REPORT OF INDUSTRIAL TRAINING

Online Signature Verification

An Industrial Training Report submitted to Manipal Academy of Higher Education in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
In
Electronics and Communication Engineering

Submitted by

Student Name: Harsh Gupta

Reg. No.: 170907514

Under the guidance of

Dr Ramya S

Assistant Professor - Selection Grade

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

MANIPAL INSTITUTE OF TECHNOLOGY

(A Constituent Institution of Manipal Academy of Higher Education)

MANIPAL – 576104, KARNATAKA, INDIA

ABSTRACT

Signature verification is one of the most widely used authentication technique with a broad spectrum of applications ranging from banking transactions to person authentication in the legal domain. In contrast to offline signatures, which uses signature as an image to process it as genuine or forged, online signatures consist of periodic time series data of pen inputs. Hence, online signatures provide flexibility in terms of features such as pen elevation, azimuth axis, and pressure at each sample point, which offline signatures simply fail to deliver. However, online signature verification comes with an additional overhead of a special input hardware to capture the data. Signatures are treated as signals with various features listed above. Online signature verification falls under classification problem since signatures are needed to be classified as genuine/ forged and multi-class classification problem to identify the person based on the input signature(if genuine). This paper focuses on the implementation of Deep Neural Network (DNN) model to classify user as well as the authenticity of the signature and compare the results of the different algorithms. A DNN is an artificial neural network which has multiple layers between the input and output layer. Each layer consists of nodes which hold a value that is mathematically calculated to learn the relationship between the input and the output. To train and test the classification model, MCYT signature dataset has been used. The dataset contains signatures & various features of 100 users.

Contents

CHAPTER I
INTRODUCTION

1.1 Introduction

Data security and privacy are one of the major issues which need to resolve in the present day digital world. There are various methods which have been developed in the past to restrict the access of data to an unauthorized person, but with the advancement in technology, these methods are not so robust anymore. Signature verification is one such area of authentication which needs to evolve. Signature Verification is a process where a signature is compared with other signature that is present in the database and checks for authenticity. This is being used by banks, intelligence agencies and high profile institution to validate the identity of a person.

A signature is a behavioural biometric which varies from person to person, which makes it very robust biometric to authenticate an individual. There are a lot of dynamic properties of a signature such as pressure, number of time the pen was lifted and the angle made by the pen, which can be used for the identification. A signature can be divided into two category – online and offline. An offline signature data is the scanned copy of the hand-written signature while the online signature data is captured in the signing process. The present method of verification either uses software which treats the signature as an image to verify the identity of the person or a handwriting expert is made to compare the signatures. By using signature as an image, we lose a lot of the dynamic property, which makes the identification of forgery tough even for an expert.

To collect the data for the online signature, we require additional hardware like a capacitive tablet or personal data assistance (PDA) which will give us the x-y coordinate, pressure and other dynamic feature reading. Using the dynamic features, we can also obtain static features such as the shape of the signature and number of points in the signature. A dynamic feature is a function of time while the static features are time-independent. The advantage of having the dynamic feature is that even if the forger is able to produce a similar-looking signature, it will be nearly impossible to provide the same dynamic feature such as the pressure and pen elevation. This makes the authentication even more robust since this data would not be available to the forger.

Signature verification can be divided into two-sub task, recognition and then verifying if the signature is genuine or not. Recognition is the part where we compare the given signature with the rest of the data to identify the user, whereas the verification part is to verify if that signature belongs to that individual or not. In the verification part, we have two types of error –

false rejection and false acceptance error. False rejection is the case where we reject the actual user while false acceptance is the case where we accept the forged identity. The average error rate is the average of false acceptance rate (FAR) and false rejection rate (FRR). The average error rate should be as close as to zero and also FAR should be minimized as much as possible since giving access to an unauthorized identity can lead to greater problems.

1.2 Motivation

The motivation for working on an online signature verification model revolves around security and data privacy of individuals. The conventional offline signature does not provide the dynamic features that online signatures do. The dynamic feature captured during the signature process increases the difficulty to forge as the forger would not have access to these properties and features.

1.3 Organization of Report

The report has been divided into five chapters. The first chapter gives an introduction to the topic – "Online signature verification" and also provides the organization of the report. The second chapter presents the background theory and the earlier work done in the field. It also provides a brief background theory about the project and the summary of the literature review. Chapter three provides the in detail information about the methods and tools used for the project. The fourth chapter provides the details about the results achieved and their analysis. The fifth and the last chapter present the summary of the work done, results achieved and also the future scope of the project. References follow this final chapter.

CHAPTER II
LITERATURE REVIEW

2.1 Background Theory and Previous works

Signature verification essentially means to compare two signatures and draw similarities between them to conclude whether the signature is genuine or not. The easiest and most straightforward way to match two signatures is to compute the correlation between them. The drawback over here is that, this is a point to point comparison, and since the signature varies for even the same person dues to rotation, translation or scaling, this method doesn't work very well. The decision is made by comparing the similarity score and a predefined threshold score. Online signature captures a lot of dynamic properties which are basically time-series data. Each series provides data on the action performed during the process of signing. As the signature of same person varies due to expansion, compression, rotation or translation, a straightforward method such as different distance metrics and autocorrelation are not the best means to calculate similarity or dissimilarity value. To overcome this problem, non-linear approaches and techniques such as Dynamic Time Warpping (DTW) algorithm and Hidden Markov Models (HMM) are commonly used in aligning two signatures. After the advancement in the field of Neural Networks and Machine Learning, methods such as Deep Neural Network (DNN), Support Vector Machine (SVM) algorithm and K-Nearest Neighbor (KNN) algorithm have also been used.

Another factor that affects the output is the features that are being used to train the model. Gupta and Joyce [1] studied the effect of using six global features - total time, the total pen-up time, number of velocity sign changes in the x- and y-direction, number of acceleration sign changes in the x and y coordinate. There are various other distance metrics mentioned in [2] that can be used to compute the distance between the given signature and the stored signature. The most common method to find the similarity between the input feature vector and the stored template is to use a classical distance measure such as Euclidean distance, Canberra Distance, Euclidean Distance, City Block distance and Hamming Distance. Kim [3] studied 75 features and selected the best combination of features for each individual by measuring the Degree of Difficulty to forge. Taherzadeh [4] extracted 30 features from the online signature and categorized them into four categories- Direction-related, Pressure-related, Dynamic-related and Shape-related. David Aristizábal [5] proposed a system that extracts 42 features from seven time-series functions by applying 6 non-linear dynamic techniques to each series.

One of the most successful and also one of the simplest methods used for verification is DTW [6, 7]. DTW is an algorithm which calculates the similarity between two sequences which may vary in time or speed. DTW finds the best non-linear alignment of two vectors such that the overall distance between corresponding vector elements is minimized in the least square sense.

In DTW, the input and reference signatures are compared by using dynamic programming (DP) matching algorithm which can manage the variability on the signature's length. DTW is used to accommodate the timing but not the X and Y coordinates of the signature. By applying DTW to X and Y signals, it will distort the shape of the signature.

Hidden Markov Models (HMM) were introduced in the pattern recognition field as a robust method to model the variability of discrete-time random signals where time or context information is available [8]. HMMs are widely used in the classification of input patterns, and they have a remarkable ability to model stroke-based sequences. These models have found to be well suited for signature modelling since they are highly adaptable to individual variability [9].
SVM is widely used for classification and regression problems. SVM uses a kernel function to find similarity or to classify data into different groups. A study to compare different classifier, which included SVM was conducted by Yenwei Lee [10] in the area of Signature verification. The best result achieved by an SVM classifier was 95.8% using Gaussian Radial Basis Function Kernel method. A later work by Marianela Parodi and Juan Carlos [11, 12] compared SVM and Random Forest Classifier which showed that the later performed better than the results achieved by SVM.

A neural network model requires both genuine and forged data. Since forged data can't be specific, a simple solution to generate this data is by deforming the original genuine data. A neural network-based approach proposed by Tarig Osman [13], used velocity segmentation and autoregressive vector modelling. A work done by Babita P [14], showed that the accuracy of the neural network-based model decreased as the number of users in the dataset increased. An accuracy of 98% was achieved when there were only ten users as compared to 40.5% when there were 100 users.

2.2 Summary

In this paper, we will be implementing Deep Neural Networks (DNN) model with different parameters. A new and relatively easier method to pre-process the data has been proposed to improve the time taken by the model to train and predict. Experimentation has been done with hidden layers to get the optimal number of layers to be used in the Neural Network model. The trained model has been tested with a different number of users to see how the accuracy is affected. The proposed method decreases the data required by the model as compared to the offline signature method, which treats the signature as an image, therefore reducing the time needed for the model to predict.

CHAPTER III
METHODOLOGY

3.1 Dataset

MCYT dataset [15] has been used, which consists of 5000 dynamic signatures, collected from 100 users. Out of 5000 signatures, 2500 are genuine, and the rest 2500 are skilled forgeries. The data set provides the following discrete-time series data of each signature: i) x-axis position, ii) y-axis position, iii) pressure applied, iv) azimuth angle of the pen and v) pen inclination (Fig 1). The sampling rate at which the data has been captured is 100Hz. Using the given data, we have extracted additional features such as the time taken for signing, number of pen ups, average pressure, the ratio of the signature, velocity in x and y-direction.

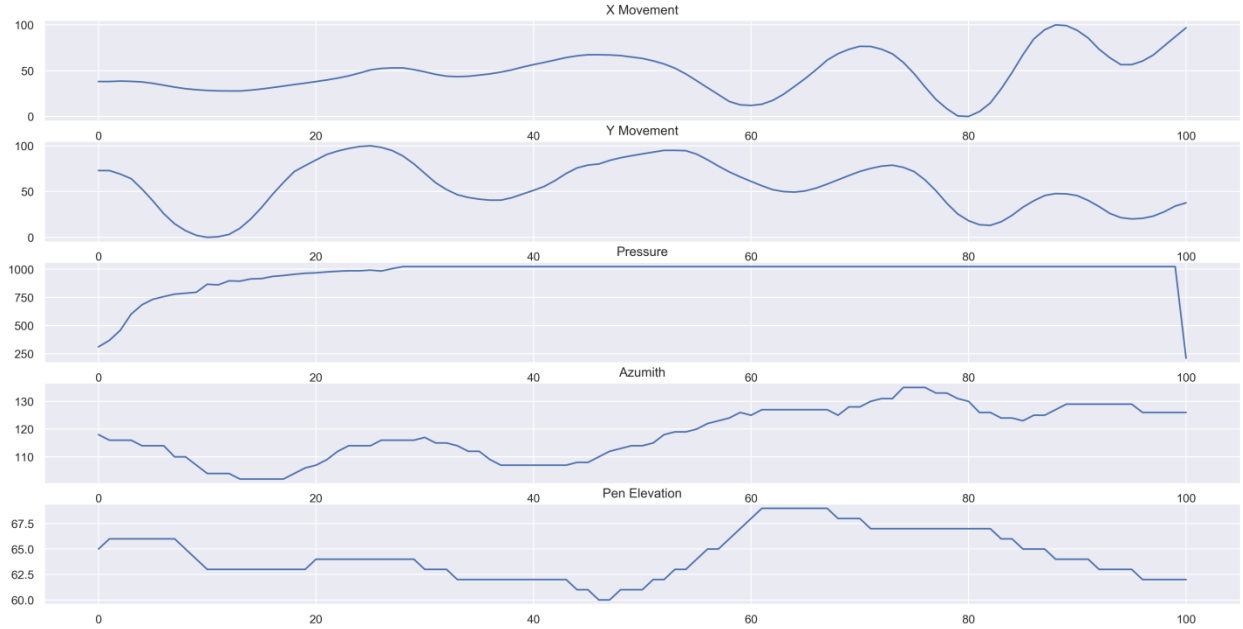

Fig 1 Features given in Dataset

3.2 Data Pre-Processing

Since everyone has a unique signature, all the signatures have a different number of data points which cannot be directly used for training the model. All the signatures must have the same number of data points. To make the sign of equal length, either the signs could be up-sampled to the signature having the highest data points or down-sampled to the signature having the least number of data points. Both methods have their own advantages and disadvantages. In this paper, the disadvantages of both the methods have been tried to be eliminated. As we would be losing points while downsampling, we need to take care that the signature doesn't lose its characteristics and while up-sampling we would be adding more points, so that might

add noise and also change the characteristic of the signature.

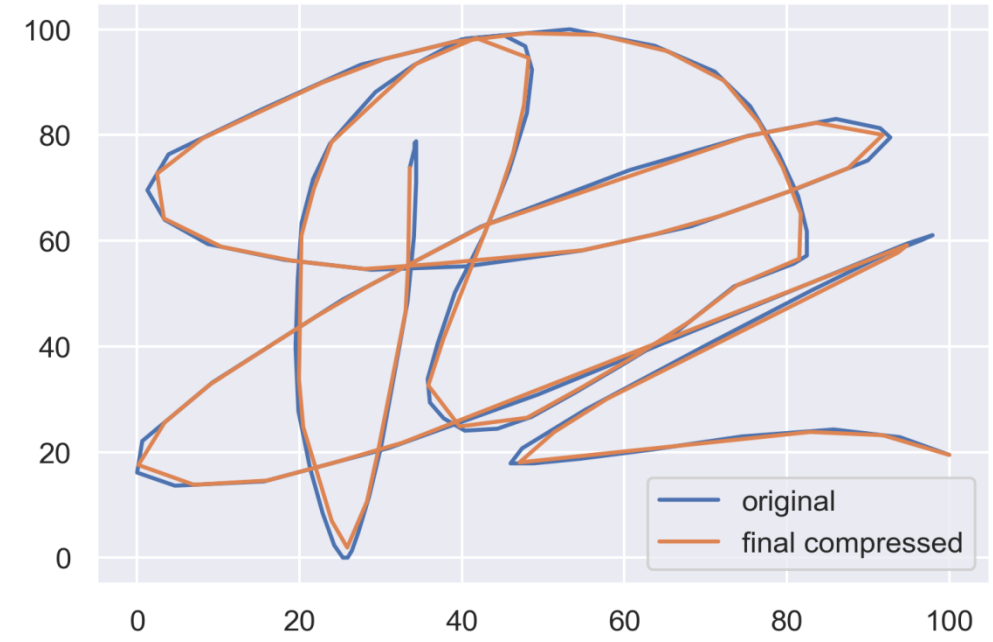

Fig 2 Orignal vs Final Compressed

First, the Ramer–Douglas–Peucker (RDP) algorithm has been applied on the signature to only keep the most significant data points in each signature while also maintain the shape and characteristics of the signature. The length of each signature after this preprocessing will still be varying but the number of points is reduced by a significant amount. As seen in the Fig 2.1 , the original signature consists of 1000 points, which was reduced to only 122 points after RDP compression while maintaining the signature curves and characteristics. Now to make all the signature of equal length, the signature with the highest number of points after RDP is selected and all the signatures were then upsampled to that particular number of points. This pre-processing made all the signatures of equal length while maintaining its characteristics and at the same time reducing the data [Fig. 2.2]. All the data was normalized so that all the data points were in the same range and their magnitude is also reduced.

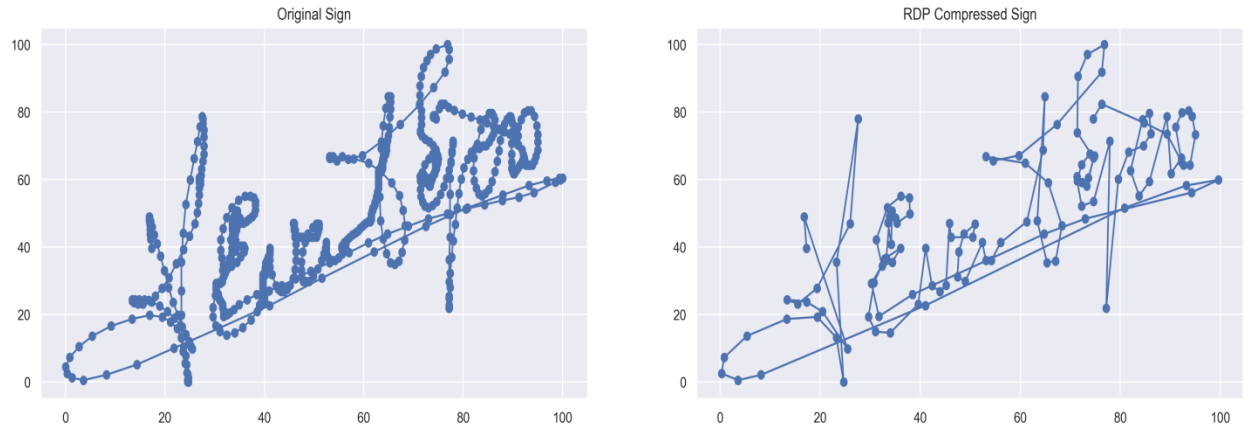

Fig 3.1 RDP Compressed signature

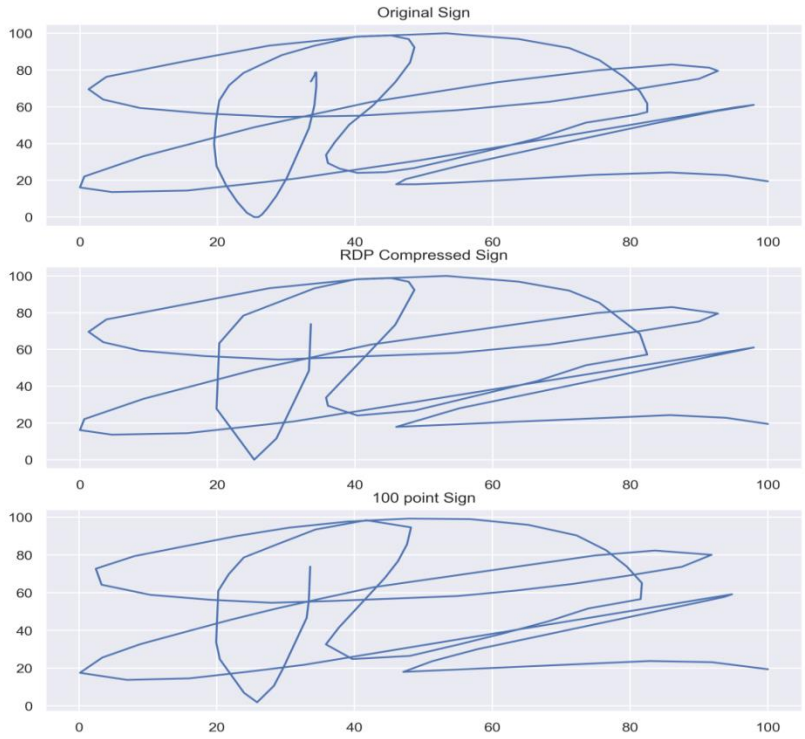

Fig 3.2 Equidistant and Normalized signature after RDP compression

Ramer–Douglas–Peucker (RDP) algorithm

The Ramer-Douglas-Peucker is an algorithm used for reducing the number of points in a curve that is approximated by a series of points. It is a recursive algorithm which keeps the first and the last point in the series and then finds the farthest point from the line formed using the first and last point. This farthest point from the line is kept because it is a point which defines the curve of the sequence. The algorithm uses a parameter Ɛ which is the distance threshold for the algorithm and set by the user. Any point in this threshold range which is not marked to be kept will be removed from the sequence. The algorithm runs recursively between the first and the farthest point and then between farthest and the last point until the base condition is reached. Once the algorithm process is completed, a new simplified sequence with only significant point is generated. Fig 3 shows an example of how RDP algorithm works.

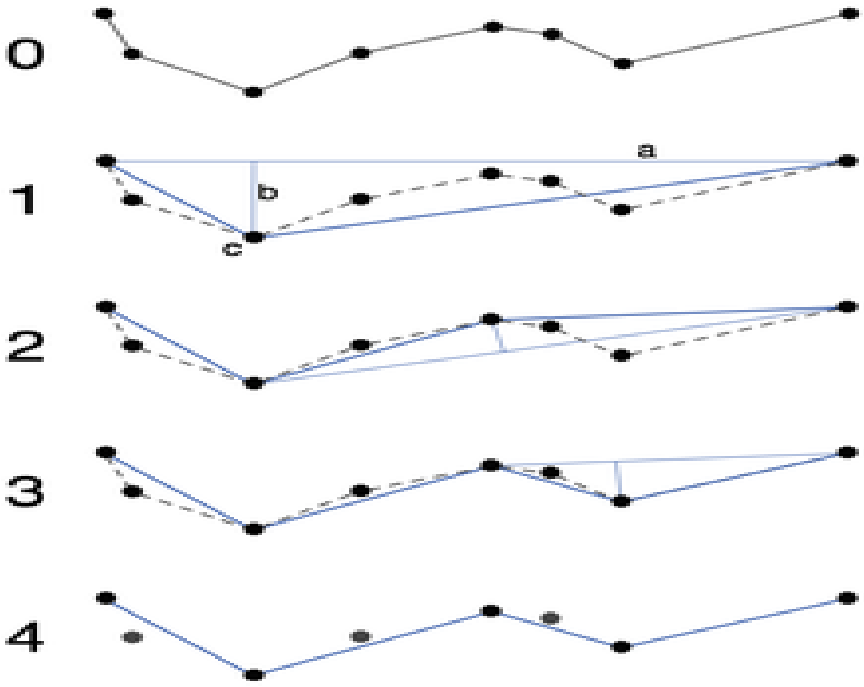

Fig 4 RDP Algorithm

Additional Derived Features

Using the dataset provided, some addition static features have been derived to improve the training data for the model. Features added to dataset include:

- *Ratio*: This is the length to width ratio of the signature
- *Number of times pen lifted*: The number of time the pen is lifted up while signing
- *Time taken to complete the sign*: The duration taken to complete the signing process.
- *Average Pressure*: The average pressure applied by the user during the signing process.
- *Average Speed*: The average speed maintained by the user during the signing process.

3.3 Neural Network Model

The main focus of this paper is to find the performance of a Neural Network with the proposed pre-processing method. The TensorFlow library in Python has been used to create the Deep Neural Network model for this work. Two different models has been implemented – one for user identification and one to find the authenticity of the signature. The same dataset with minute changes has been used to train both the models. Experimentation with the parameters of neural network such as number of layers, number of nodes, learning rate, activation function, loss function and optimizer, has been done to come up with the best set of parameters to achieve higher accuracy.

Our dataset consists of 25 genuine and 25 forged signature of each user. The dataset was divided into training, testing and validation set with training set having 15 each of genuine and forged data, and testing and validation set have 5 each of genuine and forged signature data. A total of ten features have been used to train the models to recognize the pattern in the signature, with each signature having 505 data points. A number of models were made to compare and study the effect of different parameters on the final output. The neural network model for the user identification consists of an input layer having 505 nodes, 3 - 6 hidden layers with a varying number of nodes and an output layer with 100 nodes. As an experiment to study the effect of different number of user on the final accuracy, the output layer was varied from 10 to 100 nodes. Different number of hidden layers was used to find the optimal number of layers for the model. The activation function used in the hidden layer was Rectified Linear Unit (ReLU), and for the output, layer Softmax was used. The initial learning rate of the model was set to 0.0005, the optimizer used was Adams, the loss function used was Sparse Categorical Crossentropy and the number of the epoch was set to 100. For the authentication model, an input layer of 505 nodes, 3 - 5 hidden layers with varying nodes and a single node output layer was used. The neural network architecture for this model used ReLU as the activation function for the hidden layer and sigmoid activation function for the output layer. A sigmoid activation function gives a value in the range of 0 to 1 as a confidence score - closer to 1 being highly confident and closer to 0 being not at all confident. This model used Adam as the optimizer, binary crossentropy as the loss function, an initial learning rate of 0.0001 and the number of epochs set to 100. The model was trained and tested upon different batches of users to make sure that the model performs well for all the users.

CHAPTER IV
RESULT ANALYSIS

The result obtained from the proposed methodology, mentioned in the previous section is presented over here. The results were evaluated on different splits of the 5000 signatures present in the MCYT dataset. The authentication model took 150 to 200 seconds for 100 epochs, i.e. 1-2 second for each epoch, to train upon roughly 3300 samples with each sample having 505 data points. A similar training time per epoch was observed for user identification model as well on 1747 samples.

```
Train on 3337 samples, validate on 249 samples
Epoch 1/100
3337/3337 [==============================] - 2s 580us/sample - loss: 789.6051 - acc: 0.5133 - val_loss: 174.2540 - val_ac
c: 0.4980
Epoch 2/100
3337/3337 [==============================] - 1s 394us/sample - loss: 363.2390 - acc: 0.5430 - val_loss: 151.7259 - val_ac
c: 0.6185
Epoch 3/100
3337/3337 [==============================] - 1s 403us/sample - loss: 199.6161 - acc: 0.5799 - val_loss: 71.2595 - val_ac
c: 0.5863
Epoch 4/100
3337/3337 [==============================] - 1s 406us/sample - loss: 145.4039 - acc: 0.6005 - val_loss: 98.0864 - val_ac
c: 0.5301
Epoch 5/100
3337/3337 [==============================] - 1s 407us/sample - loss: 44.6781 - acc: 0.6191 - val_loss: 152.6038 - val_ac
c: 0.6426
```

Fig 5 Training time recorded for User authentication model

```
Train on 1747 samples, validate on 749 samples
Epoch 1/150
1747/1747 [==============================] - 3s 1ms/sample - loss: 2467.1087 - acc: 0.0143 - val_loss: 736.5818 - val_ac
c: 0.0187
Epoch 2/150
1747/1747 [==============================] - 1s 502us/sample - loss: 450.6239 - acc: 0.0372 - val_loss: 272.4848 - val_ac
c: 0.0427
Epoch 3/150
1747/1747 [==============================] - 1s 495us/sample - loss: 240.4409 - acc: 0.0452 - val_loss: 183.0451 - val_ac
c: 0.0240
Epoch 4/150
1747/1747 [==============================] - 1s 495us/sample - loss: 194.5648 - acc: 0.0761 - val_loss: 225.1504 - val_ac
c: 0.0654
Epoch 5/150
1747/1747 [==============================] - 1s 496us/sample - loss: 141.2507 - acc: 0.0750 - val_loss: 134.2406 - val_ac
c: 0.0761
```

Fig 6 Training time recorded for User identification model

4.1 User Identification Model

The number of hidden layers used while training the model was varied to find the optimal of layers to be used in the User Identification Model. The final model consisted of six hidden layers with 400, 350, 250, 200, 150 and 125 nodes and output layer having 100 nodes. Table 1 shows a comparison between the accuracy and number of hidden layers used.

Number of Hidden Layers	Training Accuracy	Validation Accuracy	Test Accuracy
2	14%	7.5%	6.4%
3	100%	67%	66.2%
4	100%	85.5%	85%
5	100%	89.5%	87%
6	99.6%	92.3%	90%

Table 1

The final model was tested with a different number of users to measure the accuracy across different sets of the user. Table 2 shows the results obtained over multiple iterations and fig 7 and fig 8 show the confusion matrix for some of the testing results.

Number of Users	Accuracy
10	95%
20	92%
50	87%
100	85%

Table 2

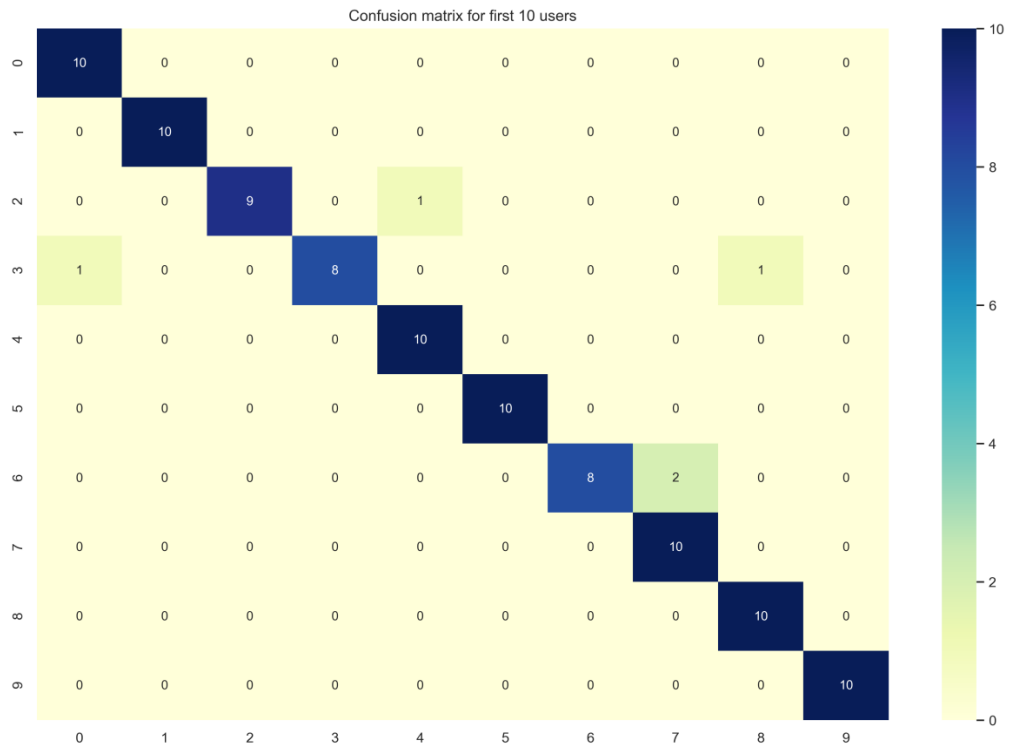

Fig 7 Confusion matrix for 10 user

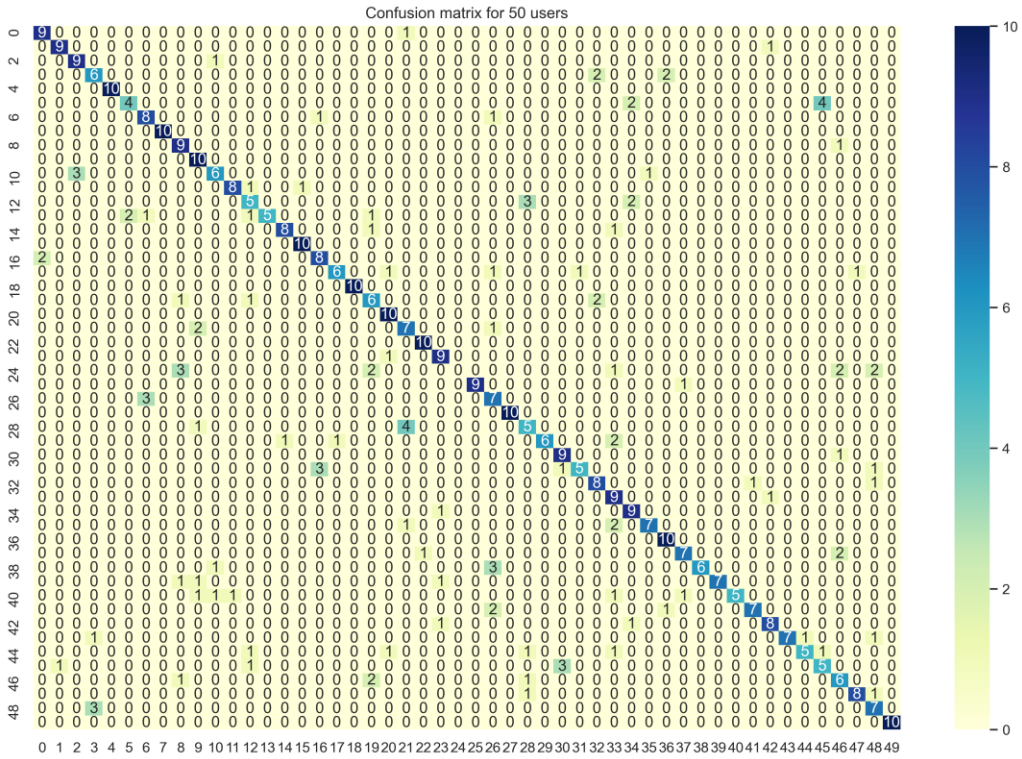

Fig 8 Confusion matrix for 50 users

4.2 User Authentication Model

The final model used six hidden layers with 350, 250, 200, 150, 100 and 50 nodes which gave an average testing accuracy of 83-85%. Figure 9 shows the confusion matrix for the authentication model. The results achieved are mentioned below

- Accuracy : 83-85%
- False Acceptance Rate: 6-8%
- False Rejection Rate : 8-10%

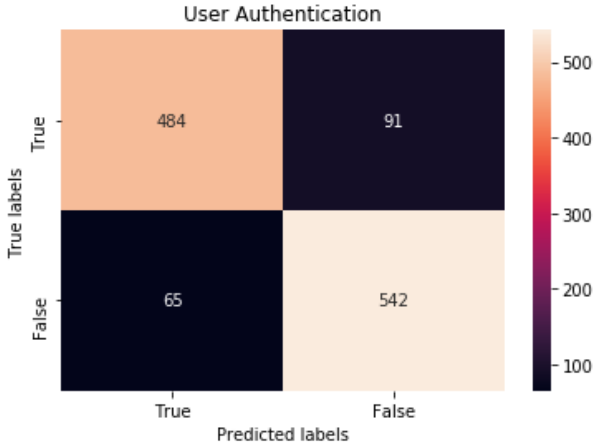

Fig 9

CHAPTER V
CONCLUSION

5.1 Conclusion

Online Signature Verification is one of the most widely used authentication systems in banking and person authentication in the legal domain. Online signature provides a whole lot of features which makes it a very robust method to verify the authenticity of a person. In this paper, we have proposed an online signature verification model using deep neural network and a pre-processing method which reduces the training and prediction time. The user identification model gave us a testing accuracy of 90% on 100 users, and the user authentication model gave us a testing accuracy of 85%.

5.2 Future work

There is a scope of improvement in the model with respect to accuracy. The current model has a False Acceptance Rate of about 6 to 8 per cent which can lead to a security breach. Additional features like jerks, velocity in x and y direction, acceleration etc. can be derived from the given data and used for training the model to improve the accuracy. Experimenting with hyperparameter tuning can also fetch us better results.

REFERENCES

[1] G. K. Gupta, R. C. Joyce, "A STUDY OF SOME GLOBAL FEATURES IN ONLINE HANDWRITTEN SIGNATURE VERIFICATION",Faculty of information technology, Monash University, 2006.

[2]G. K. Gupta, "The State of the Art in On-line Handwritten Signature Verification", Monash Univ., Australia, 2006.

[3] S. H. Kim et al., "applying personalized weights to a feature set for on-line signature verification" in proceedings of the Intl.Conference on Document Analysis and Recognition, Vol. 2, PP. 882-885, 1995

[4] G. Taherzadeh, R. Karimi, A. Ghobadi, P. Vahdani Amoli and S. Mirjalili, "Categorizing Global and local features of On-line signature verification using DTW and Fuzzy logic", Multimedia University, Selangor, Malaysia.

[5] D. Ahmedt-Aristizabal, E. Delgado-Trejos, J.F.V. Bonilla, and J.A. Jaramillo-Garzón, "Dynamic signature for a closed-set identification based on nonlinear analysis", in Proc. IJCB, PP.1-8, 2011.

[6] D. Sakamoto, H. Morita, T. Ohishi, Y. Komiya, and T. Matsumoto, "On-line signature verification algorithm incorporating pen position, pen pressure and pen inclination trajectories", Proc. ICASSP, PP. 993–996, 2001.

[7] M. Faundez-Zanuy, "On-line signature recognition based on VQ-DTW", Pattern Recognition, Vol. 40, No. 3, PP.981-992, 2007.

[8] J. Fierrez-Aguilar, J. Ortega-Garcia, D. Ramos, and J. G.-Rodriguez, "HMM-based on-line signature verification: Feature extraction and signature modelling", Pattern Recognit. Lett.,Vol. 28, No. 16, PP. 2325–2334, Dec. 2007.

[9] D. Impedovo and G. Pirlo, "automatic signature verification: the state of the art", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. 38, NO. 5, PP. 609-635, SEPTEMBER 2008.

[10] Y. Lee, "developing a biometric authentication system using dynamic signature verification statistical learning and soft computing approaches", PHD thesis, Capella University, Nov. 2004.

[11] M. Parodi and J. C. Gomez, "Online Signature Verification Based on Legendre Series Representation. Consistency Analysis of Different Feature Combinations", CIARP 2012, LNCS 7441, PP. 715–723, 2012.

[12] M. Parodi, J. Gómez and M. Liwicki, "Online Signature Verification Based on Legendre Series Representation. Robustness Assessment of Different Feature Combinations", ICFHR, 2012.

[13] O. T. Abd-Elgadir-Mohammed, "a neural-network-based online signature verification system using vector autoregressive modeling and a novel velocity segmentation scheme", PHD thesis, Univ. of Detroit Mercy, 2009.

[14] Babita P "Online Signature Recognition Using Neural Network". J Electr Electron Syst ,2015

[15] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero and Q.-I. Moro, "MCYT baseline corpus: a bimodal biometric database", IEE Proc.-Vis. Image Signal Process., Vol. 150, No. 6, December 2003