GOUTAM DAS

CLASSIFYING USERS ON THE BASIS OF KEYSTROKE DYNAMICS

UNDER THE GUIDANCE OF
PROF. PUSPANJALI MOHAPATRA

in partial fulfillment for the award of the degree of Bachelor in Technology in Computer Science & Engineering

Department of Computer Science & Engineering International Institute of Information Technology Bhubaneswar, India

ABSTRACT

Legitimate user authentication is an important part of the problem related to the computer and system security. The maintenance of security becomes even more difficult when an invalid user gets the system access information. The variables that help make a handwritten signature a unique human identifier also provides a unique digital signature in the form of a stream of latency periods between keystrokes.

Keystroke dynamics - the analysis of typing rhythms to discriminate among users has been proposed for detecting impostors (i.e., both insiders and external attackers). Such a system will determine if the current user is the genuine one or not. The approaches use typing biometrics of a user during login to identify a user. We used two data-sets, a new one consisting of 12 subjects which we created(IIITBh dataset) and another of 51 subjects(CMU dataset), each one typing a typical, strong password of 10 characters multiple times over multiple sessions. This thesis presents a suite of techniques and their analysis for password authentication using neural networks and SVM.

ACKNOWLEDGEMENTS

First of all, I wish to thank our project guide Prof. Puspanjali Mohapatra for her invaluable guidance in this project, the detailed explanations, the patience and the precision in the suggestions. Second, I wish to thank my team members for their competence and kindness. My sincere thanks to all the people who have contributed to this thesis through their research and knowledge.

CONTENTS

1 INTRODUCTION

Biometrics is the science of establishing the identity of an individual based on the physical, chemical or behavioral attributes of the person. The relevance of biometrics in modern society has been reinforced by the need for large-scale identity management systems whose functionality relies on the accurate determination of an individual's identity in the context of several different applications.[Anil K. Jain and Ross, 2008]

Compromised passwords and shared accounts are frequently exploited by both external attackers and insiders. External attackers test whether accounts use default or common passwords. Insiders compile lists of shared or compromised passwords for later use, e.g., to damage the system in the event that they are fired or demoted. If we had some means, other than knowledge of a password, with which to identify exactly who is logging into an account, and to discriminate between the genuine user of an account and an impostor, we could significantly curb these security threats.

1.1 CHARACTERISTICS OF BIOMETRIC

A number of biometric characteristics are being used in various applications. Each biometric has its pros and cons and, therefore, the choice of a biometric trait for a particular application depends on a variety of issues besides its matching performance (Table 1.2). Jain et al.[S. Pankanti, 1999] have identified seven factors that determine the suitability of a physical or a behavioural trait to be used in a biometric application.

- Universality: Every individual accessing the application should possess the trait.
- Uniqueness: The given trait should be sufficiently different across individuals comprising the population.
- Permanence: The biometric trait of an individual should be sufficiently invariant over a period of time with respect to the matching algorithm. A trait that changes significantly over time is not a useful biometric.
- Measurability: It should be possible to acquire and digitize the biometric trait using suitable devices that do not cause undue inconvenience to the individual. Furthermore, the acquired raw data should be amenable to processing in order to extract representative feature sets.
- Performance: The recognition accuracy and the resources required to achieve that accuracy should meet the constraints imposed by the application.

- Acceptability: Individuals in the target population that will utilize the application should be willing to present their biometric trait to the system.

- Circumvention: This refers to the ease with which the trait of an individual can be imitated using artefacts (e.g., fake fingers), in the case of physical traits, and mimicry, in the case of behavioral traits. No single biometric is expected to effectively meet all the requirements (e.g., accuracy, practicality, cost) imposed by all applications (e.g., Digital Rights Management (DRM), access control, welfare distribution). In other words, no biometric is ideal but a number of them are admissible. The relevance of a specific biometric to an application is established depending upon the nature and requirements of the application, and the properties of the biometric characteristic.

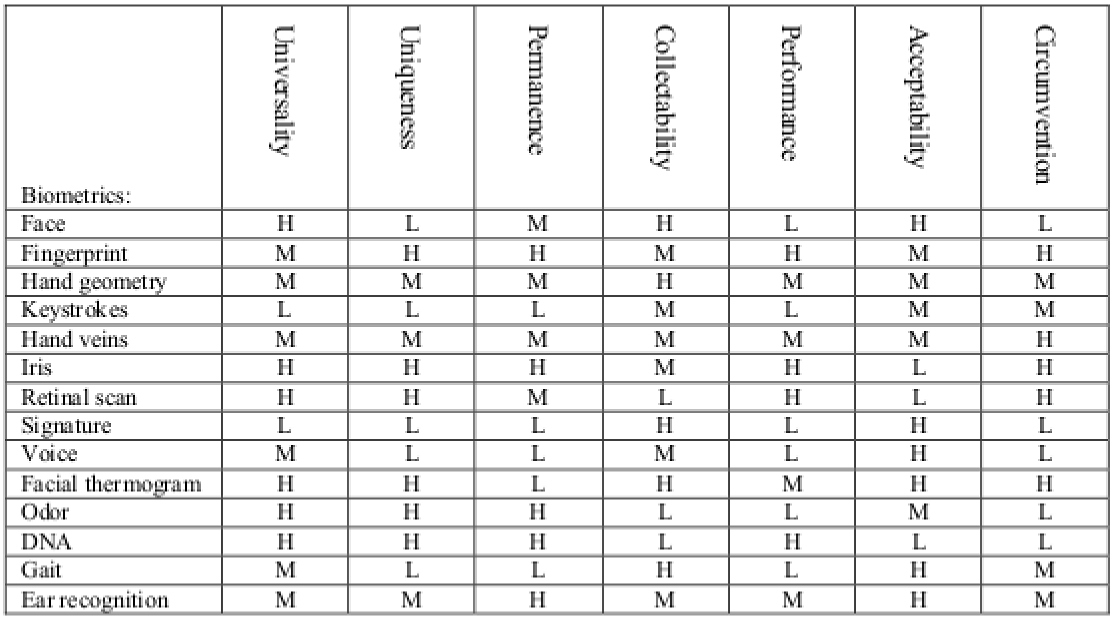
1.2 KEYSTROKE DYNAMICS

Keystroke Dynamics is a form of behavioural biometrics where a user is recognized by analysing his/her typing rhythms. Keystroke Dynamics can be used as a biometric identifier as each person has an almost unique pattern of typing. It can be used either for continuous authentication or for static authentication, for example for strengthening the username-password system. Keystroke Dynamics can be used to strengthen the user-name/password authentication system because it will not only consider the value of the password, but also how the password was typed.

When a person types, the latencies between successive keystrokes, keystroke durations, finger placement and applied pressure on the keys can be used to construct a unique signature (i.e., profile) for that individual. For well-known, regularly typed strings, such signatures can be quite consistent. Furthermore, recognition based on typing rhythm is not intrusive, making it quite applicable to computer access security as users will be typing at the keyboard anyway.

The input type used for keystroke biometric systems can be divided into long or short text. Short inputs normally consist of username [W. Chang, 2005; Obaidat and Sadoun, 1997], password [D.-T. Lin, 1997; Yu and Cho, 2003], or text phrase [Pavaday and Soyjaudah, 2007; S. Bleha and Hussien, 1990], while long inputs are usually referred to paragraphs of text enclosing 100 words or more [Samura and Nishimura, 2009; Zhao, 2006]. For instance, Gaines et al. [Crawford, 2010] considered whether typists could be identified by analysing keystroke times as they transcribed long passages of text. Techniques for analysing passages are different from those for analysing passwords, and the same evaluation data cannot always be used to assess both. This thesis deals with short text passwords.

The work [Anil K. Jain and Ross, 2008] proposes a table that presents evaluation of these parameters for biometric techniques in the following scale: H=High, M=Medium, L=Low. The evaluation is presented in the below table .



Biometrics:	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand geometry	M	M	M	H	M	M	M
Keystrokes	L	L	L	M	L	M	M
Hand veins	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retinal scan	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial thermogram	H	H	L	H	M	H	H
Odor	H	H	H	L	L	M	L
DNA	H	H	H	L	H	L	L
Gait	M	L	L	H	L	H	M
Ear recognition	M	M	H	M	M	H	M

Figure 1: Evaluation of biometric techniques

1.3 RELATIVE MERITS OF KEYSTROKE DYNAMICS

1.3.1 Advantages

The advantage of biometric methods over passwords is that reproducing password is considered a sufficient condition for being authorized; exact reproducing biometric input may be detected, as biometric data (from the same individual) are never exactly the same. Thus using keystroke dynamics authorization techniques is interesting because of lack of hardware requirements (contrary to most biometric techniques) and keeping the main advantage of biometric techniques: uniqueness.[Rybnik et al., 2008]

Biometrics also offers certain advantages such as negative recognition and non-repudiation that cannot be provided by tokens and passwords[S. Prabhakar and A. K. Jain, 2003]. Non-repudiation is a way to guarantee that an individual who accesses a certain facility cannot later deny using it (e.g., a person accesses a certain computer resource and later claims that an impostor must have used it under falsified credentials).[Anil K. Jain and Ross, 2008]

Continuous monitoring and authentication have often been sidelined yet they are relatively important. Keystroke dynamics biometrics offer a way to continuously validate the legitimate identity of a user. As long as user interaction with the system through input devices persists, keystroke pattern can be constantly monitored and re-evaluated.

Keystroke patterns are harder to be reproduced than written signatures. This is because most security systems only allow limited number of erroneous input attempts before locking down the account. Additionally, integration of keystroke dynamics biometrics leaves random password guessing attack obsolete, and stolen credentials become entirely insignificant, since successful possession of secret key is only a mere condition of the entire authentication chain. Even if it does get compromised, a new typing biometric template can be regenerated easily by choosing a new password.[Pin Shen Teh and Yue, 2013]

1.3.2 Disadvantages

The European standard for access-control systems (EN-50133-1) specifies a false-alarm rate of less than 1%, with a miss rate of no more than 0.001% [CENELEC, 2002]. At present, no anomaly detector has achieved these error rates in repeated evaluation, and so keystroke dynamics could not be deployed as a sole access-control technology.

The advantages of keystroke biometrics come at the tradeoff of significant variability and an uncontrolled authentication environment. Unlike other physiological biometrics such as fingerprints, retinas, and facial features, all of which remain fairly consistent over long periods of time, typing patterns can be rather erratic. Thus, having low permanence.[Rybnik et al., 2008]

1.4 VERIFICATION VS IDENTIFICATION

Keystroke dynamics authentication can be categorized as verification and identification. Verification refers to the process of proofing a validity of claimed identity. In other words, "Is this person really who he or she declares to be." On the contrary, identification denotes "Is this person in our database, if yes, to whom this presented identity belongs to." Identification is generally more time consuming, slower in responsiveness, and require higher processing capacity. Verification is the most common scenario in our society's security access control environment whereas identification mode has its own unique usage such as forensic investigation and intrusion detection.

Majority of keystroke dynamics research works have been investigated in the form of verification mode (89%) compared to identification (5%). This thesis deals with identification of user.[Pin Shen Teh and Yue, 2013]

This thesis deals with the testing of two models for keystroke dynamics i.e a neural network and a SVM model. These models are tested against two datasets. The first dataset is a dataset described in [Killourhy and Maxion, 2000] and the second dataset is a new dataset created for testing purposes. The performance of these two datasets with the models are compared in this work.

2 LITERATURE SURVEY

Keystroke-dynamics research was inspired by much older work that distinguished telegraph operators by their keying rhythms. This capability—which was allegedly quite useful during World War II for identifying radio operators and tracking troop movements [Gladwell, 2005] was first formally investigated by Bryan and Harter[Bryan and Harter, 1897, 1899] as part of a study on skill acquisition in telegraph operators.

Keyboard typing rhythms were first considered as a means of distinguishing typists in the mid 1970s. Spillane[Spillane, 1975] suggested in an IBM technical bulletin that typing rhythms might be used for identifying the user at a computer keyboard. That bulletin described keystroke dynamics in concept; it laid out no specific classifier and reported no empirical evaluation results. While telegraph key served as an input device in those days, likewise, computer keyboard, mobile keypad, and touch screen are common input devices in the 21st century.

Furthermore, it has been noted that keystroke pattern has the same neurophysiological factors that make hand written signature unique, where humans have relied on to verify identity of an individual for many centuries. In fact, keystroke pattern is capable of providing even more unique feature for authentication, which includes key press duration and latencies, typing rate, and typing pressure.

Since [Forsen et al., 1977] first investigated in 1977 whether users could be distinguished by the way they type their names, many different techniques and uses for keystroke dynamics have been proposed. [Peacock et al., 2004] conducted an extensive survey of the keystroke dynamics literature. Not all of that research is relevant to the use considered in this work, namely, analysing password-typing times. For instance, [Gaines et al., 1980] considered whether typists could be identified by analysing keystroke times as they transcribed long passages of text. Techniques for analysing passages are different from those for analysing passwords, and the same evaluation data cannot always be used to assess both.

Further, even among studies of password-typing times, not all of the extant techniques can be evaluated using the same procedure. One class of techniques is anomaly detection. The typing samples of a single, genuine user are used to build (or train) a model of the user's typing behaviour. When a new typing sample is presented, the detector tests the sample's similarity to the model, and outputs an anomaly score. In contrast, another class of techniques is multi-class classification. The typing samples of multiple users are used to find decision boundaries that can be used to distinguish each user from the others. Since anomaly detectors train on a single user's data, while multi-class classifiers train on multiple users' data, these two techniques require different evaluation procedures.

[Haider et al., 2000] in 2000 presented a suite of techniques for password authentication using neural networks, fuzzy logic, statistical methods, and several hybrid combinations of these approaches. The approaches presented in their paper used typing biometrics of a user, in addition to conventional login information, to identify a user. The system limited the password length to 7 characters, which is considered a standard password length. During the learning process users were required to enter the password 15 times. The delays between each of the characters of the password string were recorded. At the end of the learning process the system has 6 vectors of length 15 representing inter-character delays. These values are then passed to the neural, statistical, and fuzzy unit.

Since many anomaly-detection algorithms have been proposed for finding imposters with the help of keystroke dynamics, [Killourhy and Maxion, 2000] set to find out the top performers among them (e.g., to identify promising research directions).

Their objective was to collect a keystroke-dynamics data set, to develop a repeatable evaluation procedure, and to measure the performance of a range of detectors so that the results can be compared soundly. They collected data from 51 subjects typing 400 passwords each, and implemented and evaluated 14 detectors from the keystroke- dynamics and pattern-recognition literature. The three top-performing detectors achieve equal-error rates between 9.6% and 10.2%. The results—along with the shared data and evaluation methodology—constitute a benchmark for comparing detectors and measuring progress.

It seems that the focus of most keystroke-dynamics research has been on finding the right classifier for each problem. Nearly every paper proposes a new classifier, and many propose several new classifiers. Yet, we also see the same families of classifier appearing in multiple studies. Neural networks have been used in over a dozen studies. Support vector machines (SVMs) and k-NNs have been independently proposed multiple times. Many different papers propose statistical methods that differ only slightly (e.g., using Scaled Manhattan vs. Scaled Euclidean as a distance metric). Each of these classifiers is evaluated and empirical results are reported (e.g., miss and false-alarm rates). When we look at how a family of classifiers has fared in multiple evaluations, we see wildly different results. Since the classifiers are essentially the same, perhaps differences in the evaluations explain the different results.

3 DATASET

Various public datasets used in Keystroke Dynamics exist in literature. To test our models, we decided to use the CMU dataset created by [Killourhy and Maxion, 2000], and in addition to that, created a dataset of our own. Both of them are explained in subsequent sections.

3.1 IIITBH DATASET

In order to test our models, we decided to create our own database of typing patterns so that we could collect data in a consistent and controlled environment, incorporating our own desired features and auditing the entire collection process. This section explains our password collecting machine, the choice of password and the features extracted.

We decided to choose an arbitrary password which would be typed by all our test subjects. The password chosen was:

.xat17padn

It consists of 10 characters with a mixture of letters, numbers and a special character and hence is a fairly strong password.

3.1.1 Data collection machine

We collected the data on a laptop running Linux OS using the laptop's standard keyboard. The keylogger application was written in Python and which when run, prompts the user to type the password. The password to be typed was kept beside the user. The user must type the password sequence correctly and press Enter in order to proceed to the next iteration. Whenever a user presses or releases a key, the program event along with its timestamp is recorded.

If the password sequence typed by a user is wrong, they are prompted to retype the password. We also chose not to incorporate Backspaces. Thus if a user makes an error and seeks to correct it using a Backspace, it would still be regarded as an error and the user would be prompted to type again.

3.1.2 Collection Procedure

Our Dataset is divided into two parts: The first file (IIITBh-Big.csv) is a database of 12 users typing the password 150 times in total over 6 sessions. Each session consists of 25 attempts and the sessions were divided over a period of 1 month in order to capture some natural variations in typing patterns and such that the subjects don't get too used to the keyboard in a single session. The second file (IIITBh-Small.csv) is a database of 5 users with 250 samples for each user collected in a single sitting of continuous 10 sessions.

3.1.3 Feature vector

From the raw data collected by the key-logger which consists of only the key and timestamps, we extracted timing features which can be used directly by our models.

The features extracted for each key were:

- The hold time for a key
- The KeyDown-KeyUp time between the key and the next key
- The KeyDown-KeyDown time between the key and the next key(which is effectively a linear combination of the first two features).

We also chose to consider the Enter key as a part of the password. The timing features are recorded in seconds as floating point numbers. Thus after extraction, a single sample of a user looks of features looks like:

The data are arranged as a table with 34 columns. Each row of data corresponds to the timing information for a single repetition of the password by a single subject. The first column, subject, is the name of the subject. The second column, sessionIndex, is the session in which the password was typed. The third column, rep, is the repetition of the password within the session (ranging from 1 to 25).

The remaining 31 columns present the timing information for the password. The name of the column encodes the type of timing information. Column names of the form H.key designate a hold time for the named key (i.e., the time from when key was pressed to when it was released). Column names of the form DD.key1.key2 designate a keydown-keydown time for the named digraph(a combination of two letters representing one sound, as in *ti* and *Ro*). Column names of the form UD.key1.key2 designate a keyup-keydown time for the named digraph (i.e., the time from when key1 was released to when key2 was pressed). Note that UD times can be negative, and that H times and UD times add up to DD times.

Consider the following one-line example of what you will see in the data:

subject	sessionIndex	rep	H.period	DD.period.t	UD.period.t	...
s002	1	1	0.2491	0.379	0.288	...

The example presents typing data for subject 2, session 1, repetition 1. The period key was held down for 0.2491 seconds (249.1 milliseconds); the time between pressing the period key and the t key (keydown-keydown time) was 0.379 seconds; the time between releasing the period and pressing the t key (keyup-keydown time) was 0.288 seconds; and so on.

Owing to the limited number of datasets freely available to the public based on keystroke dynamics, we decided to make our dataset public, such that it can be freely used in future works.

3.2 CMU DATASET

The CMU is a public dataset which was proposed by [Killourhy and Maxion, 2000]. It is one of the most important and widely used dataset in keystroke literature.

The CMU dataset consists of a database of 51 users providing 400 samples divided over 8 sessions with each session consisting of 50 attempts. There was atleast a one day difference between the sessions. A single password was typed by all users: ".tie5Roanl".

3.2.1 Data collection machine

A laptop with an external keyboard to collect data was used, and a Windows application was developed that prompts a subject to type the password. The application displays the password in a screen with a text-entry field. In order to advance to the next screen, the subject must type the 10 characters of the password correctly, in sequence, and then press Enter. The subject must type the password correctly 50 times to complete a data-collection session. Whenever the subject presses or releases a key, the application records the event (i.e., keydown or keyup), the name of the key involved, and what time the event occurred. An external reference clock was used to generate highly accurate timestamps.

3.2.2 Collection procedure

51 subjects from within the university were recruited. Subjects completed 8 data-collection sessions (of 50 passwords each), for a total of 400 password-typing samples. They waited at least one day between sessions, to capture some of the day-to-day variation of each subject's typing. The set of subjects consisted of 30 males and 21 females. There were 8 left-handed and 43 right handed subjects. The median age group was 31–40, the youngest was 18–20 and the oldest was 61–70. The subjects' sessions took between 1.25 and 11 minutes, with the median session taking about 3 minutes.

3.2.3 Feature vector

The extracted features contained in the database are: hold time, interval between two pressures, and interval between the release of a key and pressure of the next one. So for a 10 character password(effectively 11, as the Enter key has also been considered), 31 features are extracted and stored. The dataset can be found in raw text, CSV or Excel files at: [K. Killourhy, 2009].

3.3 SUMMARY

This chapter explained the datasets used in our study to classification of users based on keystroke dynamics. We used two different datasets, one the dataset collected in CMU which contains 51 users typing a password 400 times. The second dataset is the IIITBh dataset, developed in IIIT Bhubaneswar, which contains the typing data of 12 users, typing a single password 150 times and typing data of 3 users typing the same password

250 times. In the subsequent chapters, we test both of these datasets against our classifiers and analyse the performance.

4 IMPLEMENTATION

4.1 SVM MODELS

4.1.1 Introduction

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.[Pedregosa et al., 2011]

The kernel function can be one of Linear, RBF, polynomial, sigmoid kernel.

4.1.2 Details

The classifier is based on the SVM.svc() library which in turn uses libsvm[C.-C. Chang and C.-J. Lin, 2011] for creating the SVM classifier. Two kernels Linear and RBF(Radial Basis Function) kernel are used for the SVM in the thesis. The results of the other two kernels sigmoid and polynomial haven't been shown as they didn't perform as well as linear and RBF.

Size of training set was set at 0.64 of available data. Size of testing set was set at 0.36 of available data.

Exhaustive grid search over the below specified parameter values for the hyperparameters C(Regularization parameter) and gamma for estimators with each of the mentioned kernel was done. We used stratified K folds with K=12 for obtaining the score during the grid search. After the grid search, the best configuration was retained and tested against the test set. The parameters for grid search are:

$'C' : [1, 101, 201, 301, 401, 501, 601, 701, 801, 901],' gamma' : [0.01, 0.1, 1.0, 0.05, 0.5, 5.0]$

The tuned parameters for the best configuration came out to be:
For linear kernel, parameters are {'kernel': 'linear', 'C': 801, 'gamma': 0.01}
For rbf kernel, parameters are {'kernel': 'rbf', 'C': 401, 'gamma': 0.05}

4.2 NEURAL NETWORK MODELS

To train against our data using neural networks, we used a feed-forward multi layer percpetron (MLP) trained using the backpropagation algorithm.

- The number of nodes in the input layer of the MLP is equal to the number columns in the feature vector(31)

- The number of output nodes is set to the number of classes. For example, in the IIITBh dataset we have 12 users, thus the output node size is set to 12.

To find a more efficient neural structure for our available data, various number of hidden nodes were simulated and examined. After the tuning process, the hidden node size was set to be 25.

For activation of the nodes, the hyperbolic tangent function(tanh) was chosen as the activation function for the hidden layers. For the output layer, we used a softmax function. At the beginning of the learning process, the weight matrices between in-put and hidden layer(M1) and between hidden and output layer(M2) are randomly initialized.

For the learning process i.e. in order to minimize our loss function, we used two optimization techniques – (1) Stochastic Gradient Descent and (2) ADAM

4.2.1 Stochastic Gradient Descent

Stochastic Gradient Descent(SGD) is a stochastic approximation of the gradient descent optimization method ofr minimizing an objective function. In contrast to batch gradient descent, SGD approximates the true gradient by considering a single training example at a time.

In stochastic (or "on-line") gradient descent, the true gradient of is approximated by a gradient at a single example:

$$w := w - \eta \nabla Q_i(w).$$

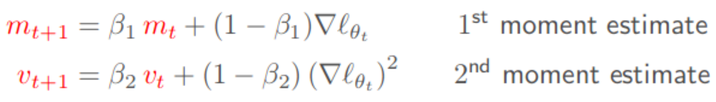
The parameters set for training using the SGD optimizer are as follows. Some of these parameters were tuned using Grid search to find the optimal values.

- The learning rate was set to be 0.01.
- The L2 regularization parameter was set to be 0.01. This was done to prevent overfitting scenario.
- SGD can also update weigts by using an optional momentum parameter which helps in better convergence. The value of this parameter was set to be 0.9

4.2.2 ADAM

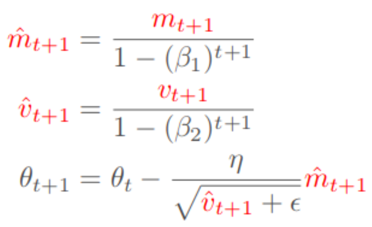
Adaptive Moment Estimation (Adam)[Kingma and Ba, 2014] is an optimizing technique that computes adaptive learning rates for each parameter. Studies have showed ADAM to be well suited for problems that are large in terms of data and/or parameters.

In addition to storing an exponentially decaying average of past squared gradients vt, Adam also keeps an exponentially decaying average of past gradients mt, similar to momentum.



$$m_{t+1} = \beta_1\, m_t + (1-\beta_1)\nabla \ell_{\theta_t} \qquad 1^{\text{st}} \text{ moment estimate}$$

$$v_{t+1} = \beta_2\, v_t + (1-\beta_2)\left(\nabla \ell_{\theta_t}\right)^2 \qquad 2^{\text{nd}} \text{ moment estimate}$$



$$\hat{m}_{t+1} = \frac{m_{t+1}}{1-(\beta_1)^{t+1}}$$

$$\hat{v}_{t+1} = \frac{v_{t+1}}{1-(\beta_2)^{t+1}}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_{t+1}+\epsilon}}\hat{m}_{t+1}$$

The two most important hyper-parameters in case of ADAM are its two decay rates, which we set as 0.9 and 0.99 respectively.

The training and testing ratio set here is the same as the ratios set in the SVM models.

4.3 SUMMARY

This chapter explained in detailed about the various classifiers used for training and testing our data. We used two classifiers using Support Vector Machines(SVM); one using the linear kernel(SVC(linear)) and another using rbf kernel(SVC(rbf)). Similarly, two classifiers were used using neural networks; one using the SGD optimizer(NN(SGD)) and another using ADAM solver(NN(adam)). The next chapters goes into detail about the results of our models tested against both the datasets.

5 PERFORMANCE MEASURES

5.1 DEFINITIONS

To measure the performance of our models against the two datasets, the following measures were used

- ROC curve: A Receiver Operator Characteristic(ROC) curve is a commonly used way to visualize the performance of a binary classifier. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. Here we use the ROC curve for a multi-class problem. The ROC curve is a common visualization of a model's accuracy, and on the basis of the ROC curve, various measures of error can be derived. One such frequently used measure is the Area under the curve(AUC), which we use here in our work.

- Precision: Precision is defined as the fraction of relevant instances among the retrieved instances. It is also referred to as positive predictive value(PPV). In classification context, where the performance of a classifier derived from four basic values (True Positives, True negatives, False Positives and False negatives), precision can be defined as:

$$\text{Precision} = \frac{tp}{tp + fp}$$

- Recall: Recall is another metric frequently used along with precision for classification tasks. Recall is defined as the fraction of relevant instances that have been retrieved over total relevant instances. In classification context,

$$\text{Recall} = \frac{tp}{tp + fn}$$

- F-measure: As both precision and recall are often used in tandem, a measure called the F-measure is often used to combine these two metrics. It is defined as the harmonic mean of precision and recall.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

5.2 RESULTS

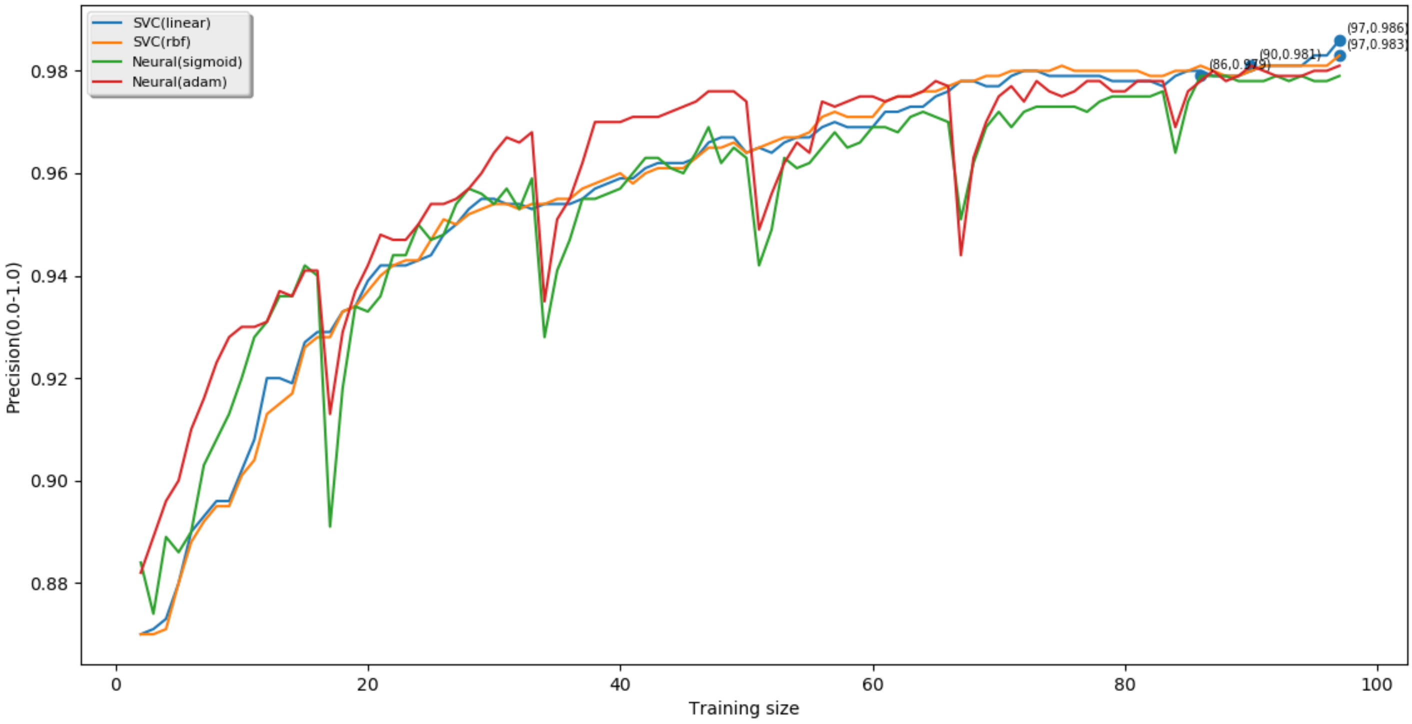


Figure 2: Precision vs Training size for IIITBh data

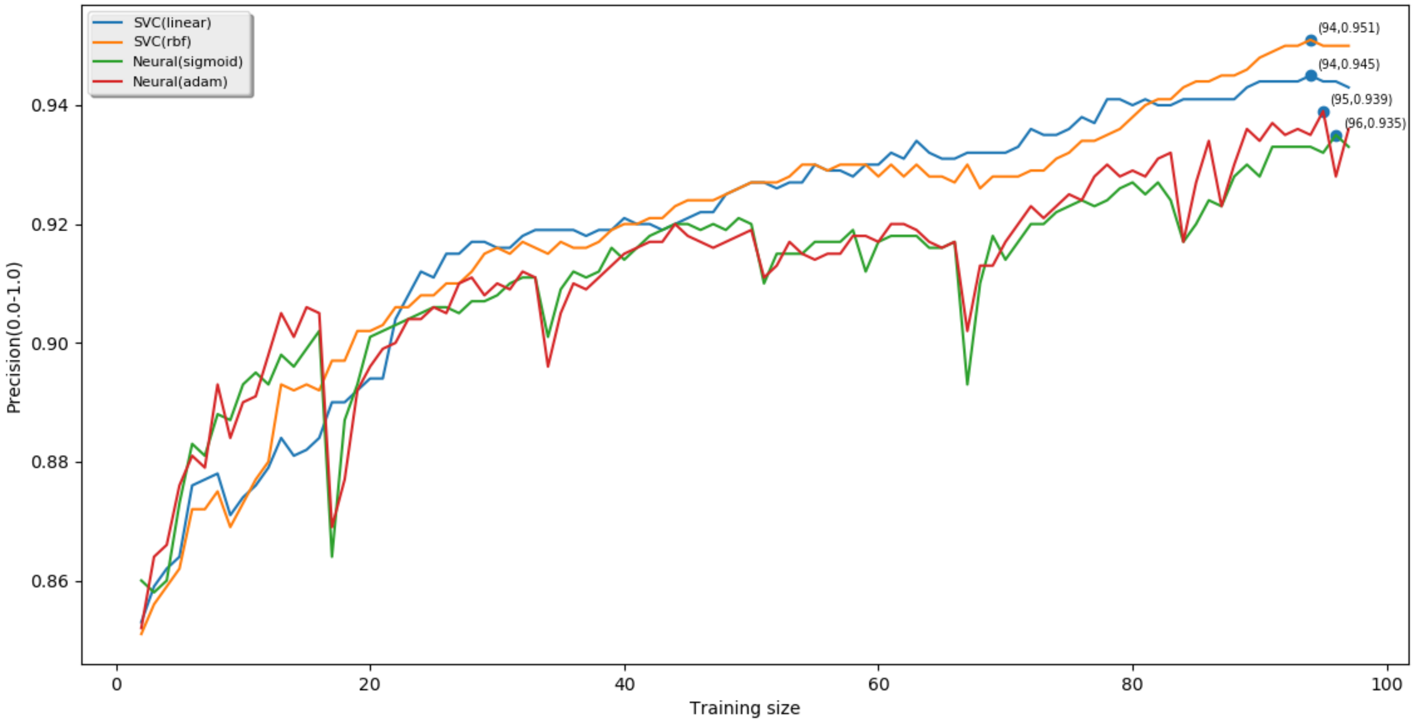


Figure 3: Precision vs Training size for CMU data

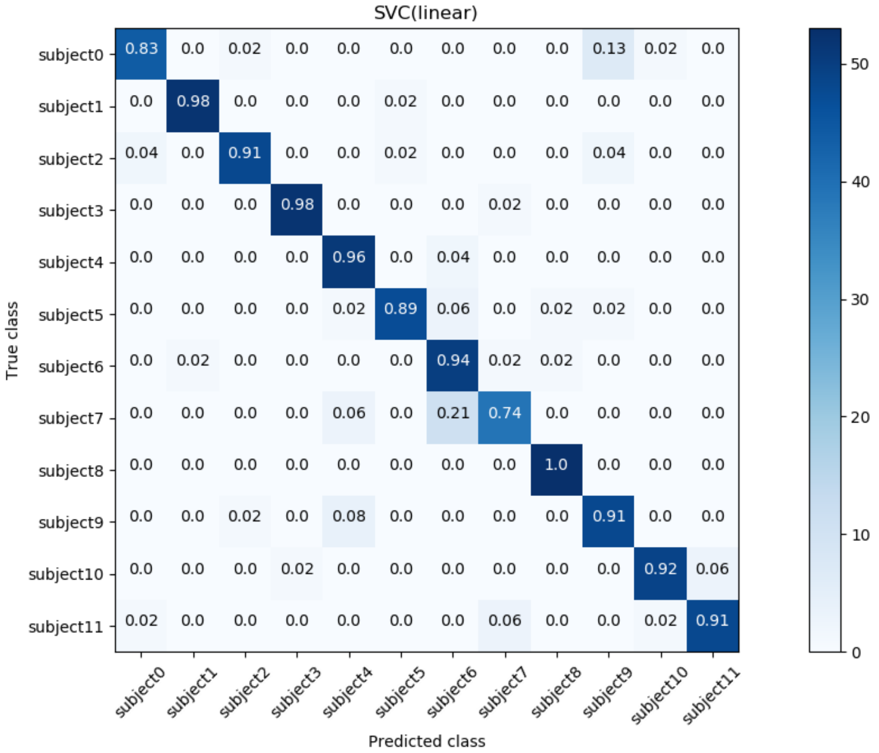


Figure 4: Confusion matrix for IIITBh data - SVC(Linear)

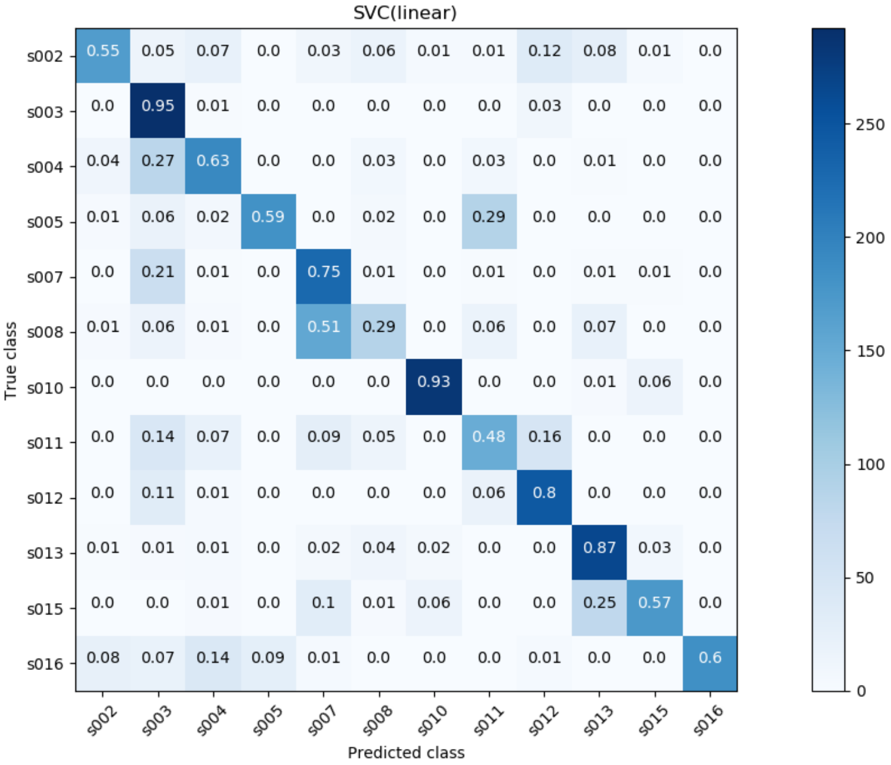


Figure 5: Confusion matrix for CMU data - SVC(Linear)

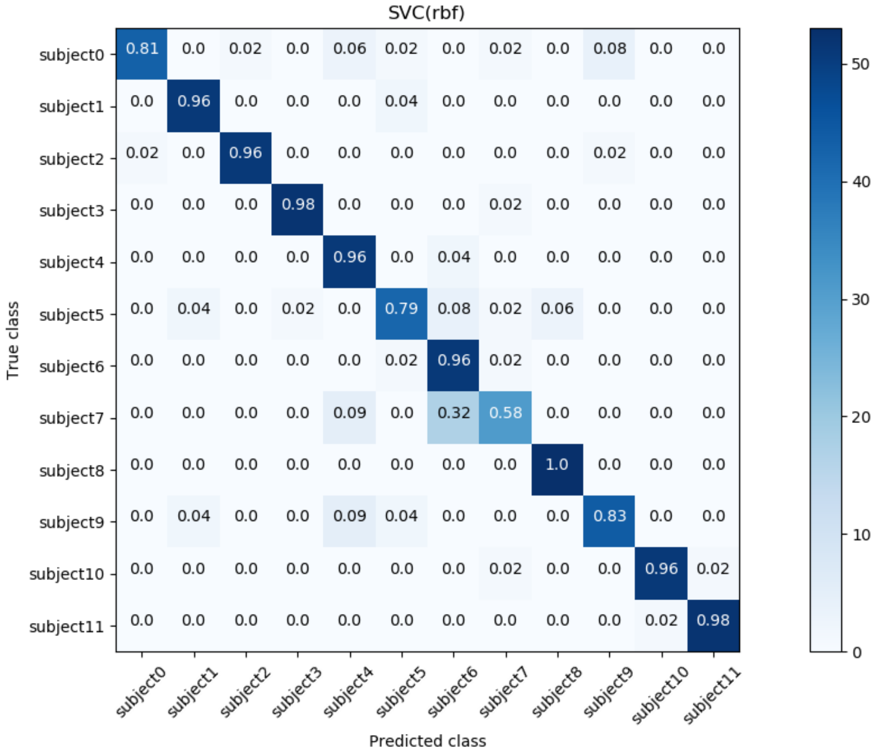


Figure 6: Confusion matrix for IIITBh data - SVC(RBF)

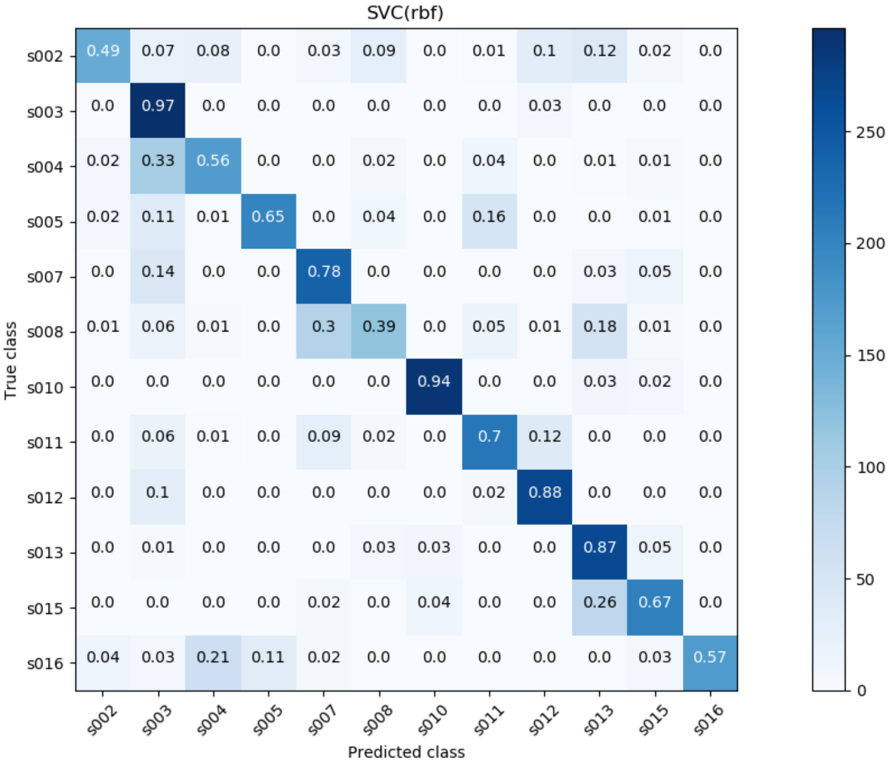
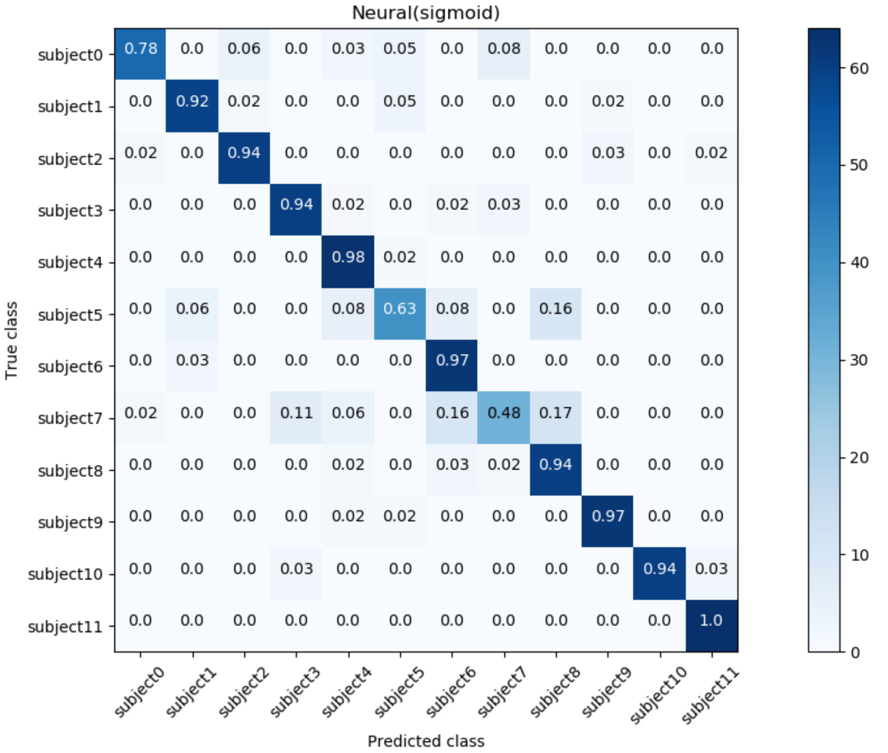
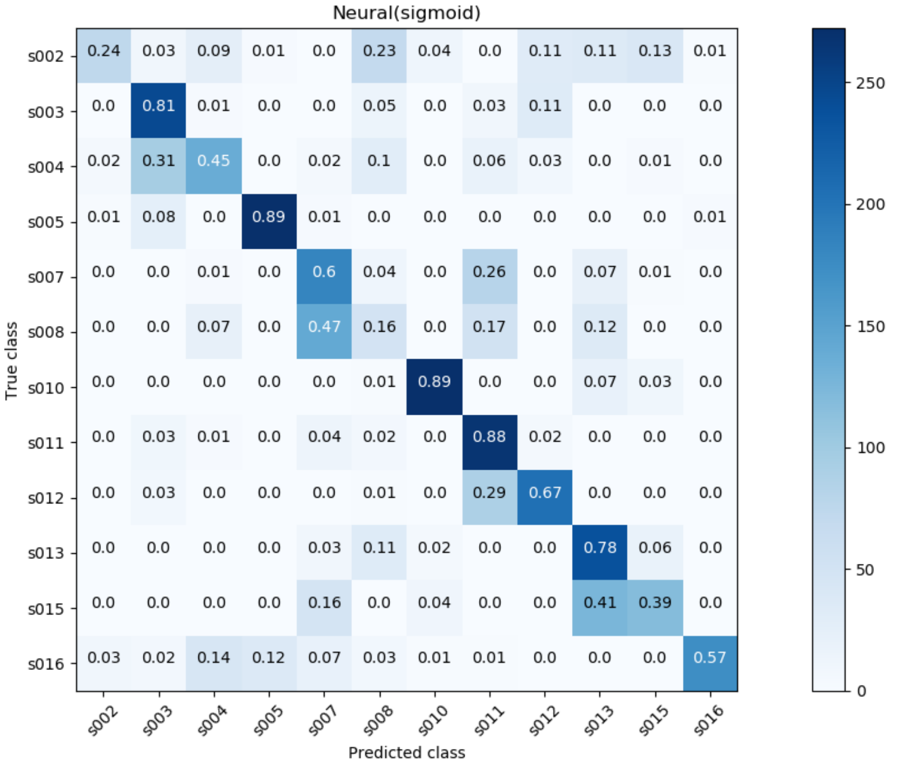


Figure 7: Confusion matrix for CMU data - SVC(RBF)



	subject0	subject1	subject2	subject3	subject4	subject5	subject6	subject7	subject8	subject9	subject10	subject11
subject0	0.78	0.0	0.06	0.0	0.03	0.05	0.0	0.08	0.0	0.0	0.0	0.0
subject1	0.0	0.92	0.02	0.0	0.0	0.05	0.0	0.0	0.0	0.02	0.0	0.0
subject2	0.02	0.0	0.94	0.0	0.0	0.0	0.0	0.0	0.0	0.03	0.0	0.02
subject3	0.0	0.0	0.0	0.94	0.02	0.0	0.02	0.03	0.0	0.0	0.0	0.0
subject4	0.0	0.0	0.0	0.0	0.98	0.02	0.0	0.0	0.0	0.0	0.0	0.0
subject5	0.0	0.06	0.0	0.0	0.08	0.63	0.08	0.0	0.16	0.0	0.0	0.0
subject6	0.0	0.03	0.0	0.0	0.0	0.0	0.97	0.0	0.0	0.0	0.0	0.0
subject7	0.02	0.0	0.0	0.11	0.06	0.0	0.16	0.48	0.17	0.0	0.0	0.0
subject8	0.0	0.0	0.0	0.0	0.02	0.0	0.03	0.02	0.94	0.0	0.0	0.0
subject9	0.0	0.0	0.0	0.0	0.02	0.02	0.0	0.0	0.0	0.97	0.0	0.0
subject10	0.0	0.0	0.0	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.94	0.03
subject11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Figure 8: Confusion matrix for IIITBh data - Neural(Sigmoid)



	s002	s003	s004	s005	s007	s008	s010	s011	s012	s013	s015	s016
s002	0.24	0.03	0.09	0.01	0.0	0.23	0.04	0.0	0.11	0.11	0.13	0.01
s003	0.0	0.81	0.01	0.0	0.0	0.05	0.0	0.03	0.11	0.0	0.0	0.0
s004	0.02	0.31	0.45	0.0	0.02	0.1	0.0	0.06	0.03	0.0	0.01	0.0
s005	0.01	0.08	0.0	0.89	0.01	0.0	0.0	0.0	0.0	0.0	0.0	0.01
s007	0.0	0.0	0.01	0.0	0.6	0.04	0.0	0.26	0.0	0.07	0.01	0.0
s008	0.0	0.0	0.07	0.0	0.47	0.16	0.0	0.17	0.0	0.12	0.0	0.0
s010	0.0	0.0	0.0	0.0	0.0	0.01	0.89	0.0	0.0	0.07	0.03	0.0
s011	0.0	0.03	0.01	0.0	0.04	0.02	0.0	0.88	0.02	0.0	0.0	0.0
s012	0.0	0.03	0.0	0.0	0.0	0.01	0.0	0.29	0.67	0.0	0.0	0.0
s013	0.0	0.0	0.0	0.0	0.03	0.11	0.02	0.0	0.0	0.78	0.06	0.0
s015	0.0	0.0	0.0	0.0	0.16	0.0	0.04	0.0	0.0	0.41	0.39	0.0
s016	0.03	0.02	0.14	0.12	0.07	0.03	0.01	0.01	0.0	0.0	0.0	0.57

Figure 9: Confusion matrix for CMU data - Neural(Sigmoid)

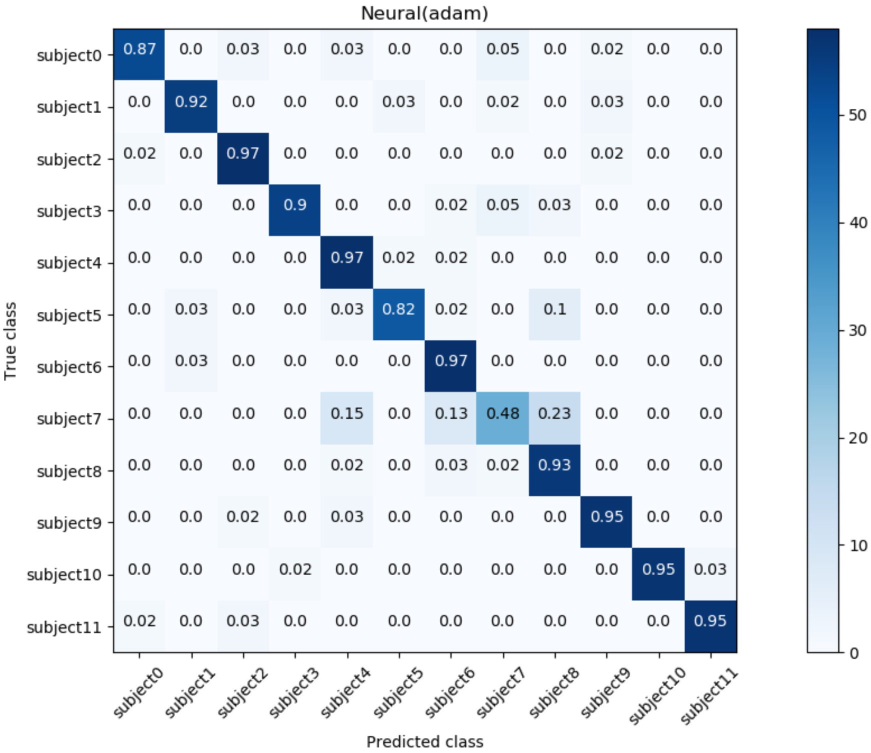


Figure 10: Confusion matrix for IIITBh data - Neural(Adam)

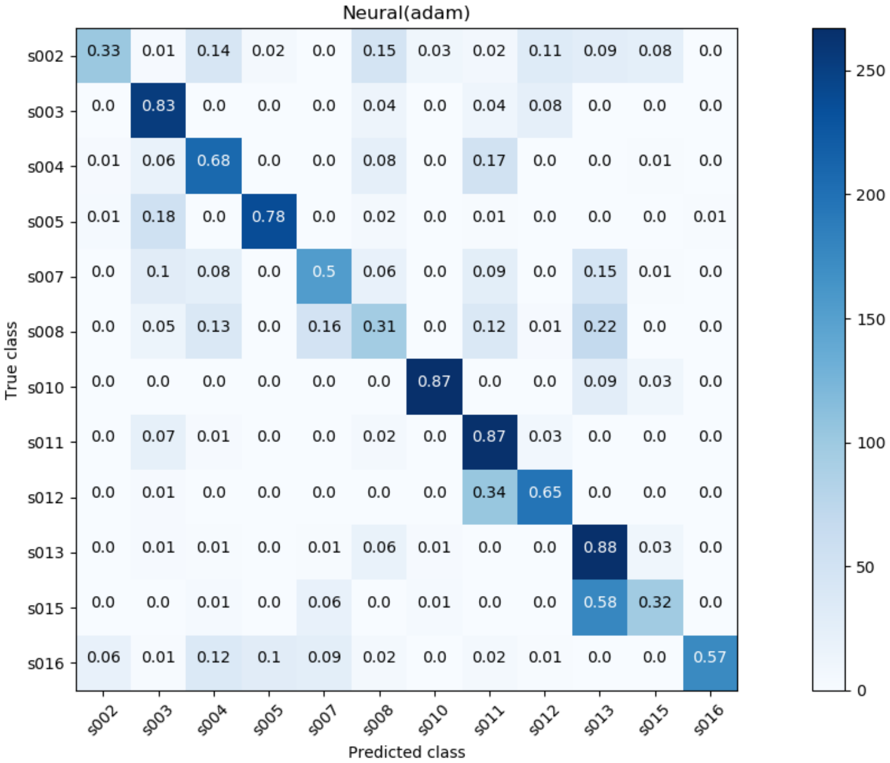
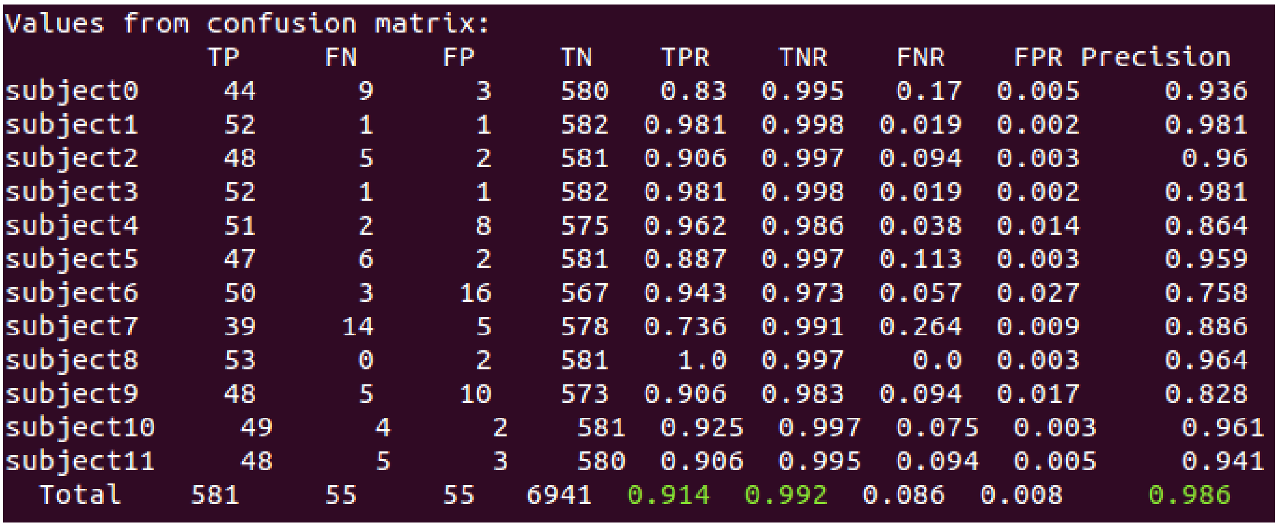


Figure 11: Confusion matrix for CMU data - Neural(Adam)

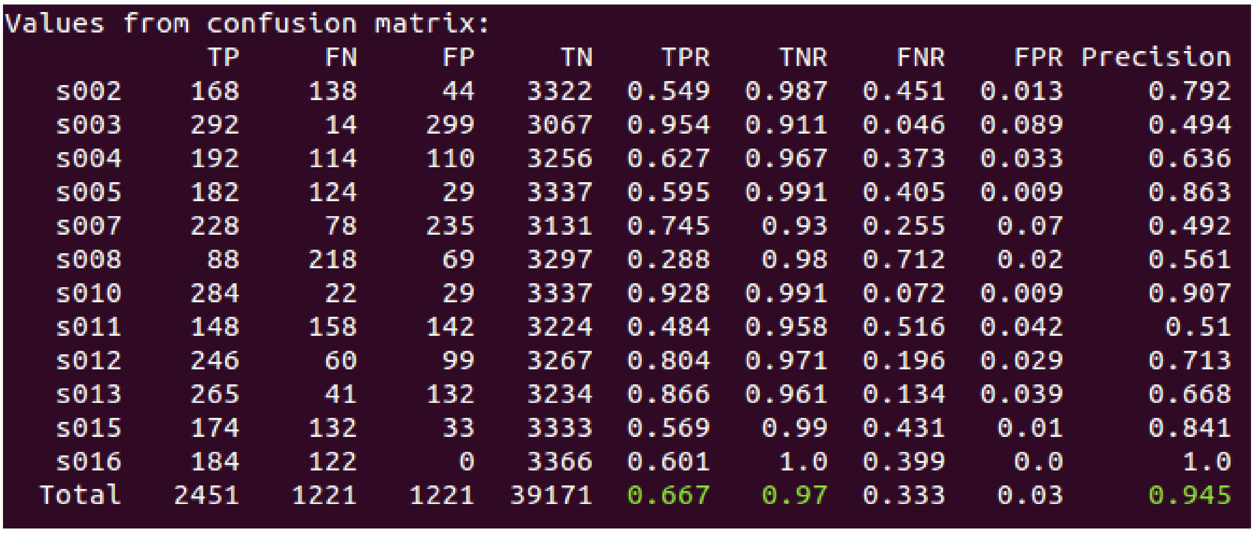
Various metrics from the confusion matrices above were calculated. These metrics include TP, FN, FP, TN, TPR, TNR, FNR, FPR, Precision.

The metrics for only the best performing classifier based on the metric Precision has been shown here which is SVC(Linear).



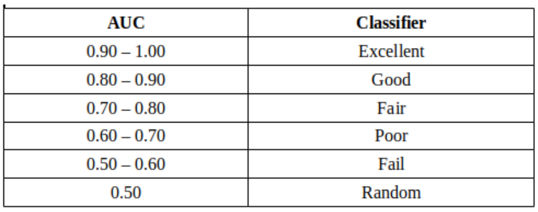
```
Values from confusion matrix:
             TP      FN      FP      TN     TPR     TNR     FNR     FPR Precision
subject0     44       9       3     580    0.83   0.995    0.17   0.005     0.936
subject1     52       1       1     582   0.981   0.998   0.019   0.002     0.981
subject2     48       5       2     581   0.906   0.997   0.094   0.003      0.96
subject3     52       1       1     582   0.981   0.998   0.019   0.002     0.981
subject4     51       2       8     575   0.962   0.986   0.038   0.014     0.864
subject5     47       6       2     581   0.887   0.997   0.113   0.003     0.959
subject6     50       3      16     567   0.943   0.973   0.057   0.027     0.758
subject7     39      14       5     578   0.736   0.991   0.264   0.009     0.886
subject8     53       0       2     581     1.0   0.997     0.0   0.003     0.964
subject9     48       5      10     573   0.906   0.983   0.094   0.017     0.828
subject10     49       4       2     581   0.925   0.997   0.075   0.003     0.961
subject11     48       5       3     580   0.906   0.995   0.094   0.005     0.941
  Total     581      55      55    6941   0.914   0.992   0.086   0.008     0.986
```

Figure 12: Metrics for SVC(Linear) on IIITBh dataset



```
Values from confusion matrix:
             TP      FN      FP      TN     TPR     TNR     FNR     FPR Precision
   s002     168     138      44    3322   0.549   0.987   0.451   0.013     0.792
   s003     292      14     299    3067   0.954   0.911   0.046   0.089     0.494
   s004     192     114     110    3256   0.627   0.967   0.373   0.033     0.636
   s005     182     124      29    3337   0.595   0.991   0.405   0.009     0.863
   s007     228      78     235    3131   0.745    0.93   0.255    0.07     0.492
   s008      88     218      69    3297   0.288    0.98   0.712    0.02     0.561
   s010     284      22      29    3337   0.928   0.991   0.072   0.009     0.907
   s011     148     158     142    3224   0.484   0.958   0.516   0.042      0.51
   s012     246      60      99    3267   0.804   0.971   0.196   0.029     0.713
   s013     265      41     132    3234   0.866   0.961   0.134   0.039     0.668
   s015     174     132      33    3333   0.569    0.99   0.431    0.01     0.841
   s016     184     122       0    3366   0.601     1.0   0.399     0.0       1.0
  Total    2451    1221    1221   39171   0.667    0.97   0.333    0.03     0.945
```

Figure 13: Metrics for SVC(Linear) on CMU dataset



AUC	Classifier
0.90 – 1.00	Excellent
0.80 – 0.90	Good
0.70 – 0.80	Fair
0.60 – 0.70	Poor
0.50 – 0.60	Fail
0.50	Random

Figure 14: Interpretation of AUC values

From figures 14, 15 and 16 we can conclude that the classifier Neural(Adam) performed better than Neural(SGD) on the basis of their AUC values as AUC values of Neural(Adam) was consistently above 0.9.

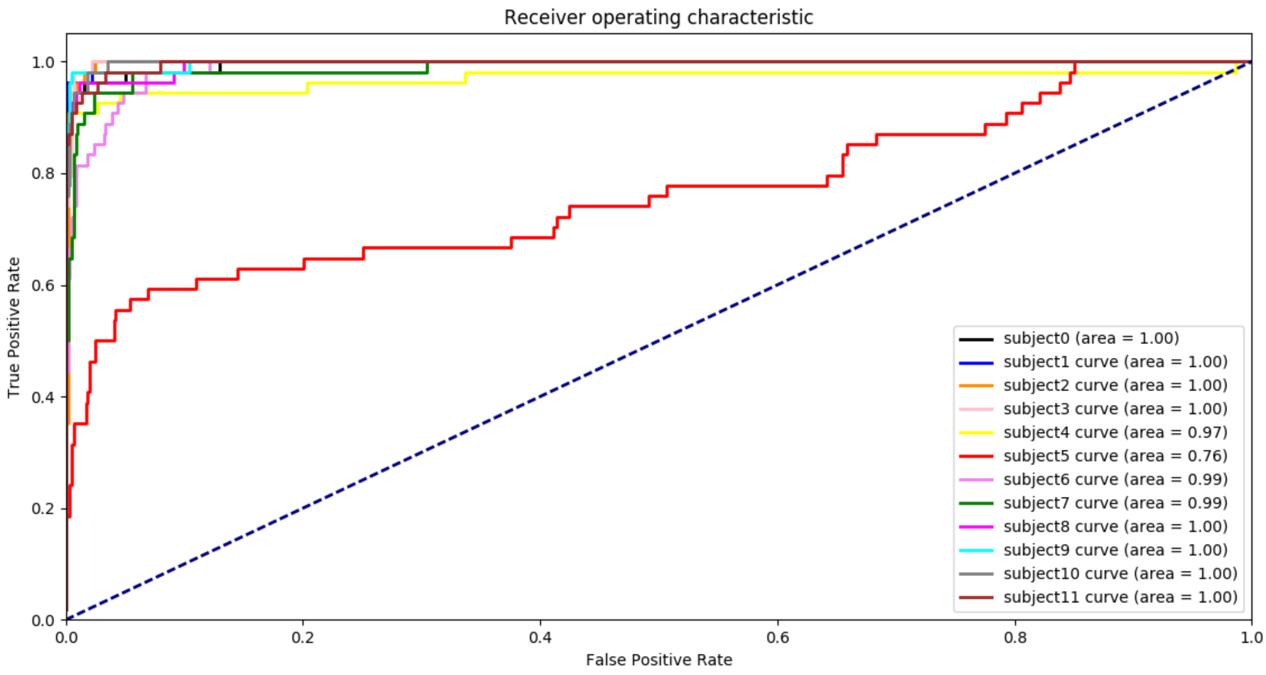


Figure 15: ROC curve for IIITBh data - Neural(Adam)

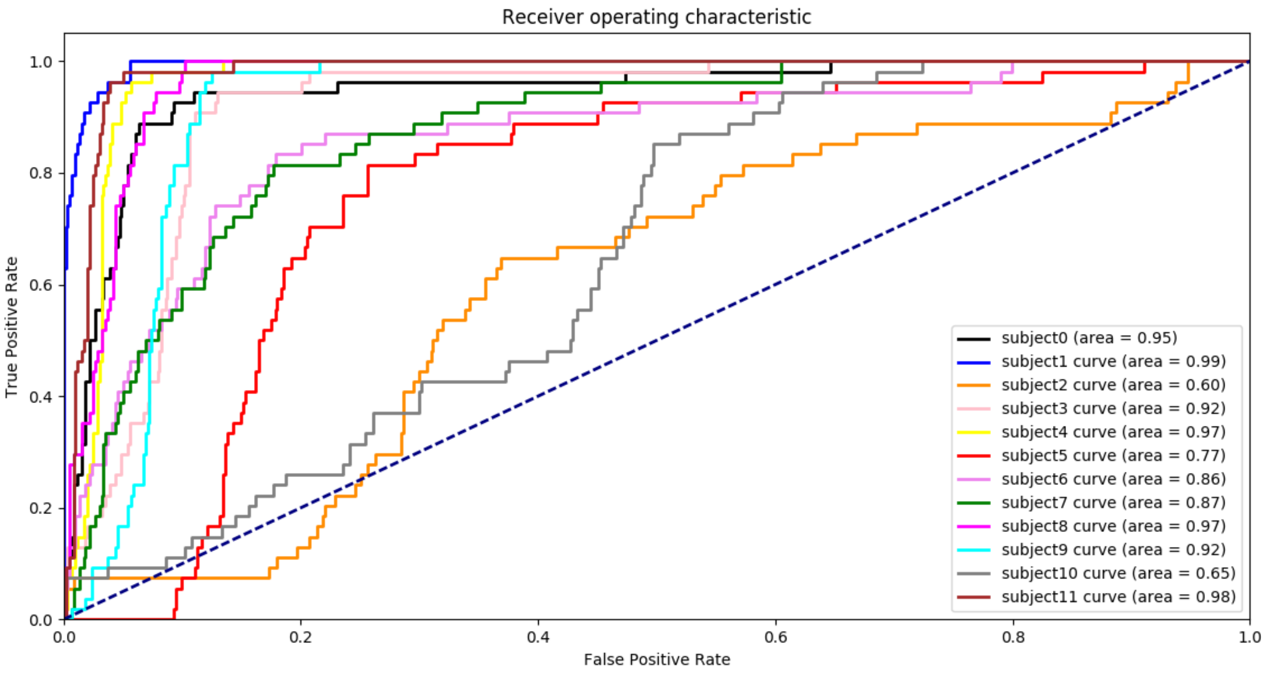


Figure 16: ROC curve for CMU data - Neural(Adam)

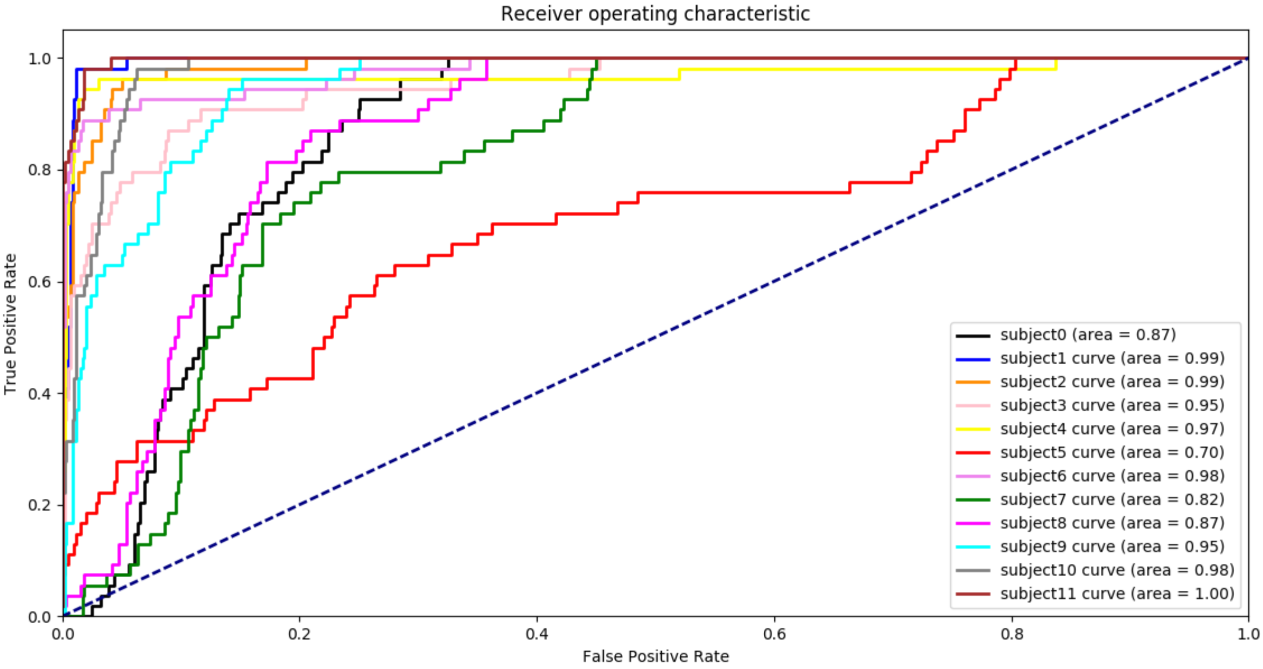


Figure 17: ROC curve for IIITBh data - Neural(SGD)

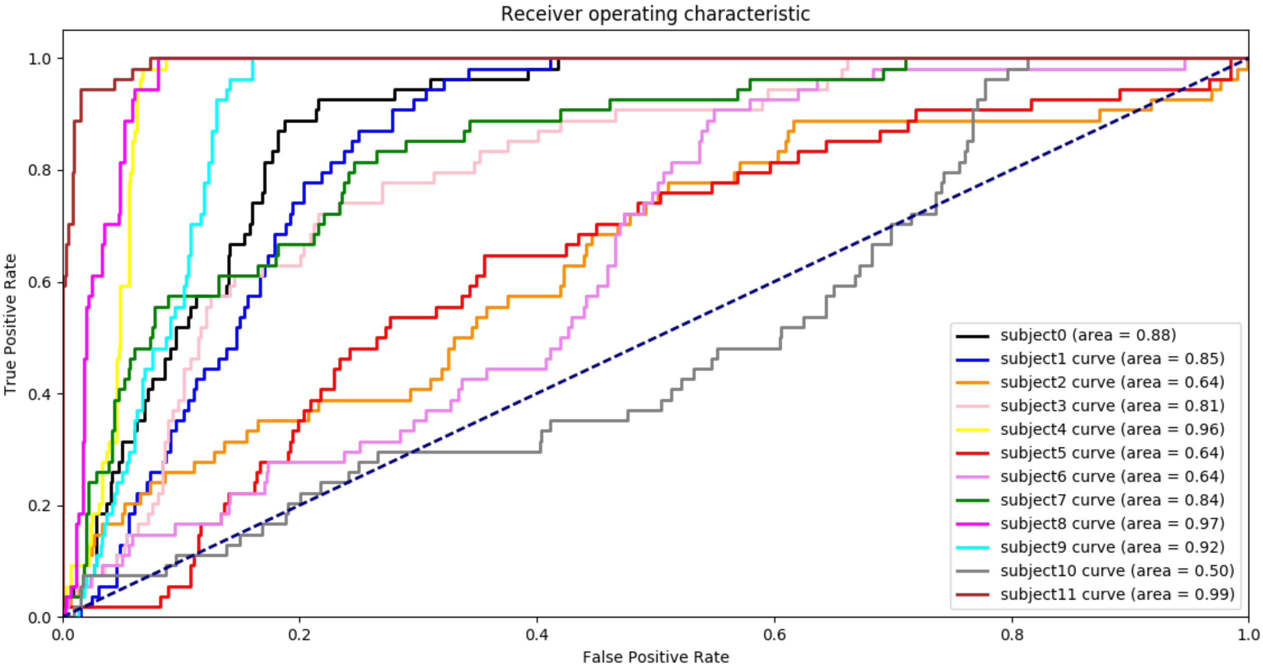


Figure 18: ROC curve for CMU data - Neural(SGD)

6 CONCLUSION AND FUTURE SCOPE

Our objective in this work has been to collect a data set, develop an evaluation procedure, and measure the performance of our classification models on an equal basis against both the CMU as well as IIITBh dataset.

From fig 2 we can see that SVC (linear) has the highest precision for IIITBh dataset.

In the process, we established that out of the four classifiers Adam Stochastic Gradient Descent, SVC(Linear), SVC(RBF), classifier SVC (Linear) performed the best on Metric Precision, as can be seen in fig 12 & 13.

From fig 14,15 and 16 , metric AUC (Area Under the Curve) we can conclude that the classifier neural (Adam) performs the best. Moreover, all of our classifiers fall in the good-excellent range as inferred form their AUC.

Our Dataset may be used for future research purposes as we have made it publicly available.

Keystroke Dynamics is a vast field with extensive scope in Computer security systems. Though the results at the moment are good, they are not good enough to completely replace physical biometric systems. But it can be used in parallel with existing systems, such that it can provide an extra layer over our current authentication systems(which generally use a combination of username and password).

Though our work only dealt with short text passwords, keystroke dynamics is being used in long text analysis as well. The future scope of our work includes:

- Adding more classification models to check their performance.
- To combine various models together. Study shows that such combination to two or more well performing classifiers can yield better results.
- The combination of the results of different algorithms for keystroke dynamics with mouse usage patterns and application usage will provide better results for identification of genuine users.

BIBLIOGRAPHY

Bryan, W. L. and N. Harter

1897 *Studies in the physiology and psychology of the telegraphic language.* Psychological Review, 4(1):27–53, 1897.

1899 *Studies on the telegraphic language: The acquisition of a hierarchy of habits.* Psychological Review, 6(4):345–375.

CENELEC

2002 *European Standard EN 50133-1: Alarm systems. Access control systems for use in security applications. Part 1: System requirements, 2002.* Standard Number EN 50133-1:1996/A1:2002, Technical Body CLC/TC79, European Committee for Electrotechnical Standardization (CENELEC).

Chang, Chih-Chung and Chih-Jen Lin

2011 "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2 (3 2011), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 27:1-27:27.

Chang, W.

2005 *Improving hidden Markov models with a similarity histogram for typing pattern biometrics*, in Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI '05), pp. 487–493, August 2005.

Crawford, H.

2010 *Keystroke dynamics: characteristics and opportunitiesg*, in Proceedings of the 8th International Conference on Privacy, Security and Trust (PST '10), pp. 205–212, August 2010.

Forsen, G., M. Nelson, and R. Staron

1977 *Jr. Personal attributes authentication techniques. Technical Report RADC-TR-77-333, Rome Air Development Center, October 1977.*

Gaines, R. S., W. Lisowski, S. J. Press, and N. Shapiro

1980 *Authentication by keystroke timing: Some preliminary results. Technical Report R-2526-NSF, RAND Corporation, May 1980.*

Gladwell, M.

2005 *Blink: The Power of Thinking without Thinking. Little, Brown and Company, New York, NY,*

Haider, S., A. Abbas, and A. K. Zaidi

2000 *A multi-technique approach for user identification through keystroke dynamics.* IEEE International Conference on Systems, Man and Cybernetics, pages 1336–1341.

Jain, Anil K. and Arun Ross

2008 *"Introduction to Biometrics". In Jain, AK; Flynn; Ross, A. Handbook of Biometrics. Springer. pp. 1–22.* ISBN 978-0-387-71040-2.

K. Killourhy, R. Maxion

2009 *Dataset*, http://www.cs.cmu.edu/~keystroke/DSL-StrongPasswordData.csv.

Killourhy, K.S. and R.A. Maxion

2000 *Comparing Anomaly Detectors for Keystroke Dynamics*, Proc. 39th Ann. Int'l Conf. Dependable Systems and Networks (DSN 09), IEEE CS, 2009, pp. 125–134.

Kingma, Diederik P. and Jimmy Ba

2014 "Adam: A Method for Stochastic Optimization," *CoRR*, abs/1412.6980, http://arxiv.org/abs/1412.6980.

Lin, D.-T.

1997 *Computer-access authentication with neural network based keystroke identity verification*, in Proceedings of the 1997 IEEE International Conference on Neural Networks, vol. 1, pp. 174–178, June 1997.

Obaidat, M. S. and B. Sadoun

1997 *Verification of computer users using keystroke dynamics*, IEEE Transactions on Systems, Man, and Cybernetics B, vol. 27, no. 2, pp. 261–269.

Pavaday, N. and K. M. S. Soyjaudah

2007 *Investigating performance of neural networks in authentication using keystroke dynamics*, in Proceedings of the IEEE AFRICON 2007 Conference, pp. 1–8, September 2007.

Peacock, A., X. Ke, and M. Wilkerson

2004 *Typing patterns: A key to user identification*. IEEE Security and Privacy, 2(5):40–47.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay

2011 "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, pp. 2825-2830.

Pin Shen Teh, Andrew Beng Jin Teoh and Shigang Yue

2013 *A Survey of Keystroke Dynamics Biometrics*, The Scientific World Journal, vol. 2013, Article ID 408280, 24 pages, 2013. doi:10.1155/2013/408280.

R. O. Duda, P. E. Hart and D. G. Stork.

2001 *Pattern Classification*, second, John Wiley & Sons, Inc., Point Roberts, Washington, USA.

Rybnik, M., M. Tabedzki, and K. Saeed

2008 *A Keystroke Dynamics Based System for User Identification*, CISIM,pp.225-230.

S. Bleha, C. Slivinsky and B. Hussien

1990 *Computer-access security systems using keystroke dynamics*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 12, pp. 1217–1222, 1990.

S. Pankanti, editors

1999 *Biometrics: Personal Identification in Networked Society. Kluwer Academic Publishers.*

S. Prabhakar, S. Pankanti and A. K. Jain

2003 *Biometric Recognition: Security and Privacy Concerns. IEEE Security and Privacy Magazine*, 1(2):33–42, March-April 2003.

Samura, T. and H. Nishimura

2009 *Keystroke timing analysis for individual identification in Japanese free text typing*, in Proceedings of the ICROS-SICE International Joint Conference (ICCAS-SICE '09), pp. 3166–3170, August 2009.

Spillane, R. J.

1975 *Keyboard apparatus for personal identification.* IBM Technical Disclosure Bulletin. 17(11):3346, April 1975.

Yu, E. and S. Cho

2003 *Novelty detection approach for keystroke dynamics identity verification*, in Intelligent Data Engineering and Automated Learning, vol. 2690, pp. 1016–1023, Springer, Berlin, Germany, 2003.

Zhao, Y.

2006 *Learning user keystroke patterns for authentication*, in Proceedings of the World Academy of Science, Engineering and Technology, vol. 14, pp. 65–70, Karnataka, India, December 2006.