Online Signature Forgery Prevention

Shalini Bhatia
Associate Professor,
TSEC, Bandra (W),
Mumbai-50, India.

Pratik Bhatia
B.E. (Comps)
TSEC, Bandra (W),
Mumbai-50, India.

Dheeraj Nagpal
B.E. (Comps)
TSEC, Bandra (W),
Mumbai-50, India.

Sandhya Nayak
B.E. (Comps)
TSEC, Bandra (W),
Mumbai-50, India.

ABSTRACT

Personal identity verification is of great importance in today's world which is full of frauds and forgeries. A signature being a very good biometric has been since a long time for personal identity verification. Signature verification is the process used to recognize an individual's handwritten signature to prevent fraud. In this project, the dynamic features of the signature are considered so as to prevent forgeries. The pressure at the pen-tip together with the x and y coordinates of the signature are measured and features extracted from these are used to verify the signature. A signature verification system using Error Back Propagation Training Algorithm is designed using Neural Network Toolbox of MATLAB to verify the signatures. The attractive features of this system are its low cost, low intrusion, good performance and use of an acceptable and natural biometric (the signature). A two-step method is proposed, which involves identification of the signature in the first step followed by individual verification. Both the steps are carried out by Neural Networks trained using Error Back-Propagation Training Algorithm.

Keywords

Signature, Biometrics, Error Back-Propagation Training Algorithm

1. INTRODUCTION

Personal identification is the process of associating a particular individual with an identity. Identification can be in the form of verification (also known as authentication), which involves authenticating a claimed identity, or recognition (also known as identification), which involves determining the identity of a given person from a database of persons known to the system. [1] Two traditional techniques widely used for automatic personal identification are knowledge-based and token-based approaches. [2] Token-based approaches use something a person has to make a personal identification, such as a passport, driver's license, ID card, credit card, or keys. Knowledge-based approaches use something a person knows to make a personal identification, such as a password or a personal identification number (PIN). However, these authentication methods are vulnerable as the person may lose the material or the information required to authenticate him/her.

Biometrics has come up as an alternative to these physical objects of authentication. Various kinds of biometrics are used nowadays and authentication methods based on recognition of iris, finger print and face are not uncommon. One such biometric that is widely used nowadays is the signature. Signature verification is one of the oldest means of identity validation and has been accepted widely while other methods have been associated with criminal investigation.

A reliable Signature Verification Technique can be applied in different areas of security applications. Handwritten signatures are already accepted as a biometric and they continue to play an important role in many domains.

2. REVIEW OF LITERATURE

Signature authentication methods have been in use since many years and the research to improve the existing techniques is still in progress. Generally speaking, signature authentication methods can be divided into two categories, viz., Offline methods and Online methods.

Offline signature authentication methods are based on images of the signature captured after the entire signature has been written down. Thus, only the static features of the image such as the shape, size and curves in the image of the signature are visible and hence, these features are the only ones that can be used for verification.

Online signature authentication methods use pen pressure pads to capture each and every point of the signature. Thus, at every point, using a tablet, the x and y co-ordinate of the point, the pressure applied while signing, the timestamp, and the azimuth and altitude angles at which the pen was held are recorded. These dynamic features are further used for verification of signatures.

One of the most common techniques of signature verification involves the use of Image Processing techniques. Liu Dong, et.al [3] have presented a novel method based on template matching approach and support vector data description to solve the difficulty of how to choose the verification threshold in signature verification. Han, Chang, et.al [4] have proposed an approach based on multiple template matching algorithms to identify the individual via few training samples.

Dynamic time warping (DTW) is an algorithm for measuring similarity between two sequences which may vary in time or speed. Based on dynamic programming, Dynamic Time Warping algorithm finds an optimal match between two sequences of feature vectors by allowing for stretching and compression of sections of the sequences. Daramola and Ibiyemi [5] have developed an online signature verification system based on Dynamic Time Warping.

A SVM is a classifier derived from Statistical Learning Theory first presented in [6]. The problem that SVM try to solve is to find an optimal hyper-plane that correctly classifies data points by separating the points of two classes as much as possible. Kour, Hanmandlu, et.al [7] have developed an online signature verification system using Genetic Algorithms (GA) and SVM. Fauziyah, Azlina, et.al [8] have also developed an online signature verification system using SVM and VBTablet 2.0.

The approach of Justino et.al [9] uses the graphometric features, that is, static features like the density of pixels and the pseudo dynamic features represented by axial slant. A Hidden Markov Model is used for the learning and verification process.

This paper presents an Online Signature Forgery Prevention system created using Neural Networks which are trained using Error Back-Propagation Training Algorithm. Verification of signatures is performed using their static as well as dynamic features.

In the following sections, the approach taken in building the Online Signature Forgery Prevention System is discussed. Section 3 provides an overview of the system. In section 4, the signature acquisition and pre-processing techniques are discussed followed by feature selection and extraction in section 5. Section 6 explains the implementation of the system. The results are presented in section 7 and finally, conclusions are given in section 8.

3. OVERVIEW OF THE PROPOSED SYSTEM

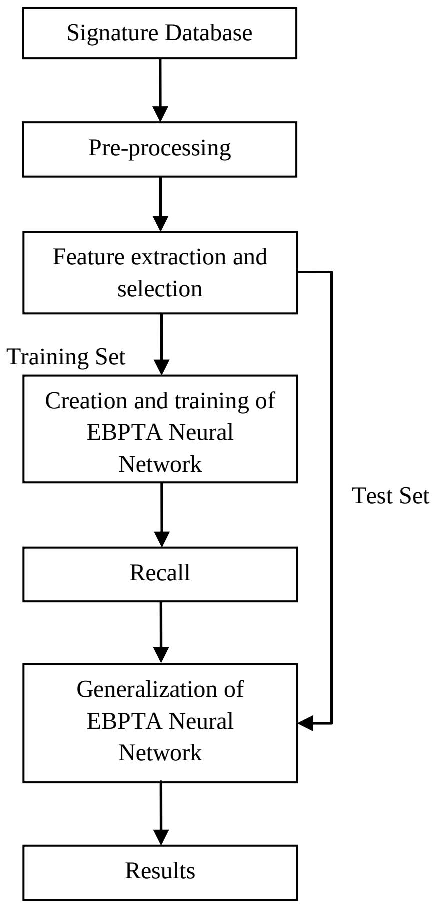
Signature Database
Pre-processing
Feature extraction and selection
Training Set
Creation and training of EBPTA Neural Network
Test Set
Recall
Generalization of EBPTA Neural Network
Results

Figure 1: Proposed System

Figure 1 illustrates the Online Signature Forgery Prevention System. The six blocks in the figure represent the six modules in the Online Signature Forgery Prevention System. The main functions of these blocks are as given below:

1. Database Acquisition: To acquire both original and forged signature samples in order to train the network.

2. Pre-processing – To pre-process the signatures and make them ready for feature extraction.

3. Feature extraction and selection – To extract features from the signatures which will be provided as input to the Neural Networks.

4. Creation and Training of EBPTA Neural Network – To create a Neural Network which trains according to the Error Back Propagation Training Algorithm and perform training of the network.

5. Recall – To test the performance of the Neural Network when the data provided as input is the same as on which the network is trained.

6. Generalization of the EBPTA Neural Network – To test the performance of the Neural Network when the data provided as input is other than that on which the network has been trained.

In the proposed Online Signature Forgery Prevention system, the authentication shall be performed in two steps. The first step is ‘Identification’. In this step, the test signature will be classified to a target user. If the expected user is same as the target user, the second step is performed, which is ‘Verification’. In this step, the signature is classified as original or forged.

The classification decisions for each of the steps are carried out by respective Neural Networks assigned for the two steps discussed above. In this paper, the Neural Network which is trained for the first step is referred to as ‘Identification Neural Network’. This is a common neural network for a set of individuals, the size of which, in case of the samples considered for this paper, is 20. For the next step, the Neural Network trained is called as ‘Individual Verification Neural Network’. This network, as the name suggests, is specific for every individual. Both the networks, namely, ‘Identification’ and ‘Individual Verification’ shall undergo through steps 4, 5 and 6 given above.

4. SIGNATURE ACQUISITION AND PRE-PROCESSING

This section explains the two signature databases that were used for the system. The first database was acquired from the website of SVC 2004 [10] where it was provided for the purpose of competing in a signature verification competition. The second database was created for the purpose of this system. The signatures were captured using a digital pen tablet and information of each signature was stored in a separate text file.

4.1 Acquired Signature Database

A database of 1600 [10] signatures, comprising of 40 signatures of 40 different users, was acquired. These 40 signatures of each user, in turn, constitute 20 original signatures and 20 skilled forgeries.

Each signature contains multiple tokens. A sample signature has been shown in Figure 2, with the tokens in the signature given in Table 1. Each token consists of a tuple having seven entries at each point in time k given by

$$s(k) = [\, s_1(k)\ \ s_2(k)\ \ s_3(k)\ \ s_4(k)\ \ s_5(k)\ \ s_6(k)\ \ s_7(k)]^T$$

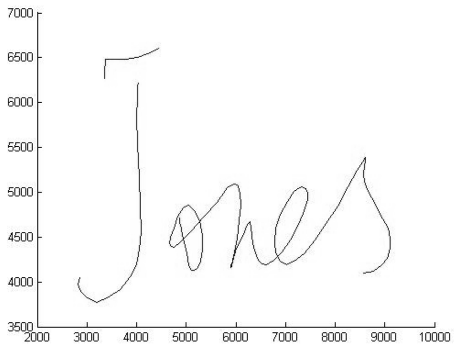

Figure 2: Signature of User 1 of Acquired Signature Database

Table 1. Signature of User 1 of Acquired Database as stored in .txt file

X	Y	Time-stamp	Pen Status	Azi-muth	Alti-tude	Pressure
327	638	31309	0	1510	370	495
336	626	31310	1	1510	370	499
336	632	31311	1	1520	380	517
337	648	31312	1	1520	380	512
. . .	. . .	. . .	. . .	. . .	. . .	. . .
857	410	31447	1	1590	570	768
837	410	31448	1	1590	570	312

4.2 Created Signature Database

A database constituting of 40 signatures each of 20 users was created in order to be used in the system. These signatures were collected using a Wacom Bamboo Touch tablet and the Java Pen Tablet Access (JPEN) library. The 40 signatures of each user consisted of 20 original signatures and 20 skilled forgeries.

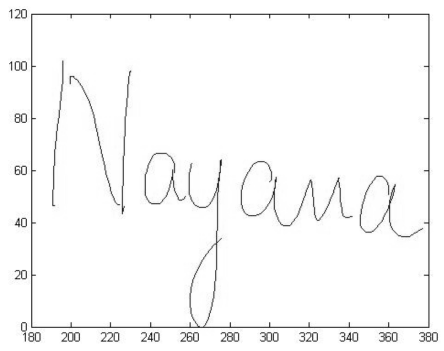

Figure 3: Signature of User 1 of Created Signature Database

Here, s_1 denotes the pen-tip x coordinate, s_2 denotes the pen-tip y coordinate, s_3 denotes the timestamp, s_4 denotes the button status, s_5 denotes the azimuth angle, s_6 denotes the altitude angle and s_7 denotes the pressure at the pen-tip p.

Each signature contains multiple tokens. A sample signature has been shown in Figure 3, with the tokens in the signature given in Table 2. Each token consists of a tuple having five entries at each point in time k given by

$$s(k) = [\, s_1(k)\ \ s_2(k)\ \ s_3(k)\ \ s_4(k)\ \ s_5(k)\,]^T$$

Here, s_1 denotes the pen-tip x coordinate, s_2 denotes the pen-tip y coordinate, s_3 denotes the timestamp, s_4 denotes the button status and s_5 denotes the pressure at the pen-tip p.

Table 2. Signature of User 1 of Created Database as stored in .txt file

X	Y	Timestamp	Pen Status	Pressure
195.899	85.08057	29794395	1	0.17399804
195.946	85.08057	29794403	1	0.22580644
195.946	85.02453	29794410	1	0.25513196
195.994	85.02453	29794418	1	0.27468230
. . .	. . .	. . .	. . .	. . .
377.019	149.4916	29799503	1	0.54936460
379.053	148.5387	29799511	1	0.29130010

4.3 Pre-processing

On analyzing the obtained dataset, it was observed that timestamps for few consecutive points for some signature samples were the same. This is an anomaly because the pen tip cannot be at two places on the tablet surface at the same time. Because of this anomaly, the values like velocity and acceleration, which are dependent on time interval, were calculated as infinite.

Another anomaly observed in the dataset obtained using JPen and Wacom Bamboo Touch was that there were points which were sampled after the last point of the signature had been obtained. Elimination of those extra points which were not part of the signature was necessary.

The steps carried out to take care of the anomalies mentioned above are as follows:

Step 1: For all signatures samples, remove the tokens at the end of the signature sample which have pressure applied (or pen status) equal to 0.

Step 2: For all signature samples, if the timestamp for any two consecutive tokens is the same, then make the timestamp value of the second token equal to the average of the timestamp values of the tokens above and below it.

5. FEATURE SELECTION AND EXTRACTION

Feature selection plays an important role in the performance of the system. Naive inclusion of features may not improve system performance. The elements that compose the Feature Vector should be such that classification accuracy of the neural network is maximized.

In case of a neural network classifier for a Signature Verification System, the following properties should be exhibited by an ideal Feature Vector:

i. High intra-user similarity
ii. Low inter-user similarity
iii. Highlighting the differences between an original and a forgery
iv. Low redundancy

Keeping the above properties in mind, the following features were included in the Feature Vector:

i. Average X
ii. Average Y
iii. Average Pressure
iv. Standard Deviation X
v. Standard Deviation Y
vi. Standard Deviation Pressure
vii. Average Velocity
viii. Average Acceleration
ix. Average Angular Velocity
x. θ mean
xi. θ median
xii. θ Standard Deviation
xiii. r mean
xiv. r median
xv. r Standard Deviation
xvi. Sections

These sixteen features along with the ID of the user, who the claimant claims to be, will be the inputs to the neural network. Thus, this list of seventeen features extracted from the signatures was stored in a MATLAB .mat file and served as the feature vector database to be used in the training and testing of the neural network.

The aforementioned features can be classified into two categories as mentioned before, online and offline.

5.1 Offline Features

As mentioned before, the features which can be obtained even after the user has signed the signature are called as offline features. The offline features here will be the representation of all the points of a signature sample in the X-Y plane.

There are two ways to represent a point in the plane:

i. Cartesian co-ordinate system
ii. Polar co-ordinate system

Both these representation systems have been used in the construction of this system where ever applicable.

5.1.1 Cartesian co-ordinate system

In the Cartesian co-ordinate system, a point is represented by its distance from a fixed reference point in the plane. The pen tablet follows this system while recording a particular signature. However, some pre-processing needs to be done before adding these components to the Feature Vector.

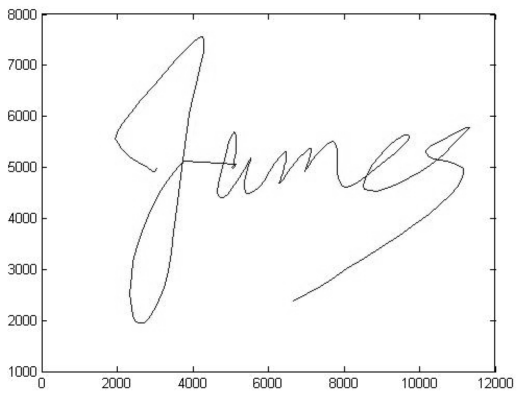

Figure 4: Before shifting the origin

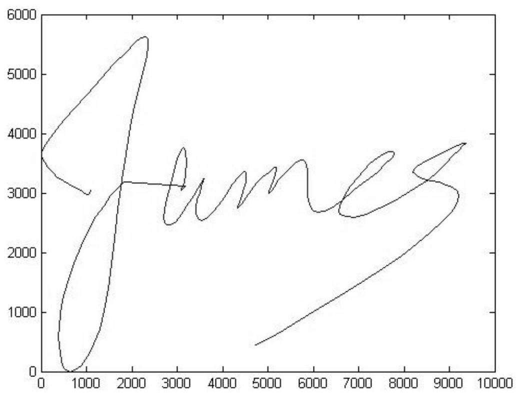

Figure 5: After shifting the origin

In order to ensure uniformity between two signature samples of the same user, it is desirable that the values of co-ordinates for all the points of the signature are decided on the basis of their distance from a reference point located on the signature itself. A solution to this problem is to shift the origin to the co-ordinate which has the minimum x and y co-ordinate values. This can be done by subtracting the minimum X value and the minimum Y values from the X and the Y columns shown in Figures 4 and 5.

Using these processed values the average and the standard deviation of the values of the (x,y) co-ordinate are calculated.

5.1.2 Polar co-ordinate system

A point (x,y) in the Cartesian Co-ordinate system can be easily converted to polar co-ordinate system to (r, θ).

To ensure high inter user dissimilarity for values of standard deviation and average for r and θ, the position of the pole should be chosen such that variation of r and θ will be maximum for all the points of the signature.

The maximum variation in θ can be obtained by choosing the origin along line y=ymax/2. Hence, the x co-ordinate on this line should be chosen such that the maximum variation of r is obtained. Figure 6 shows that r can vary from 0 to $\sqrt{(x_{max}/2)^2 + (y_{max}/2)^2}$ if pole is chosen at the centre whereas r varies from 0 to $\sqrt{(x_{max}/2)^2 + y_{max}{}^2}$ if the position of the pole is chosen along the centre-top or centre-bottom as shown in Figure 7. Keeping this aspect in mind, centre-bottom (0,ymax/2) has been chosen as the pole co-ordinate.

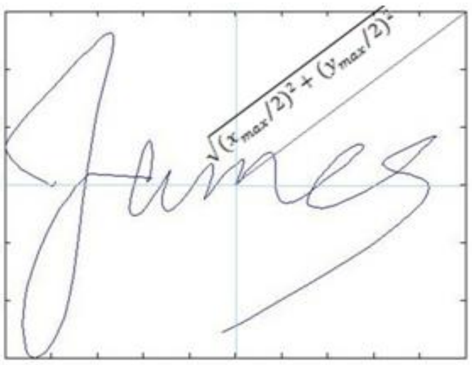

Figure 6: The value of r when pole is chosen as the exact centre

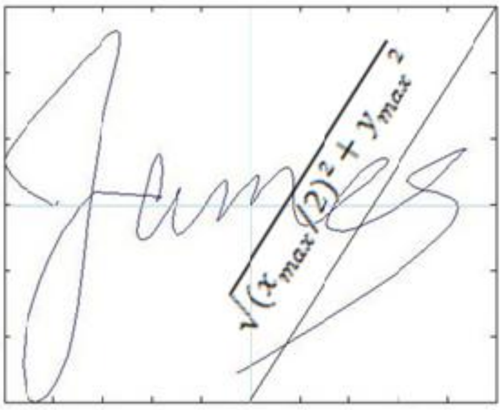

Figure 7: The value of r when the pole is chosen at the bottom centre position

Thus to convert the Cartesian co-ordinate system to Polar form, the following steps must be performed:

i. Subtract xmin value from all the points in the X column and ymin from all the points in the Y column; the reasons for this step have been mentioned in the Cartesian coordinate system section.

ii. Next, subtract all the values in the Y column of the dataset by ymax/2. This makes the Cartesian origin shift to (0, ymax/2).

iii. After step 2, calculate the value of r and θ for each point using the formulae given below.

$$x = r\cos\theta \tag{5.1}$$

$$y = r\sin\theta \tag{5.2}$$

5.2 Online Features

These features are more difficult to replicate compared to the offline characteristics, which are based on the shape of the signature. So, even if two signatures are nearly the same in appearance, online features can still distinguish between an original signature and a forgery.

In this project, the following online features have been included:

i. Velocity

ii. Acceleration

iii. Angular Velocity

iv. Pressure

v. Sections

The method of extraction of each feature is as given below.

5.2.1 Velocity

The velocity of the pen-tip while the user is signing is not easy to replicate. Hence, this is an important feature to distinguish between an original signature and a forgery.

The formula for velocity is given as follows:

$$v = \frac{\sqrt{(x1-x2)^2+(y1-y2)^2}}{t2-t1} \tag{5.3}$$

In general, it is observed in the database that total time taken by the impostor to forge the signature is more than the total time taken by the legitimate user to sign. However, this information is not considered during the calculation of velocity. To highlight the difference between an original signature and a forgery, it was decided to replace the method of calculation of time interval between two sample points, by replacing it with relative time interval as follows:

$$\text{relative time difference} = \frac{\text{time difference between two sample points}}{\text{total time taken to complete the signature}} \tag{5.4}$$

$$r(t) = \frac{t_2-t_1}{t_{end}-t_{start}} \tag{5.5}$$

Here, the numerator represents the time difference between two sampling points and the denominator represents the total time taken for the user to complete the signature.

By using new relative time interval, the formula for velocity becomes:

$$v = \frac{\sqrt{(x1-x2)^2+(y1-y2)^2}}{t_2-t_1} \times (t_{end} - t_{start}) \tag{5.6}$$

The scalar part of velocity i.e. speed is a very important feature for distinguishing between an original signature sample and a forgery. This is because at sharp curves of the signature, even a skilled forger might slow down whereas the legitimate user will have nearly constant velocity for all signature samples. Thus, inclusion of this dynamic feature offers better chances for rejecting a forgery, which is not possible by the offline characteristics. Besides, this feature also plays an important role for identification purposes as the rate at which a person signs and the total time a person takes to complete the signatures varies from user to user.

5.2.2 Acceleration

Like velocity, this dynamic feature also takes advantage of the low confidence of the imposter while making the forgery and therefore, improves the chances of rejecting a forgery. Also, like velocity, this quantity also plays an important role for identification purposes as it varies from user to user.

Also, the points made in pre-processing of velocity are applicable even for calculating acceleration. Hence, the formula for relative time interval $r(t)$ was used, instead of using actual time interval.

Thus, the formula for acceleration becomes:

$$a = \frac{|v_2 - v_1|}{t_2 - t_1} \times (t_{end} - t_{start}) \qquad (5.7)$$

5.2.3 Angular Velocity

The arguments made in favor of including the relative time interval for linear velocity and linear acceleration are applicable even to angular velocity; hence, the formula for angular velocity is:

$$\omega = \frac{|\theta_2 - \theta_1|}{t_2 - t_1} \times (t_{end} - t_{start}) \qquad (5.8)$$

5.2.4 Pressure

The pressure applied at the pen-tip cannot be easily emulated by the forger and, hence, is a very important attribute for distinguishing between an original signature and a forgery. This has been recorded by the tip of the electronic pen, and can be directly used as it is from the data set of the signature sample. Statistical operations like standard deviation and average are used to summarize this information and then included in the Feature Vector for training the neural network.

5.2.5 Sections

The pen status indicates whether the pen is up or down. The value 1 indicates that the pen is up and the value 0 indicates it is down. The number of times a person lifts the pen while signing is exactly same for any two signature samples of the same user. This is because every person has a particular style of writing. Some people lift their pen after writing every alphabet while people who use the cursive style of writing do not lift their pen as many times as the latter. Also, while emulating a steep curve on the signature, a forger might lift the pen to get the shape right, whereas the legitimate user will not do so. Thus, this information is vital for detecting forgeries. To calculate this quantity, the number of transitions of the pen status value from 0 to 1 is counted, which indicates the pen has been lifted during signing.

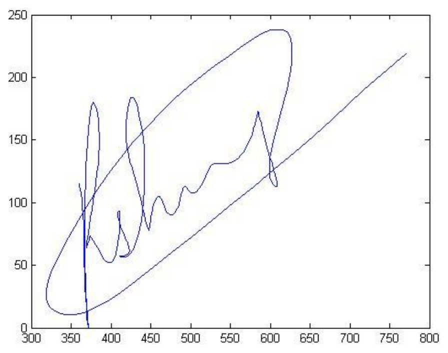

Figure 8: Signature Sample

Consider the signature image in Figure 8 as a sample image for the purpose of this discussion. Table 3 is the tabular representation of the signature in the form of a sequence of tokens. The feature vector extracted for the sample is as given below.

Feature Vector = [0.3191 0.5469 0.0003 0.2125

0.2197 8.40e-05 7.531 567.12

1083 120.252 136.91 43.865

178.74 179.55 60.656 1]

Table 3. Tabular Representation of Signature Sample

X	Y	Timestamp	Pen Status	Pressure
359	490	18984612	1	0.2453
359	489	1898420	1	0.2316
359	489	18984627	1	0.2258
360	490	18984665	1	0.2326
⋮	⋮	⋮	⋮	⋮
768	387	18984671	1	0.4877
765	389	18984679	1	0.3264

6. IMPLEMENTATION OF PROPOSED SYSTEM

The proposed system has two variants of Neural Networks:

i. Identification Network

ii. Individual Verification Network

Both the variants of networks have been trained using the Error Back-Propagation Training Algorithm (EBPTA). The following sections explain the creation and training of the networks and the overall structure and functioning of the proposed system.

6.1 Creation and Training of EBPTA Neural Networks

The two types of Neural Networks correspond to two steps in the classification system. The first step is Identification step and the second step is Verification. The methodology of creating the neural network is the same for both. The features extracted from all the signatures are used to create a Neural Network which is trained using Error Back Propagation Training Algorithm (EBPTA). The number of input nodes, the number of neurons in the hidden layer, the number of neurons in the output layer, the goal, the number of epochs, the learning rate and the momentum constant and other such parameters of the neural network are decided and the neural network is created based on this information using Neural Network Toolbox of MATLAB.

The entire database is divided into two parts – the training set and the test set. In the training phase, 75% of signature samples (30 signatures) of a particular user are used and the remaining 25% (10 signatures) are set aside for testing. There are equal number of original and forged signature samples in both training and testing sets. Since EBPTA is a supervised algorithm, during training phase even the desired output is provided along with the Feature Vector. The network is trained until the output it generates is as close as possible to the desired output. However, the attributes of training samples and the desired output is different for the networks as it is influenced by the role of that network.

6.2 Overall Structure and Functioning of Proposed System

The role played by each of the networks in the authentication process determines the architecture and the type of training samples used to train that network. This is discussed as follows.

The Identification Neural Network has the job of mapping the test signature to its respective target user from among a set of users. Thus, the input to this network is the Feature Vector of the Signature and the output is the user-id. From this it can be concluded that the input layer of the neural network has nodes equal to the number of elements in the Feature Vector. Also, the number of nodes in the output layer is equal to the number of users in a given Identification set. At a time only one node gets activated in the output layer, and the vertical position of the node that is activated is equal to the user-id. If the signature is mapped to the expected user-id, it passes the first step and proceeds to the verification step. If it fails this Identification step it is declared as a forgery and the authentication process stops here.

The Individual Verification Neural Network is responsible to verify whether the signature is original or forged. Thus, the number of input nodes is equal to the number of elements in the Feature Vector. The number of output nodes however is only two, corresponding to the two cases of ‘Original’ or ‘Forgery’. Also, it would be easier to train separate networks for separate individuals as the classification problem would break down in smaller modules. Hence, separate networks are trained for every individual.

Now, looking at the training samples, the Identification Neural Network requires only the Original set of signature samples to be trained on. It is not concerned with the classification of a Forgery. Random Forgeries would automatically be eliminated by this network due to the nature of training it has undergone.

The Individual Verification Network, on the other hand, requires that it should be able to distinguish between originals and skilled forgeries that pass the Identification Neural Network. Hence, it is required that this network is trained on both Original and Forged samples.

The approach taken in the proposed system, to split the problem of Forgery Prevention in two steps has many advantages as follows:

i. Breaking the classification problems into simpler problems results in lower complexity of the Neural Networks, which in turn, results in faster training of the entire system.

ii. This system has intrinsic ability to handle both skilled and random forgeries. The Identification Module would automatically eliminate random forgeries by classifying them under the wrong user-id. Further, the Individual Verification module can be trained using comparatively larger number of features corresponding to a given user in a reasonable amount of time.

7. RESULTS

The results obtained are shown in this section.

7.1 Testing Results for Acquired Database

The results for the Individual Verification networks and Identification networks generated for the acquired database are discussed in sections 7.1.1 and 7.1.2.

7.1.1 Individual Verification Networks

Table 4 shows the performance of 5 sets of 40 networks having different number of neurons in the hidden layer with learning rate = 0.5 and momentum constant = 0.5. Each of these networks attained a performance of 1e-5. The FAR and FRR of each of these networks observed during the recall phase was 0.

Table 4. Individual Verification Networks of Acquired Database with lr = 0.5 and mc = 0.5

LR = 0.5 MC = 0.5 Goal = 1e-5 Recall FAR = 0 Recall FRR = 0			
Hidden Neurons	Correctly Classified out of 40	Overall FAR	Overall FRR
15	25	17	1.5
11	18	20	2.5
8	19	20	2.5
7	20	18	2.5
4	18	18	3.5

Table 5 shows the performance of 5 sets of 40 networks having different values for learning rate and momentum constant with number of neurons in the hidden layer = 15. Each of these networks attained a performance of 1e-5. The FAR and FRR of each of these networks observed during the recall phase was 0.

Table 5. Individual Verification Networks of Acquired Database with hidden neurons = 15

Hidden Neurons = 15 Goal = 1e-5 Recall FAR = 0 Recall FRR = 0				
LR	MC	Correctly Classified out of 40	Overall FAR	Overall FRR
0.9	0.1	17	21	3.0
0.8	0.1	16	23	2.0
0.7	0.2	18	21	3.5
0.6	0.2	17	19	4.0
0.5	0.5	25	17	1.5

7.1.2 *Identification Networks*

Table 6 shows the performance of 5 networks having different number of neurons in the hidden layer with learning rate = 0.8 and momentum constant = 0. Each of these networks was trained for 100000 epochs.

Table 6. Identification Networks of Acquired Database with lr = 0.8 mc = 0

Learning Rate = 0.8 Momentum = 0 Epochs = 100000			
Hidden Neurons	Performance	Recall Error	Test Error
35	0.0331	32.6667	34.5
30	0.0457	45	46.0
20	0.0260	25	27.0
15	0.0058	5.1667	9.0
12	0.0197	18	23.0

Table 7 shows the performance of 5 networks having different values for learning rate and momentum constant with number of neurons in the hidden layer = 15. Each of these networks was trained for 100000 epochs.

Table 7. Identification Networks of Acquired Database with hidden neurons = 15

Hidden Neurons = 15 Epochs = 100000				
Learning Rate	Momentum	Performance	Recall Error	Test Error
1	0	0.0137	13	19.5
0.8	0	0.0058	5.1667	9.0
0.7	0.1	0.0245	21.3333	22.5
0.6	0.1	0.0175	13.5	20
0.5	0.2	0.0146	10.8333	21.5

7.2 Testing Results for Created Database

In this section, the test results obtained for the created database are shown.

7.2.1 *Individual Verification Networks*

Table 8 shows the performance of 5 sets of 40 networks having different number of neurons in the hidden layer with learning rate = 0.9 and momentum constant = 0.1. Each of these networks attained a performance of 1e-5. The FAR and FRR of each of these networks observed during the recall phase was 0.

Table 8. Individual Verification Networks of Created Database with lr = 0.9 and mc = 0.1

Learning Rate = 0.9 Momentum = 0.1 Goal = 1e-5 Recall FAR = 0 Recall FRR = 0			
Hidden Neurons	Correctly Classified out of 20	Overall FAR	Overall FRR
15	16	2.5	0
13	16	2.5	0.5
11	16	1.5	0.5
10	16	2.5	0
9	15	2	0.5

Table 9 shows the performance of 5 sets of 40 networks having different values for learning rate and momentum constant with number of neurons in the hidden layer = 15. Each of these networks attained a performance of 1e-5. The FAR and FRR of each of these networks observed during the recall phase was 0.

Table 9. Individual Verification Networks of Created Database with hidden neurons = 15

Hidden Neurons = 15 Goal = 1e-5 Recall FAR = 0 Recall FRR = 0				
LR	MC	Correctly Classified out of 20	Overall FAR	Overall FRR
1	0	16	3	0
0.9	0.1	16	2.5	0
0.7	0.3	17	3.5	0
0.5	0.5	15	3	1
0.3	0.7	14	4	0

7.2.2 *Identification Networks*

Table 10 shows the performance of 5 networks having different number of neurons in the hidden layer with learning rate = 0.8 and momentum constant = 0.2.

Table 10. Identification Networks of Created Database with lr = 0.8 and mc = 0.2

Learning Rate = 0.8 Momentum = 0.2				
Hidden Neurons	Epochs	Performance	Recall Error	Test Error

19	108267	5.208e-5	0	2
18	100295	1.072e-2	5	5
17	100138	2.005e-2	10	14
16	116902	1.005e-2	5	6
15	100062	6.934e-5	0	1

Table 11 shows the performance of 5 networks having different values for learning rate and momentum constant with number of neurons in the hidden layer = 19.

Table 11. Identification Networks of Created Database with hidden neurons = 19

Hidden Neurons = 19					
LR	MC	Epochs	Performance	Recall Error	Test Error
1	0	130462	3.664e-5	0	3
0.9	0.1	116354	9.376e-3	4.6667	4
0.8	0.2	108267	5.208e-5	0	2
0.7	0.3	100119	2.005e-2	10	11
0.6	0.4	71874	2.011e-2	10	10

8. CONCLUSIONS

The Online Signature Forgery Prevention System presented in this paper tackles the problem of Signature Verification by dividing it into two parts; Identification of the Individual followed by Verification of signature to differentiate between original signature and forgery.

In the system presented in the paper, a signature was classified as original or forged in two steps. The first step was the identification step. A neural network was trained to identify the user number of the test signature. Hence, this network was trained only on original signatures and its output was the user number which the network has classified as the signer. If the identified user number was same as the user number of the claimant, only then the signature was forwarded to the individual network, otherwise the signature was declared as a forgery in the first step itself. If the signature passed the identification test, then it was sent to the Individual network of the user, which was the same as the one used in the Individual system.

The first module of the system i.e. Identification network provided protection against random forgeries whereas the second module i.e. Individual Verification networks provided protection against skilled forgeries. Thus, the proposed system is well-equipped to deal with both kinds of forgeries.

9. REFERENCES

[1] Jain, A.K.; Hong, L; Pankanti, S., Communications of the ACM, Biometric Identification, Vol. 43, No. 2, pp. 91-98, (2008)

[2] Miller, B., IEEE Spectrum, Vital signs of identity, Vol. 31, No. 2, pp. 22–30, (1994)

[3] Dong, L.; Yun-Jian, G.; Xue-Yong, Z., 2010 International Conference on Computer Application and System Modeling (ICCASM), On-line signature verification based on template matching approach and support vector data description, Vol. 12, pp. 681-685, (2010)

[4] Han, C.C.; Chang, P.C.; Hsu, C.C.; Jeng, B.S., IEEE Proceedings - 33rd Annual International Carnahan Conference on Security Technology, An on-line signature verification system using multi-template matching approaches, pp. 477-480, (1999)

[5] Daramola, S.A.; Ibiyemi T.S., International Journal of Engineering & Technology IJET-IJENS, Efficient on-line signature verification system, Vol. 10, No. 4, pp. 48-52, (2010)

[6] Boser, B.E.; Guyon, I.M.; Vapnik, V.N., Proceedings of the Fifth Annual Workshop on Computational Learning Theory, A training algorithm for optimal margin classifiers, pp. 144-152, (1992)

[7] Kour, J.; Hanmandlu, M.; Ansari, A.Q., 2011 International Conference on Image Information Processing (ICIIP), Online signature verification using GA-SVM, pp. 1-4, (2011)

[8] Fauziyah, S.; Azlina, O.; Mardiana, B.; Zahariah, A.M.; Haroon, H., 6th International Symposium on Mechatronics and its Applications (ISMA), Signature verification system using Support Vector Machine, pp. 1-4, (2009)

[9] Justino E.R.; Yocoubi A.E.; Bortolozi F.; Sabourin R., 4th IAPR International on Document Analysis Systems, An Off-line Signature Verification System Using HMM and Graphometric features, (2000)

[10] http://www.cse.ust.hk/svc2004/download.html