.63636	0.63636	0.875	0.66667	1

Table 19. Precision values for ‘FATHER’ Gesture

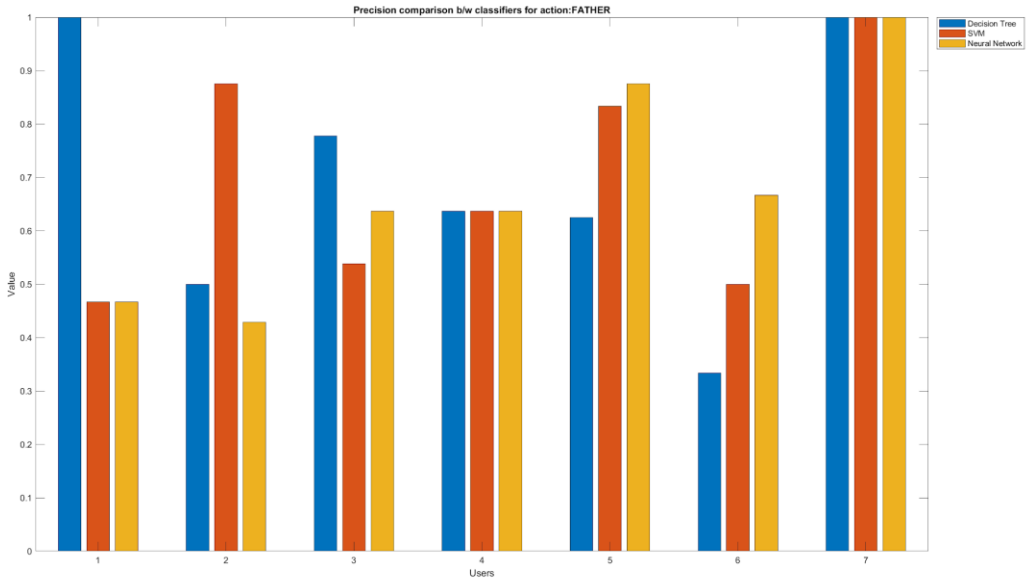


Figure 19. Precision values for ‘FATHER’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	0.875	1	1	0.875	0.625	0.25	0.625
SVM	0.875	1	1	0.875	0.625	0.375	0.75
NN	0.875	0.42857	1	0.875	0.875	0.75	0.875

Table 20. Recall values for 'FATHER' Gesture

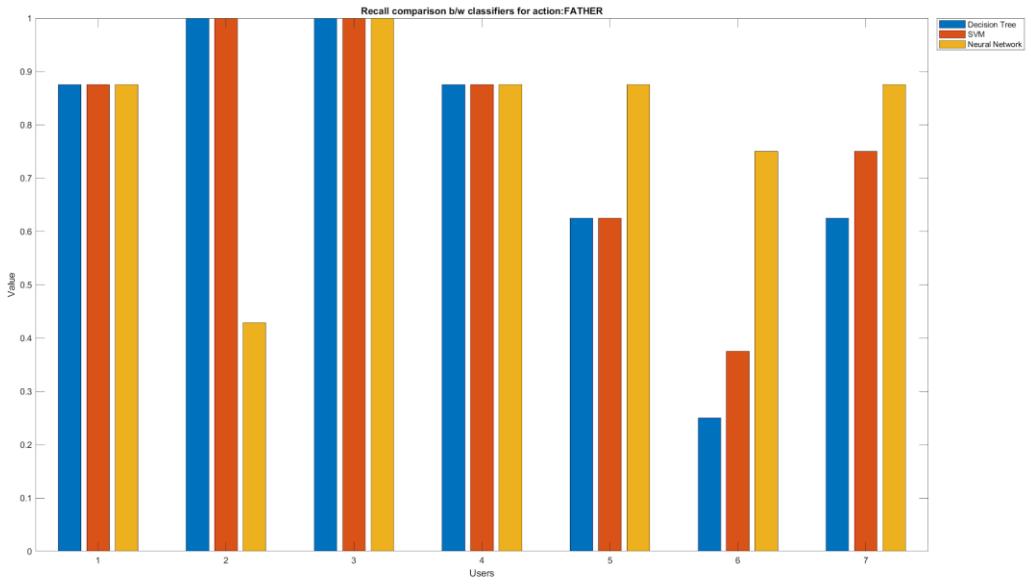


Figure 20. Recall values for 'FATHER' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	0.93333	0.66667	0.875	0.73684	0.625	0.28571	0.76923
SVM	0.6087	0.93333	0.7	0.73684	0.71429	0.42857	0.85714
NN	0.6087	0.42857	0.77778	0.73684	0.875	0.70588	0.93333

Table 21. F1 values for 'FATHER' Gesture

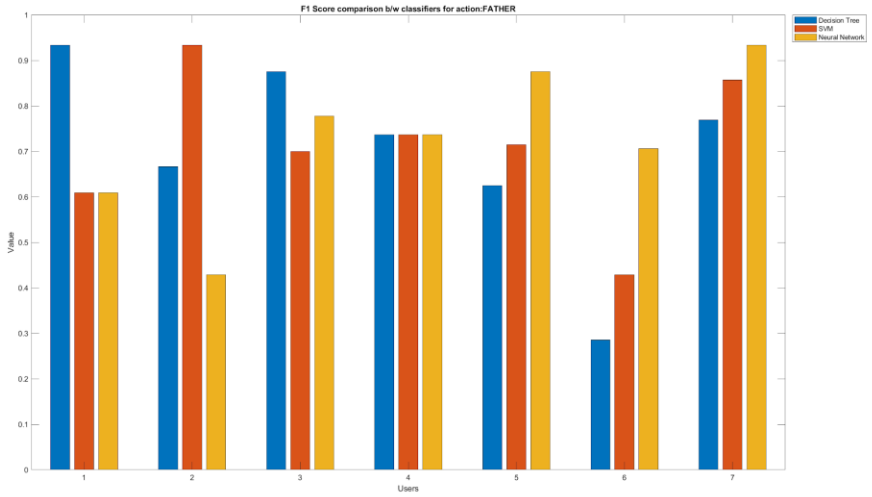


Figure 21. F1 values for 'FATHER' Gesture

8. ‘FIND’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	0.31579	0.88889	0	0.4	0.83333	0.35294	0.88889
SVM	0.33333	0.88889	NaN	0.57143	1	0.44444	0.77778
NN	0.5	0.88889	0.5	0.57143	1	1	0.88889

Table 22. Precision values for ‘FIND’ Gesture

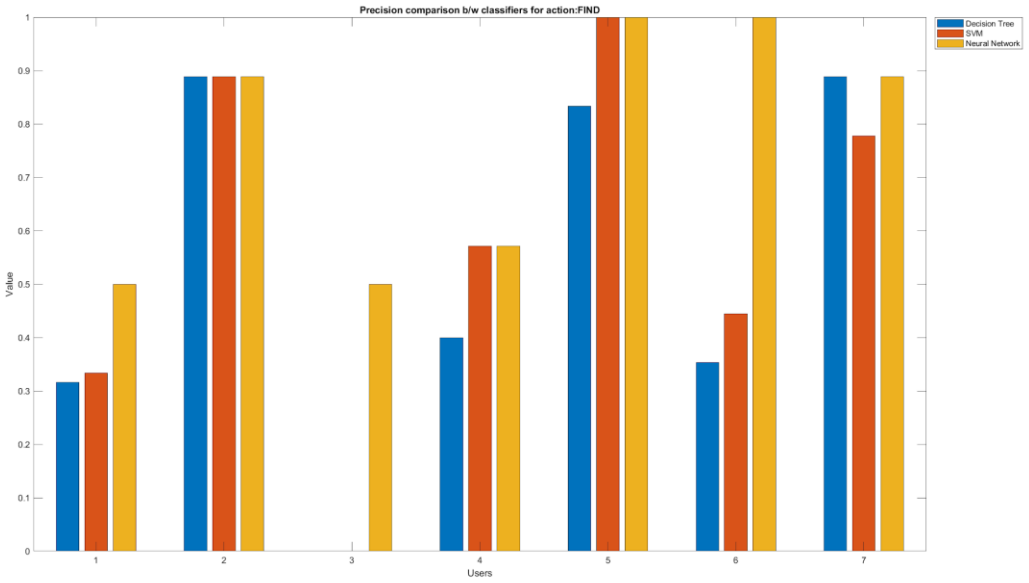


Figure 22. Precision values for ‘FIND’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	0.75	1	0	0.75	0.625	0.66667	1
SVM	0.75	1	0	1	1	0.88889	0.875
NN	0.875	1	0.125	1	0.75	0.77778	1

Table 23. Recall values for 'FIND' Gesture

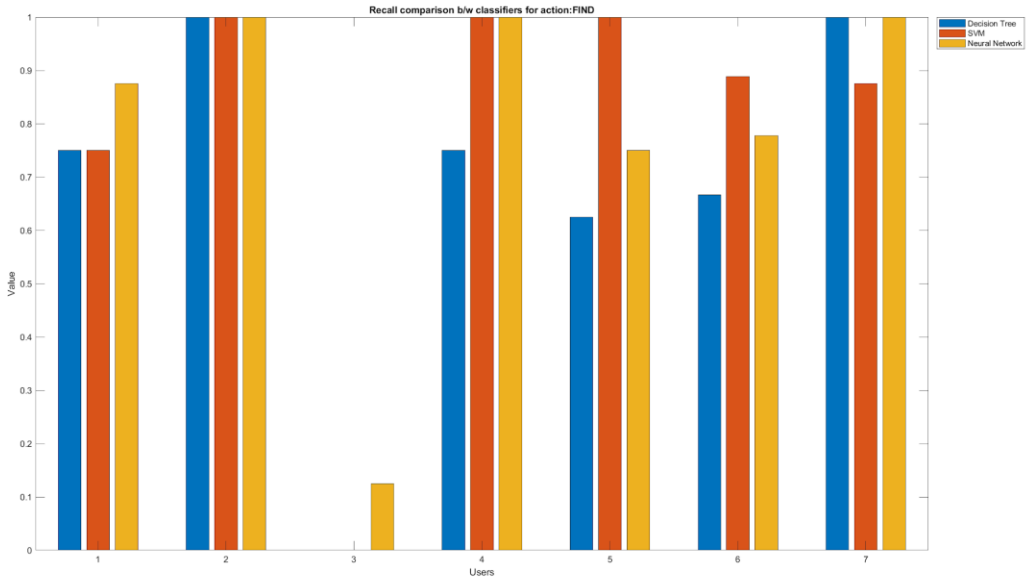


Figure 23. Recall values for 'FIND' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	0.44444	0.94118	NaN	0.52174	0.71429	0.46154	0.94118
SVM	0.46154	0.94118	NaN	0.72727	1	0.59259	0.82353
NN	0.63636	0.94118	0.2	0.72727	0.85714	0.875	0.94118

Table 24. F1 values for 'FIND' Gesture

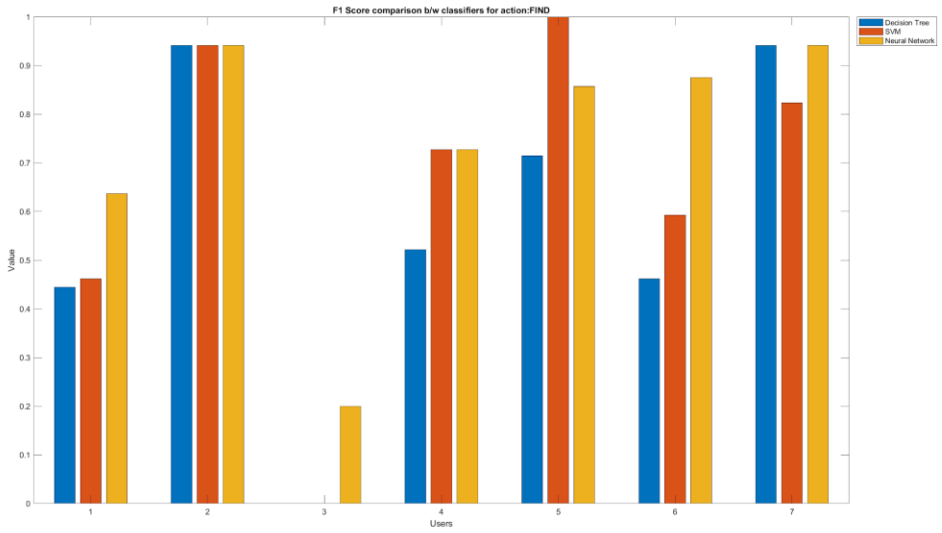


Figure 24. F1 values for 'FIND' Gesture

9. ‘GO OUT’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	0.42105	0.71429	0.2	0.5	0.66667	0.6	0.57143
SVM	0.42105	0	0.8	1	0.88889	1	0.75
NN	0.4	0.45455	0.875	0.66667	0.8	1	NaN

Table 25. Precision values for ‘GO OUT’ Gesture

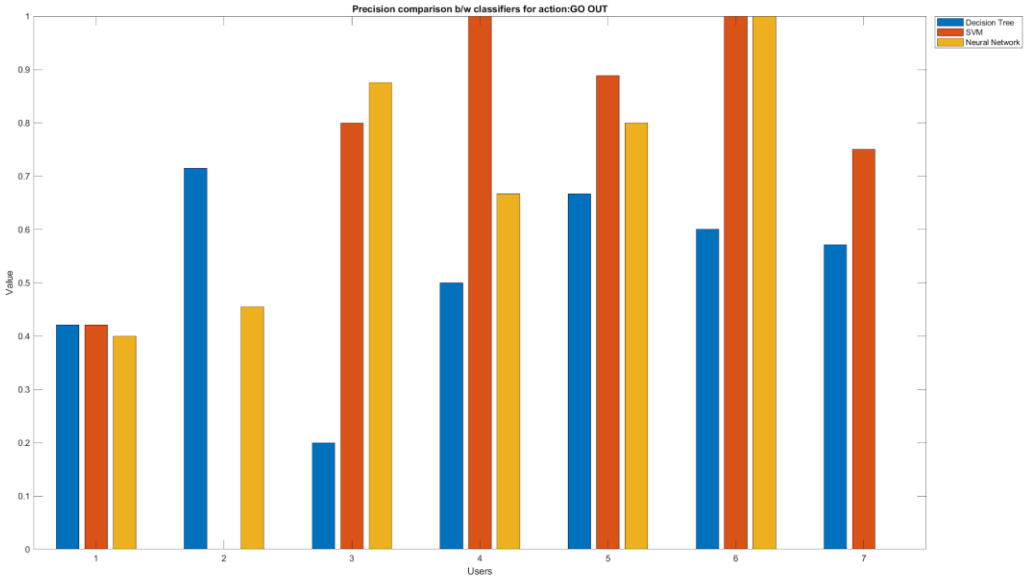


Figure 25. Precision values for ‘GO OUT’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	1	0.625	0.125	0.25	1	0.375	1
SVM	1	0	1	1	1	1	0.75
NN	0.75	0.625	0.875	0.25	1	0.375	0

Table 26. Recall values for 'GO OUT' Gesture

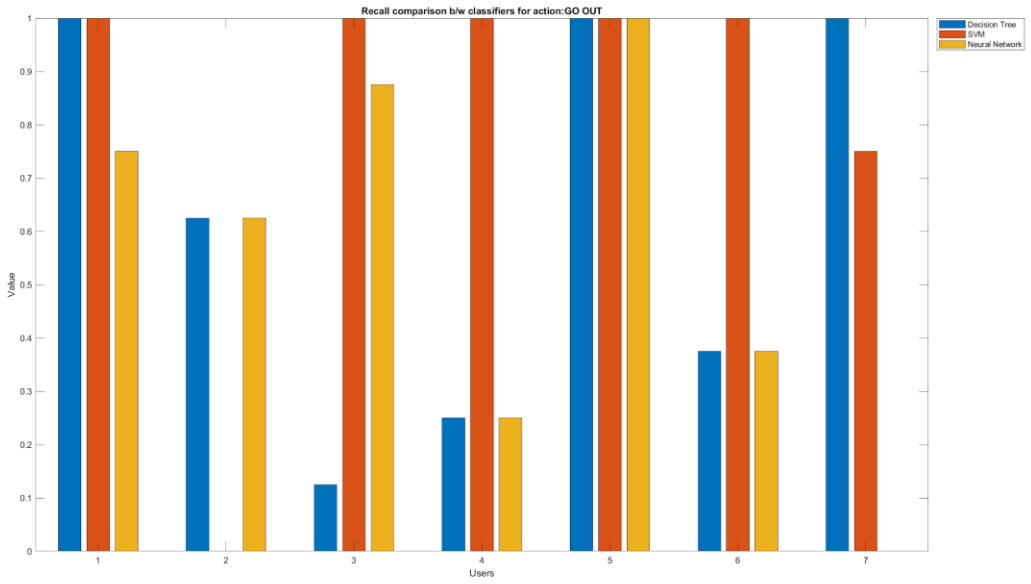


Figure 26. Precision values for 'GO OUT' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	0.59259	0.66667	0.15385	0.33333	0.8	0.46154	0.72727
SVM	0.59259	NaN	0.88889	1	0.94118	1	0.75
NN	0.52174	0.52632	0.875	0.36364	0.88889	0.54545	NaN

Table 27. F1 values for 'GO OUT' Gesture

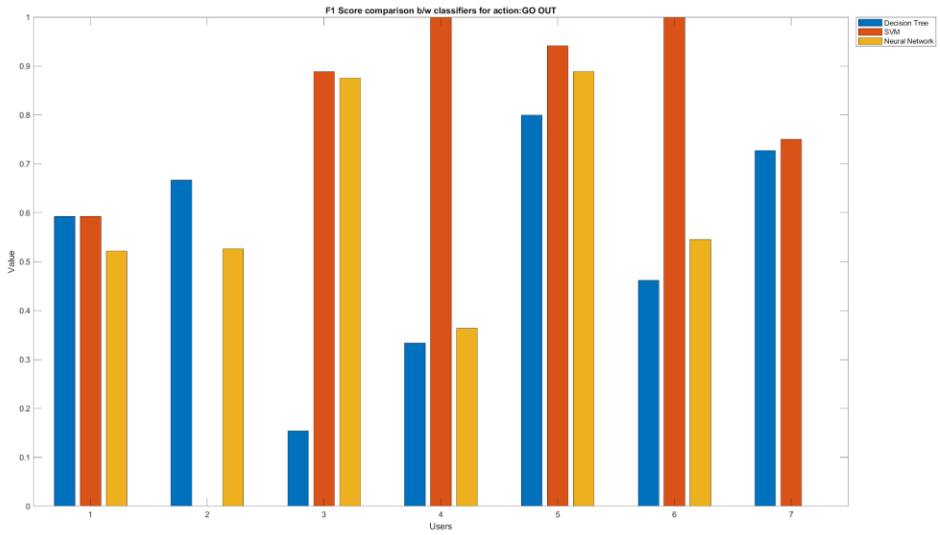


Figure 27. F1 values for 'GO OUT' Gesture

10. ‘HEARING’ Gesture

Precision

	User1	User2	User3	User4	User5	User6	User7
DT	0	0.63636	0.7	0.88889	0.41667	0.6	0.5
SVM	0	1	0.875	0.8	0.61538	0.88889	0.53333
NN	0	0.5	1	0.8	0.66667	0.71429	0.33333

Table 28. Precision values for ‘HEARING’ Gesture

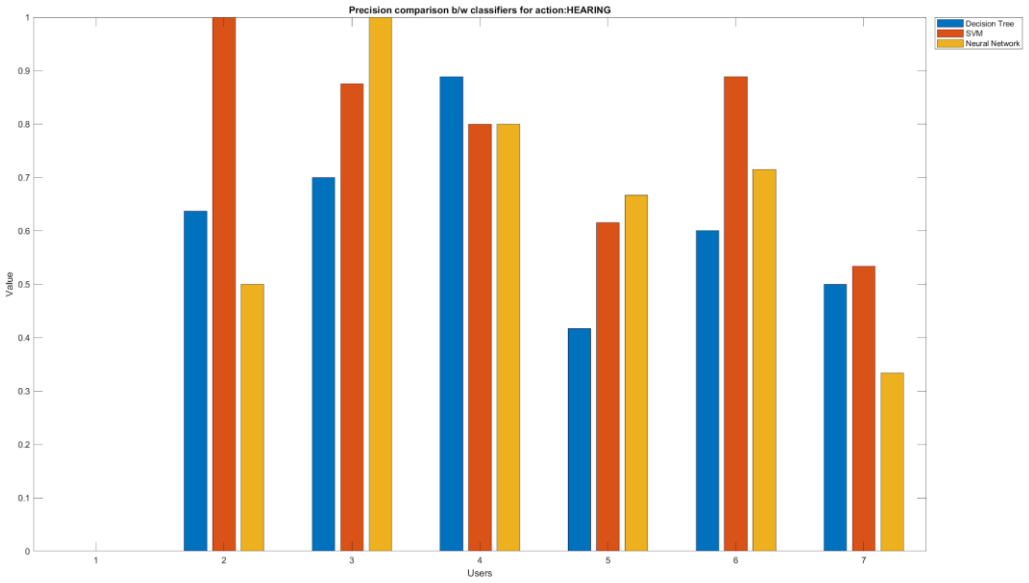


Figure28. Precision values for ‘HEARING’ Gesture

Recall

	User1	User2	User3	User4	User5	User6	User7
DT	0	0.875	0.875	1	0.625	0.75	0.625
SVM	0	0.25	0.875	1	1	1	1
NN	0	0.25	0.625	1	1	0.625	0.25

Table 29. Recall values for 'HEARING' Gesture

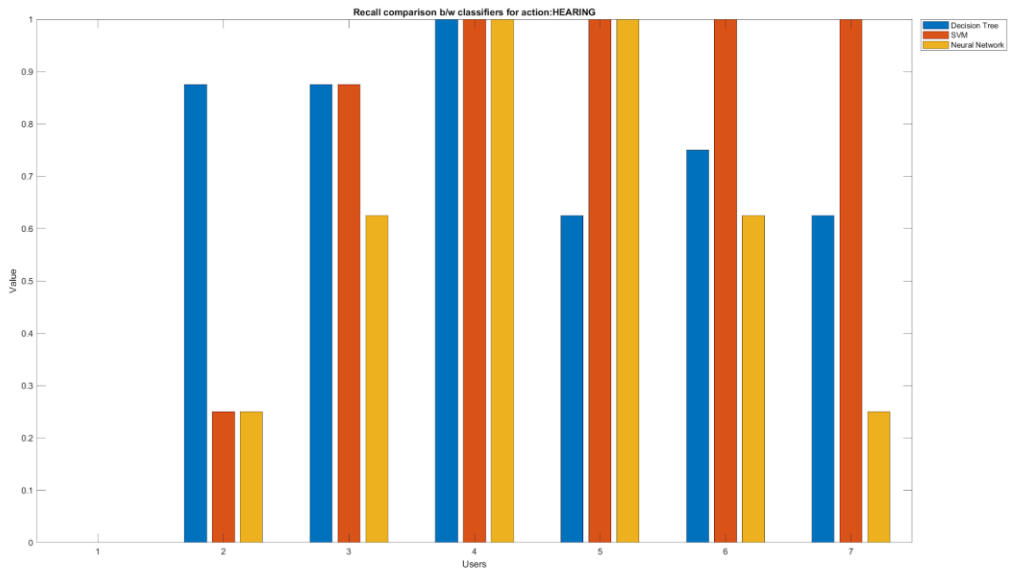


Figure 29. Recall values for 'HEARING' Gesture

Formula 1 (F1)

	User1	User2	User3	User4	User5	User6	User7
DT	NaN	0.73684	0.77778	0.94118	0.5	0.66667	0.55556
SVM	NaN	0.4	0.875	0.88889	0.7619	0.94118	0.69565
NN	NaN	0.33333	0.76923	0.88889	0.8	0.66667	0.28571

Table 30. F1 values for 'HEARING' Gesture

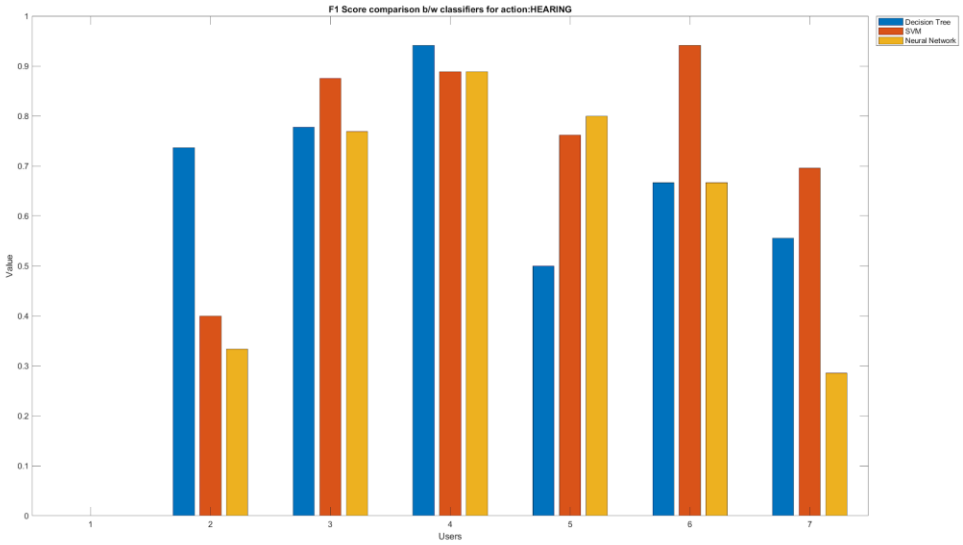


Figure 30. F1 values for 'HEARING' Gesture

V. CONCLUSION

- From the plots, we could see that overall, the results of SVM are better as compared to Decision Trees and Neural Network.
- **Decision trees** are relatively easy to understand only when there are fewer decisions to make. Here, we are having 236 features for each instance which makes the tree large and dozens of decision nodes. The more decisions there are in a tree, the less accurate any expected outcomes are likely to be.
- **SVMs** are a set of supervised learning methods primarily used in classification, pattern recognition, and regression. For binary classification, the idea behind SVMs is to separate the data in some "optimal" method. SVMs are meant to perform binary classification more accurately because of the way it creates the hyperplane to discriminate classes.

- **Neural Networks** are not probabilistic. They have the potential to give accurate results though less data for training, class imbalance, and less dense network makes it inefficient in this classification problem.