Jurnal Mekanikal
December 2005, No. 20, 22 - 41

INTELLIGENT BIOMETRIC SIGNATURE VERIFICATION SYSTEM INCORPORATING NEURAL NETWORK

Lim Boon Han
Musa Mailah

Department of Applied Mechanics
Faculty of Mechanical Engineering
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
Email: bhlim80@yahoo.com, musa@fkm.utm.my

ABSTRACT

In this paper, the practical development of a novel but yet simple, low cost and user-friendly PC-based biometric technique to verify human signature is presented. The system is based on data acquired from time duration, pen position and pressure signals of handwritten signatures. A preprocessing stage of the acquired signature signals involving the use of a number of common feature extraction techniques is rigorously highlighted. An intelligent feature of the system is made possible through the application of a multilayer feedforward neural network that is used together with suitable algorithms to complement the verification process. The results of the study showed that the system is effective and promising in identifying correct human signatures presented to the system.

Keywords *: Biometrics, neural network, back error propagation, signature verification*

1.0 INTRODUCTION

There are a number of methods in which a person can gain access to a system that demands specific authorization or permission. Amongst them are by way of using password, key, token, card, personal-identification-number (PIN) and biometric method. The first five methods may often result in undesirable situation such as the vital element/s (of the methods) being forgotten, lost or stolen, but not for the biometric method. Biometric approaches can be classified as iris scan, retinal scan, fingerprint, facial recognition, signature verification, voice recognition and hand/finger geometry. These involve trait or characteristic of a person that is essentially distinctive, will not (or hardly) change with time and more often unique to each and every individual person [1]. It is thus obvious that the biometric concept is suitably applied to prospective systems that demand the presence of excellent security and safety measures to be in place. The entry or access to these 'sensitive premises' is only permitted to specific individuals who have thoroughly passed the identification cum verification procedure. The subject of interest in this study is the signature verification procedure that deals with the process of verifying handwritten signature patterns of human individuals.

of interest in this study is the signature verification procedure that deals with the process of verifying handwritten signature patterns of human individuals.

Basically, there are two main categories of signature verification system, namely the on-line and off-line signature verification methods. Off-line signature verification is normally an image processing technique with the application of a suitable verification algorithm. This is essentially a method whereby the user is supposed to have completely written down the signature onto a template that was later captured by a CCD camera [2] or scanner [3] to be processed. This signature was later compared to those already stored in the database for verification prior to making any decision. On the other hand, on-line signature verification comprises not only the static image, but also dynamic features like capturing of the total signing duration, pressure along signature path, acceleration, and speed measurement. There are two approaches used for on-line signature verification; one involves function calculation and the other is based on parameter calculation [4]. The former procedure calculates the XY coordinate (position), pressure, speed and acceleration of the signature and transforms it into a function. This is rather time consuming but produces higher accuracy while the latter method results in less accuracy but saves time by extracting the signature properties locally and globally [1, 4].

One of the important aspects of signature verification is the data acquisition process in which essential information regarding the parameters of the handwritten signature is obtained. Data acquisition is normally performed through an input device in the form of a tablet or digitizer. Some researchers use digitized tablet with pressure sensitive surface [5-7] while others utilise a special pen equipped with pressure sensor [8] and signal conditioning element that can effectively extract the pressure distribution characteristic and XY coordinate of the on-line signature. The data was acquired through a computerized data acquisition procedure to be processed. The test signature was subsequently transformed into relevant data and later compared to those in the database for verification and matching operation. A number of algorithms can be used to preprocess the data retrieved on-line including normalization [9-11], linear prediction coding [12], dynamic time warping [11, 13-15], tree matching [11], smoothing of data [5, 9], noise reduction [11], segmentation [9] and combination among them. For verification phase, some researchers use neural network [2, 7, 12, 16], autoregressive [17], statistical model [9, 18, 19], hidden Markov model [20], string matching [5] and a combination of them [16, 21, 22]. Each of these techniques has its advantages and disadvantages. For example, a statistical method has yielded a fast calculation but less accuracy and hence appropriate for point of sale [19]. Meanwhile, neural network can produce more accurate results at the expense of additional time consumed in training the signature data. Some researchers developed data parallelism and algebraic parallelism strategies to speed up the computation time [23].

The paper is structured as follows - the first part relates the methodology employed in the study followed by a description of the design of the graphic user interface (GUI) and data acquisition technique. The preprocessing of the signatures involving a number of selected algorithms is subsequently discussed. Next, the application of the neural network method is presented and ultimately, the verification procedure and its performance are presented in the later part of the paper.

2.0 METHODOLOGY

A biometric system that integrates both hardware and software elements are proposed in the study. A digitized tablet with pressure sensitivity surface together with a special pen is used to capture the pressure signal and XY coordinate. This device sends data in packet through the USB port with the support of Wintab 32-bit interface API and related library. The Wintab32 API and its reference commands can be found in [24].

The software elements of the system comprises Microsoft Visual Basic, MATLAB v6.0, Microsoft Jet Engine, Microsoft Access and associated components. Microsoft Visual Basic is used as graphic user interface (GUI) and programming for communication between tablet and computer. MATLAB and its relevant components are implemented to perform training, learning and verifying the signatures. Microsoft Access has been utilized for the design of signature database while Microsoft Jet Engine provides the mechanism to read and write to the database (in the form of *mdb* files) created from Microsoft Access.

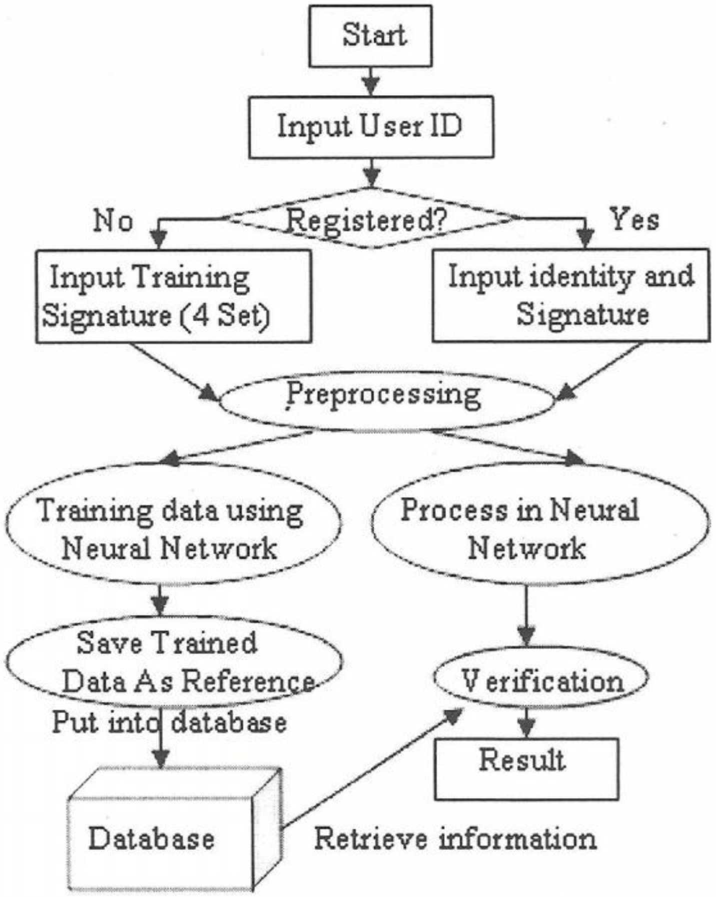

Start
Input User ID
Registered?
No
Yes
Input Training Signature (4 Set)
Input identity and Signature
Preprocessing
Training data using Neural Network
Process in Neural Network
Save Trained Data As Reference
Put into database
Verification
Result
Database
Retrieve information

Figure 1: A flow chart of the signature verification process

The system is designed specifically for the verification of 'non-character' signature. Thus, 'letter by letter' signature like the Chinese, Japanese and English characters will not be recognized by the proposed system since specialized algorithms need to be used for the verification of such signatures [2, 25]. The different number of stroke for the same signer in different signature captured will be considered as forger for the algorithms used in this system [26].

There are five stages to be considered in presenting the whole process of signature verification. First of all, the user is required to register four sets of signatures for storage and training of the neural network. In case the user has already registered, he/she is prompted to select his/her identity as contained in the database. The next step will be the preprocessing of the signatures received (as discussed in section 4.0 of this paper). The third stage is to carry out features extraction procedure from the given signature information which is later used for training using neural network (for reference). The fourth stage deals with the process of comparing the signatures submitted with reference signatures stored in database for verification using neural network method. The final step is the storage of the processed signature in the database and the performance evaluation to verify the system's accuracy. The whole process is summarized in Figure 1.

3.0 DATA ACQUISITION AND GRAPHIC USER INTERFACE (GUI)

3.1 Data Acquisition

The essential information of a handwritten signature that can describe traits of a person needs to be acquired first before it is preprocessed. For this purpose, a hardware device in the form of a pen and tablet device is used in the study. This is shown in Figure 2. The hardware system contains a pressure sensitive element (at the surface of the tablet) that can sense the physical movement of the signature operation when the user writes his/her signature using the pen on the tablet. The signals of the signature are digitized through a signal conditional element that is already built-into the system. Later, this information will undergo or pass through a number of processes for future utilisation.

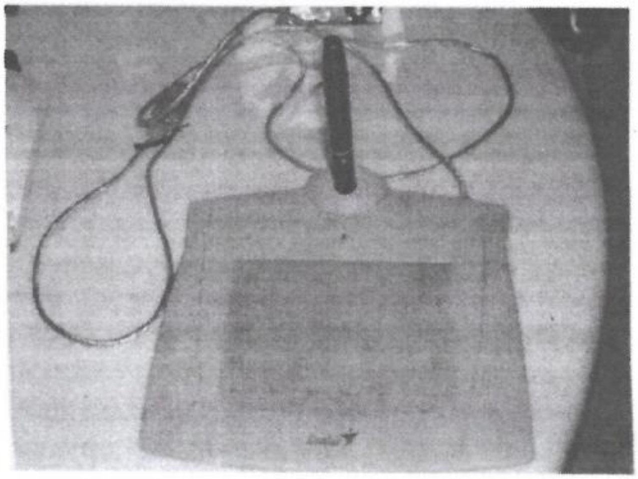
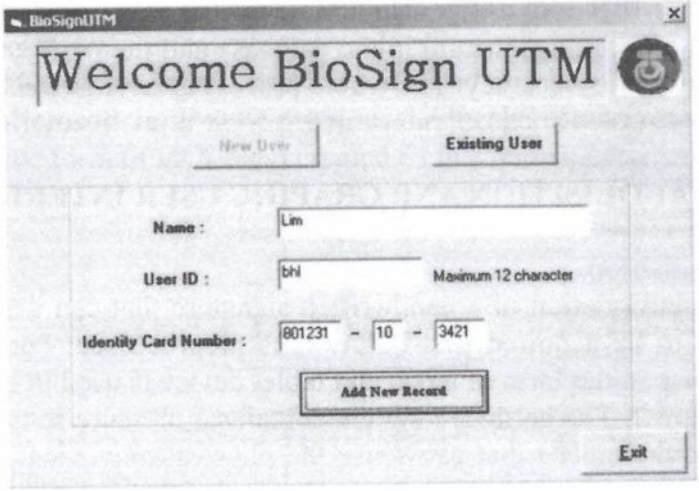


Figure 2: Input device comprising a pen and a digitizing tablet

3.2 GUI

The GUI is designed and constructed in Visual Basic environment to create user-friendly and better management features. Figure 3 displays the main *welcome* GUI window for the user to input his/her identity information through the input pen and tablet device. New user is prompted to input this information prior to signing four sets of signatures that will be stored in the database for reference and neural network training. The actual written signature operation shall be executed using the input hardware in which the signature will be displayed in the 'sign box' as shown in another GUI *signing* window of Figure 4. Meanwhile, existing or registered user needs to inform the system his/her identity before proceeding to sign only a set of signature for the purpose of verification. Figure 5 illustrates the database designed in Microsoft Access to store user information, signatures and status of verification.

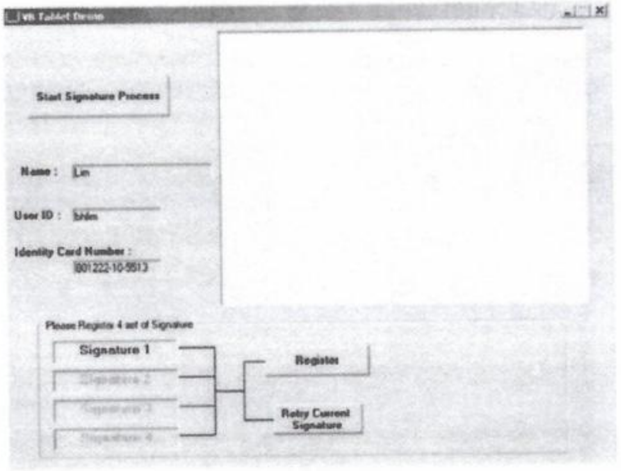


Name : Lim

User ID : bhl Maximum 12 character

Identity Card Number : 801231 - 10 - 3421

Figure 3: The *welcome* window

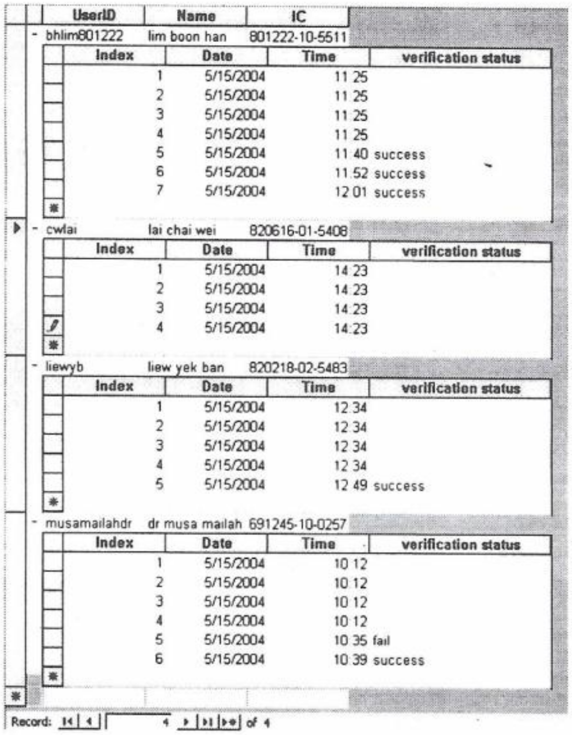


Name : Lin

User ID : bhlm

Identity Card Number : 801222-10-5513

Figure 4: The *signing* window



UserID	Name	IC
bhlim801222	lim boon han	801222-10-5511

Index	Date	Time	verification status
1	5/15/2004	11:25	
2	5/15/2004	11:25	
3	5/15/2004	11:25	
4	5/15/2004	11:25	
5	5/15/2004	11:40	success
6	5/15/2004	11:52	success
7	5/15/2004	12:01	success

UserID	Name	IC
cwlai	lai chai wei	820616-01-5408

Index	Date	Time	verification status
1	5/15/2004	14:23	
2	5/15/2004	14:23	
3	5/15/2004	14:23	
4	5/15/2004	14:23	

UserID	Name	IC
liewyb	liew yek ban	820218-02-5483

Index	Date	Time	verification status
1	5/15/2004	12:34	
2	5/15/2004	12:34	
3	5/15/2004	12:34	
4	5/15/2004	12:34	
5	5/15/2004	12:49	success

UserID	Name	IC
musamailahdr	dr musa mailah	691245-10-0257

Index	Date	Time	verification status
1	5/15/2004	10:12	
2	5/15/2004	10:12	
3	5/15/2004	10:12	
4	5/15/2004	10:12	
5	5/15/2004	10:35	fail
6	5/15/2004	10:39	success

Figure 5: Database design

4.0 PREPROCESSING OF SIGNATURES

After the signatures from the new user have been collected, the information must be preprocessed. This involves a number of procedures including reduction of noise, rectification of error during the data acquisition process, elastic warping against each other, normalization and smoothing of data. The preprocessed signals need to undergo adequate and rigorous training procedure using neural network technique and later stored as useful database for the verification operation at the later stage. All the algorithms are programmed and executed using MATLAB.

For security reason, there are two main factors that determine the acceptance of the data for a new user to register successfully. They are the signature time duration and degree of similarities amongst the four sets of signatures obtained. If the time taken for the user to write down each of his signature varies or deviates 15% from each other, he/she is prompt to sign again. This can be mathematically expressed as,

$$\tau_{\text{current}} \times 0.15 \geq \tau_{\text{mean}} \tag{1}$$

If all signatures follow this rule, then the next step is to normalize the signatures. Three signatures (2, 3 and 4) will be rotated, scaled and transformed against the first signature. The transformation procedure is given as,

$$X^{i}_{\text{transform}} = X^{i} + \delta x^{i}_{\text{center}}\ ;\ X^{1}_{\text{transform}} = X^{1} \tag{2}$$

$$Y^i_{\text{transform}} = Y^i + \delta y^i_{\text{center}}\ ;\ Y^1_{\text{transform}} = Y^1 \qquad (3)$$

X-*Y* coordinates (Cartesian coordinate) of the signature pattern set was then converted to polar form before being rotated with reference to the start point and end point angles of ‘first set signature. The resulting transformation was then converted back to the Cartesian coordinate once the rotation process was completed.

$$\theta^i_r = ((\theta^1_s - \theta^i_s) \times 0.5) + ((\theta^1_e - \theta^i_e)\, 0.5) \qquad (4)$$

The next step in normalization is to scale the signatures to fit the first signature in *X* and *Y*-axes. This is done by finding the maximum and minimum values of *X*-*Y* coordinate of the first set signature and later fit other signatures to its size.

$$\Theta X^i = ((\Theta X^1_{\max} - \Theta X^1_{\min}) / (\Theta X^i_{\max} - \Theta X^i_{\min}))\, X^i \qquad (5)$$

$$\Theta Y^i = ((\Theta Y^1_{\max} - \Theta Y^1_{\min}) / (\Theta Y^i_{\max} - \Theta Y^i_{\min}))\, Y^i \qquad (6)$$

Even though the *X*-*Y* dimensions of all signatures have been normalized, the third dimension (parameter) related to time is not applicable when neural network algorithm is used. Each signature has different time duration and this variation is not linear, hence *Dynamic Time Warping* (DTW) algorithm is implemented to obtain a point-to-point correspondent between these four sets of signatures. The utilization of DTW to warp *X*-*Y* coordinates of the signature is not appropriate and will distort the original shape of signature [9], hence only the pressure signal is warped. The algorithm is introduced as in Equation (7). The algorithm is modified from the one used in [16, 27]. First of all, there are four sets of signatures related to pressure signal that are in turn represented by four different number of points. Interpolation is applied to make sure all signatures have the same number of points, *n* as written in the following equation:

$$P_i = j_i(n) \quad n = 1,2,3 \ldots N \qquad (7)$$

Next, the end and start point constraints are expressed as in Equations (8a) and (8b), the local continuity constraint in Equation (9) and the global path constraint in Equations(10a) and (10b).

Start point constraint: $P_i(1) = 1$ (8a)

End point constraint: $P_i(N) = j_i(N)$ (8b)

Local continuity constraint: $P_i(n+1) \geq P_i(n)$ (9)

Global path constraints:

$$1 + \frac{(i(k) - 1)}{E_{\mathrm{MAX}}} \leqslant j(k) \leqslant 1 + E_{\mathrm{MAX}}(i(k) - 1) \qquad (10a)$$

$$M + E_{\mathrm{MAX}}(i(k) - N) \leqslant j(k) \leqslant M + \frac{(i(k) - N)}{E_{\mathrm{MAX}}}. \qquad (10b)$$

The constrained area is shown in Figure 6 along with its nomenclature. The local path constraint is depicted as shown in Figure 7 with its related distance measurements described by Equations (11a) to (11f).

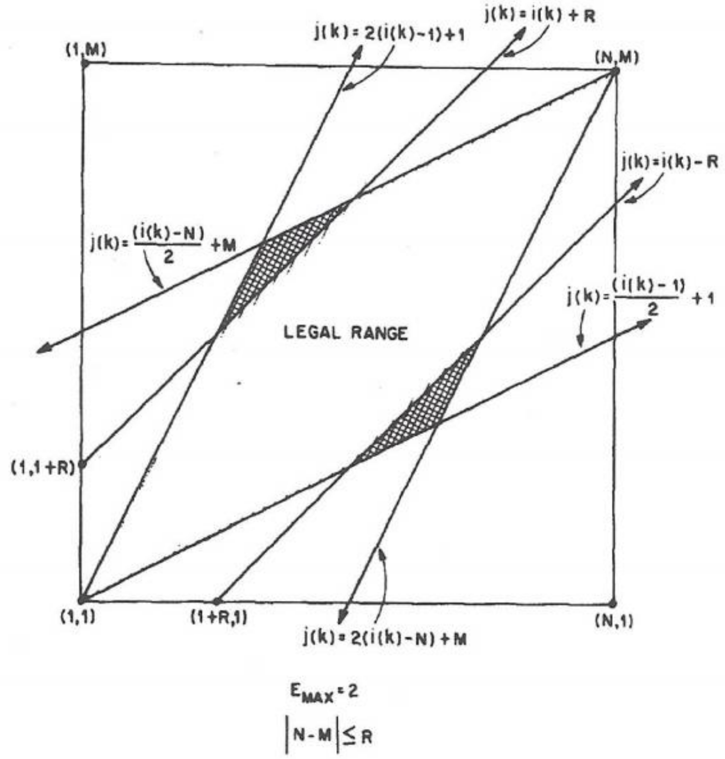


Figure 6: Global path constraint

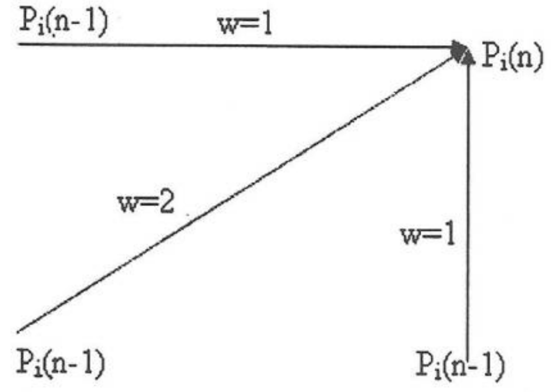


Figure 7: Local path constraint

$$d(P_{ij}) = d(i,j) = (P_i(n) - P_j(n))^2 \tag{11a}$$

$$E(F) = \sum_{n=1}^{N} d(P_{ji}(n)) \times w(n) \tag{11b}$$

$$w(n) = (P_i(n)\text{-}P_i(n\text{-}1)) + (P_j(n)\text{-}P_j(n\text{-}1)) \tag{11c}$$

$$D(P_i, P_j) = \underset{F}{MIN}\left[\frac{\sum_{n=1}^{N} d(P_{ij}(n) \times w(n)}{M}\right] = \frac{1}{M} g_n(P_{ij}(n)) \tag{11d}$$

$$M = \sum_{n=1}^{N} w(n) \tag{11e}$$

$$g_n(P_{ij}(n)) = \underset{P_{ij}(n-1)}{MIN}\left\lfloor g_{n-1}(P_{ij}(n-1) + d(P_{ij}(n) \cdot w(n)\right\rfloor \tag{11f}$$

From Figure 7 and Equation (11f), the final equation in the calculation of DTW is as follows:

$$g(P_i, P_j) = MIN\begin{bmatrix} g(P_i(n), P_j(n-1)) + d(P_i(n), P_j(n) \\ g(P_i(n-1), P_j(n-1)) + 2d(P_i(n), P_j(n) \\ g(P_i(n-1), P_j(n)) + d(P_i(n), P_j(n) \end{bmatrix} \tag{12}$$

After passing through the DTW algorithm, the input data were then smoothed out using cubic smoothing spline algorithm to remove the undesirable spikes and noises during the process of recording the input data via the digitized tablet. The smoothing algorithm is adapted from MATLAB *Curve Fitting Toolbox* and given by,

$$P\sum_i w_i(y_i - s(x_i))^2 = (1-P)\int \frac{d^2s}{dx^2}dx \tag{13}$$

Limitation has also been deliberately introduced to make sure that each of the signals would not vary too much from each other by means of checking their *Euclidean distances*. This helps the neural network to perform faster and better in training by limiting the input layer that contains data variation of less than 3%. This condition can be presented mathematically as follows:

$$dX^{ij} = (\Sigma dX^i(n) - dX^j(n)))/N < \min(dX^{ij}) + (\Sigma(X)/N)\ 0.03 \tag{14}$$

$$dY^{ij} = (\Sigma(dY^i(n) - dY^j(n)))/N < \min(dY^{ij}) + (\Sigma(Y)/N)\ 0.03 \tag{15}$$

$$dP^{ij} = (\Sigma(dP^i(n) - dP^j(n)))/N < \min(dP^{ij}) + (\Sigma(P)/N)\ 0.03 \tag{16}$$

Upon completion of the preprocessing procedure, all signals were plotted. Figures 8 to 11 show the input signals before preprocessing while Figures 12 to 15 reveal the signals after preprocessing.

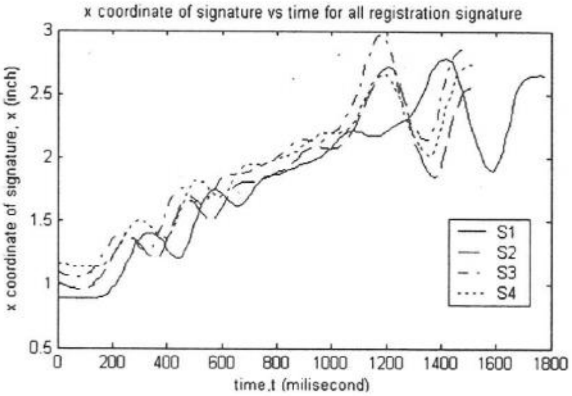


Figure 8: Original *X* coordinates

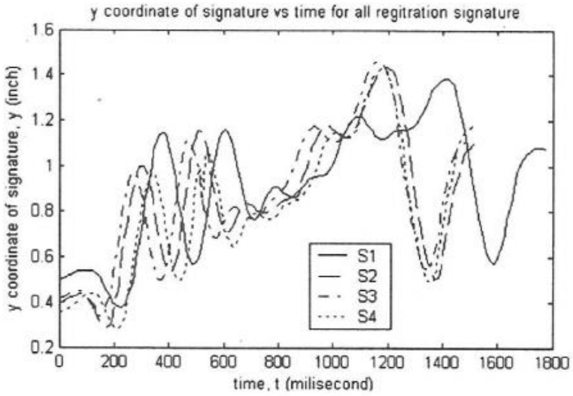


Figure 9: Original *Y* coordinates

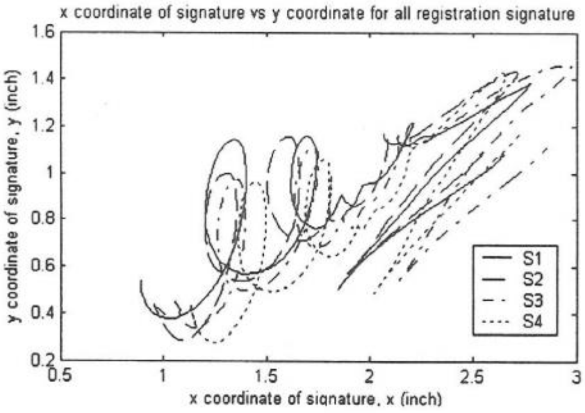


Figure 10: Original shapes of the signature

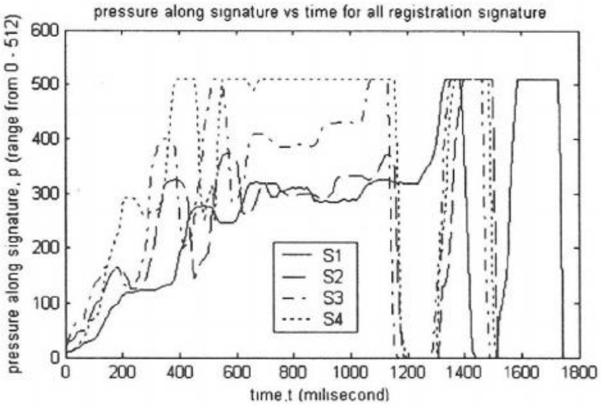


Figure 11: Original pressure distribution patterns

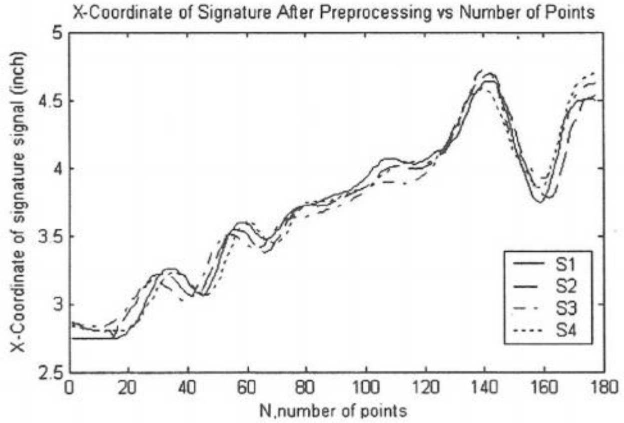


Figure 12: *X* coordinates after normalization

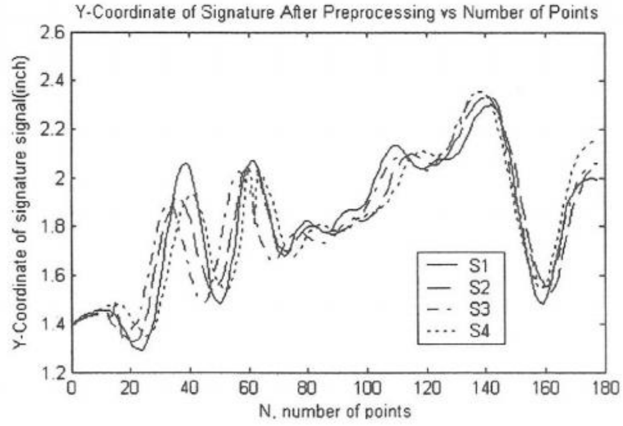


Figure 13: *Y* coordinates after normalization

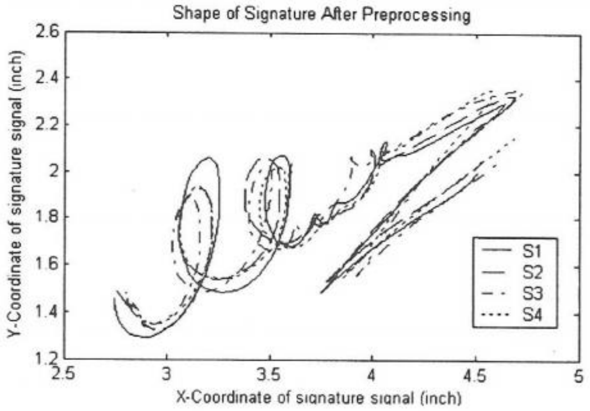


Figure 14: Shapes after normalization

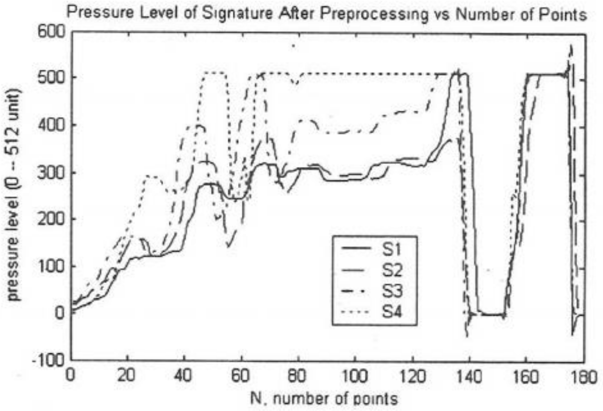


Figure 15: Pressure distribution patterns after DTW

5.0 APPLICATION OF NEURAL NETWORK

The implementation of neural network in the study involves two stages, namely the off-line training process and the on-line verification procedure. For the off-line training process, data are introduced into the neural network system after all the signals have been preprocessed as described in the previous section. At this stage, two neural networks were used in the training process. The first network trained the shapes of the signatures while the second trained the pressure distribution of the signature related to the *X-Y* coordinates.

The structures of both the first and second networks were each constructed of three layers representing the input, hidden and output layers according to the typical multilayer feedforward (MLF) network configuration as shown in Figure 16.

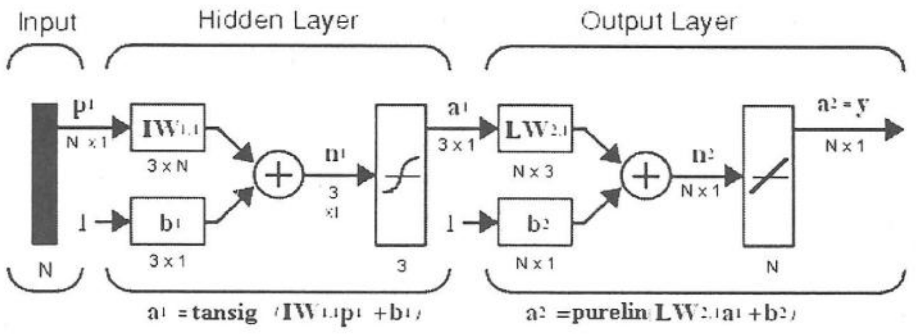


Figure 16: Neural network architecture used in the study

Both networks were trained using the error-backpropagation algorithm that comprises a gradient descent method with momentum and adaptive learning rate parameters. The second layer of both networks incorporates the hyperbolic tangent sigmoid transfer function, while the output/third layer consists of a linear transfer function. The number of neurons for the input layer and output layer of both networks is assigned as *N* that actually defines the number of points in each signal after the signatures signal has been preprocessed. *N* will differ depending on the total signature duration and signature capturing rate. It normally takes a value ranging from 100 to 200.

Meanwhile, three neurons were set in the hidden layer for both networks. It should be noted that the number of neurons and hidden layers for the hidden layer were obtained after a number of trial runs prior to its actual implementation.

In the first network and for the training of the shape signatures, a total of four columns of *X* coordinate signal were assigned to the input layer with the corresponding target of four columns of *Y* coordinate signal at the output. This constitutes the training pair necessary for the off-line training process. The rate of this network to converge to its performance goal relies on *Euclidean distance* calculated, i.e., how far these signals differ from each other. The performance goal is set to 0.005 with momentum constant 0.5 and learning rate 0.02. Figure 17 shows an example of the performance of the networks in training the relevant signal for network 1. The second network is designed to facilitate identification of the pressure distribution of the signatures. A total of eight columns of the *X-Y* coordinates were assigned for the input layer while four columns of the pressure signals were used in the output layer. The performance goal is set to 0.001 with the same value of momentum constant and learning rate as in the first network one. Figure 18 displays an example of the performance in training one of the signals.

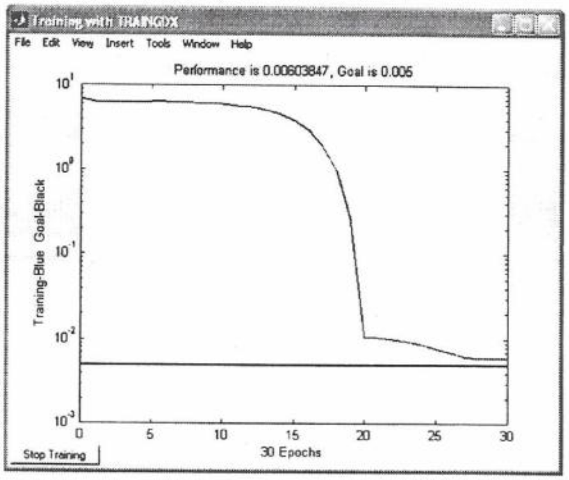


Figure 17: Training of the first network

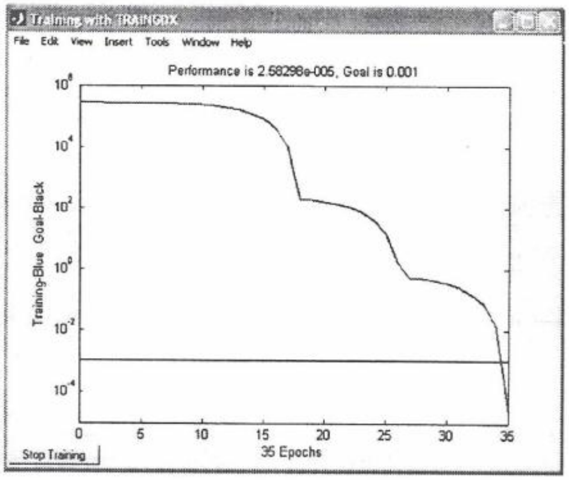


Figure 18: Training of the second network

6.0 VERIFICATION ALGORITHM AND PERFORMANCE EVALUATION

The verification process has to undergo a number of stages. Firstly, the existing user is instructed to input his/her personal information before other additional information (related to the user) and trained reference signatures can be loaded into system. Upon completing the loading of the data, the user is then required to write down his/her signature (on the tablet using a special pen) before verification procedure starts. The data is then captured and sent to MATLAB environment for the verification process. The first step in verification is to check the signature time duration. This is exactly the same procedure to the one described during registration. Once the signature has passed the time duration check, the *X-Y* coordinate and pressure data will be normalized and dynamically warped individually as already explained in the preprocessing section. After preprocessing the signature, the *X* coordinate data will be placed into the first neural network for verification. *Euclidean distance* is later calculated based on the neural network verification result and *Y* coordinate value generated by the user when writing down the signature during registration. A tolerance of about 15% is specified for the verification of shape signature as shown in Figure 19. The equation describing the limit is

$$\Sigma[Y_{ave}(0.15)] < \Sigma(Y_{ave} - Y_{verify}) < \Sigma[Y_{ave}(-0.15)] \qquad (17)$$

After the signature passes the shape verification process, the last step is to verify the pressure distribution pattern of the signature. The *X-Y* coordinates collected for the initial verification process become the input and pressure data as the output of the neural network. A tolerance band is defined so that the test signature pattern does not exceed this limit for the data to be outright accepted. The tolerance band can be clearly seen in Figure 20.

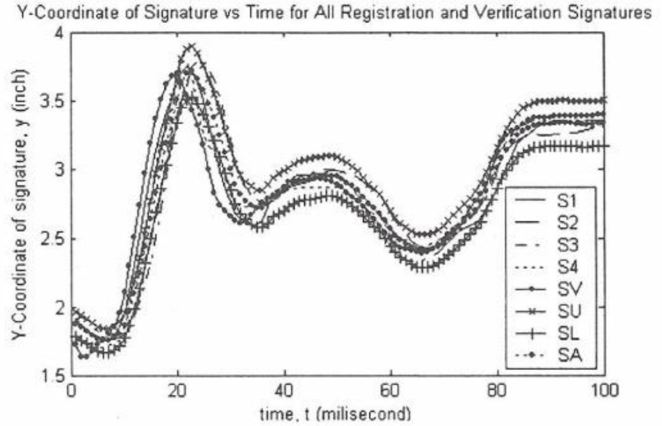


Figure 19: Verification of shape signatures

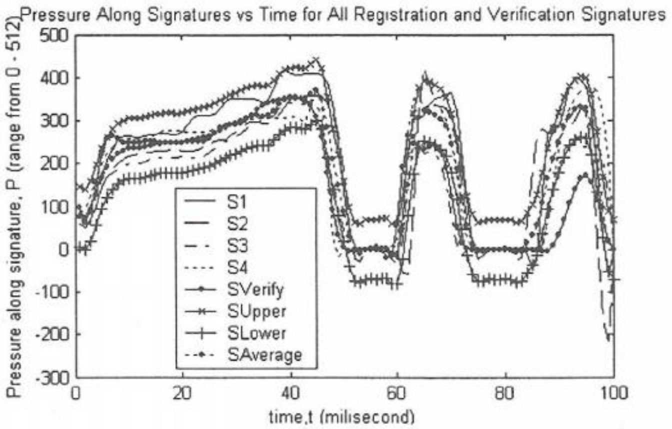


Figure 20: Verification of pressure signals

This system is tested with 30 signatures collected from three different users. Each of them is required to sign 10 times for verification purpose. Random forgery is applied to test the system with 10 forger signatures for each user. A total of 60 signatures have been collected. The test results produce 17% False Rejection

Rate (FRR) which is a Type I error and 0% False Acceptation Rate (FAR) which is a Type II error. The definition of the types of error can be seen in Figure 21 [4]. In simple terms, the bigger the Type I error, the smaller will be the Type II error and vice-versa.

The result shows that the thresholds or tolerances are not finely tuned based on Figure 21. Hence, it does not necessarily imply the system is very accurate since it is not tested using standard (or public) database pertaining to the signature records. Further testing needs to be done in order to render the system more reliable. Besides, only random forgery is applied without taking into account the skilled forgery. This will in turn artificially reduce the Type I and Type II errors.

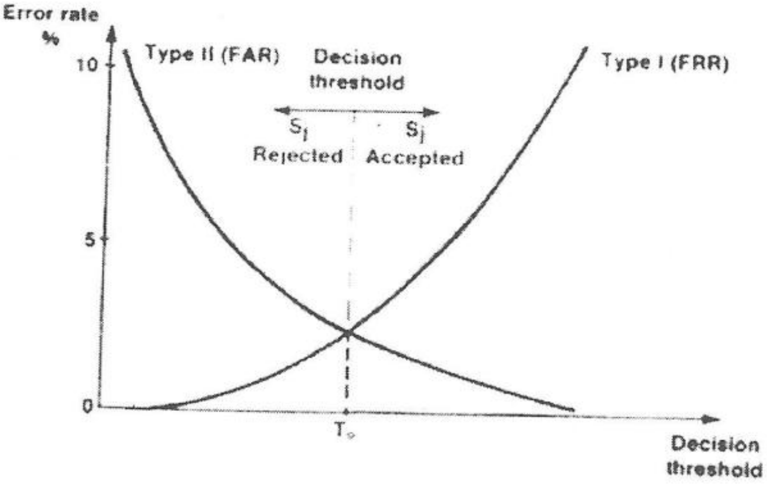


Figure 21: Error rate evaluation [4]

A photograph of the hardware of the biometric system that is ready to be operated can be seen in Figure 22.

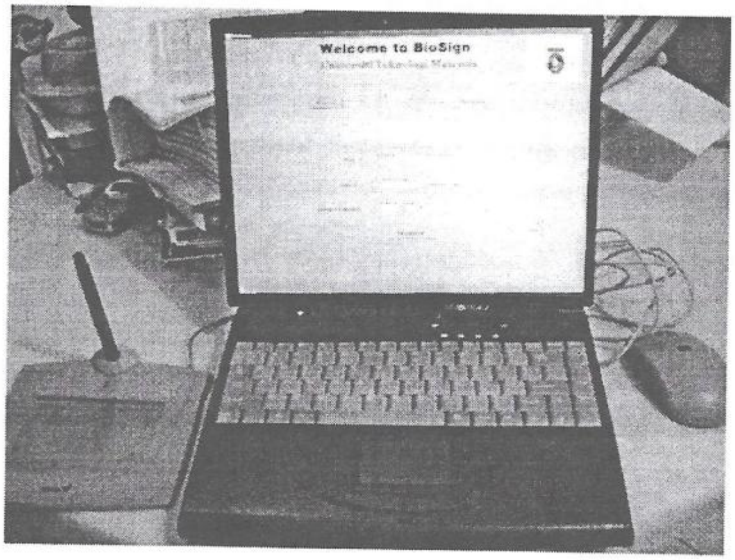


Figure 22: The proposed biometric system

7.0 CONCLUSION

A simple but effective PC-based signature verification system has been developed. The system is able to readily capture the *X-Y* coordinate, pressure signal and time information through the data acquisition system. A number of algorithms have been successfully adapted and applied to preprocess all the input data and later verify the test signatures. The time duration variation is checked before shape of the signature is verified using *Euclidean distance* calculation while the pressure distribution signal is verified using neural network. For performance evaluation, the verification algorithm has achieved 17% Type I error and 0% of Type II error. Future investigation that can be carried out should include other preprocessing or verification algorithms as means of benchmarking and factors that may affect the signature parameters related to pressure, speed and tolerance to enhance the system performance.

ACKNOWLEDGEMENT

The authors would like to acknowledge *Cajun Centurion*, one of the *Eng-Tips Forum* members for his generosity in clarifying the database design.

NOMENCLATURE

τ_{current}	current signature time duration
τ_{mean}	mean time duration of all signatures
i, j	signature set number (1,2, 3, 4) i $\neq$ j
$X^{i}_{\mathrm{transform}}$	*X* coordinate of signature after transformation
X^{i}	original *X* coordinate of ith set signature
$\delta x^{i}_{\mathrm{center}}$	distance of ith signature *X* coordinate centre point from 1st signature *X* coordinate centre point
$Y^{i}_{\mathrm{transform}}$	*Y* coordinate of signature after transformation
Y^{i}	original *Y* coordinate of ith set signature
$\delta y^{i}_{\mathrm{center}}$	distance of ith signature *Y* coordinate centre point from 1st signature *Y* coordinate centre point
θ^{i}_{r}	rotation angle of ith signature
θ^{i}_{s}	start point angle of ith signature
θ^{i}_{e}	end point angle of ith signature
ΘX	scale of *X* coordinate
ΘY	scale of *Y* coordinate

max	maximum value of each X and Y coordinate
min	minimum value of each X and Y coordinate
P_i	ith set of pressure data after interpolation
j_i	function represent ith set signature pressure data
n	points of signature signal (1,2….N), N= last point
d	a measure of the difference between two sequence distance
$E(F)$	weight summation of distances on warping function F
w	weight coefficient
D	Normalized distortion value between two sequences P_i and P_j
M	denominator to compensate the effect of N
$g_n(P_{ij}(n))$	Dynamic Time Warping (DTW) Equation
$\mathrm{d}X^{\mathrm{ij}}$	average *Euclidean distance* between ith and jth signature of X coordinate
$\mathrm{d}Y^{\mathrm{ij}}$	average *Euclidean distance* between ith and jth signature of Y coordinate
$\mathrm{d}P^{\mathrm{ij}}$	average *Euclidean distance* between ith and jth signature of the pressure value
Y_{ave}	average Y coordinate value
Y_{verify}	Y coordinate generated by first network

REFERENCES

1. LecLerc, F. and Plamondon, R., (1994), "*Automatic Signature Verification : The State of the Art – 1989 – 1993*", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 643-660.
2. Cardot, H. and Revenu, M., (1994), "*A Static Signature Verification System Based on a Cooperating Neural Networks Architecture*", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 679-692.
3. Md. Yazid Miskom@Miskan, (1999), Pengecaman Tandatangan Kaedah Rangkaian Neural (Rambatan Balik), Kertas Projek (Sarjana Muda Sains Komputer), Universiti Teknologi Malaysia, Skudai.
4. Plamondon, R. and Lorette, R., (1989),"*Automatic Signature Verification And Writer Identification – The State of The Art*",Pattern Recognition, Vol. 22, no.2, pp. 107-131.
5. Jain, A.K., Griess, F.D. and Connell, S.D., (2000), "*On-Line Signature Verification*", Preprint submitted to Elsevier Science.

6. Nelson, W. and Kishon, E., (1991), "*Use of Dynamic Features for Signature Verification*", IEEE Transactions on Systems, Man and Cybernetics, Vol.1, pp. 201-205.

7. Hamilton, D.J., Whelan, J., McLaren, A. and MacIntyre, I., (1995), "*Low Cost Dynamic Signature Verification System*", Procs. of Conference on European Convention on Security and Dection, No.408.

8. Tham, H.K., Palaniappan, R., Raveendran, P. and Takeda, F., (2001), "*Signature Verification System Using Pen Pressure for Internet and E-commerce Application*", Copyright ISSRE and Chillarege Corp.

9. Hastie, T. and Kishon, E., (1992), "*A Model for Signature Verification,*", Technical Report, AT&T Bell Laboratories.

10. Ma, M.M., Wijesoma, W.S., Sung, E., (2000), "*An Automatic On-line Signature Verification System Based on Three Models*", Conference on Electrical and Computer Engineering, 2000 Canadian, Vol. 2, pp. 890-894.

11. Dullink, H., Daalen, B.V., Nijhuis, J., Spaanenburg, L. and Zuidhof, H., (1995), "*Implementing a DSP Kernel for Online Dynamic Handwritten Signature Verification Using the TMS320 DSP Family*", Texas Instruments Incorporated, France.

12. Wu, Q.Z., Jou, I.C. and Lee, S.Y., (1997), "*On-Line Signature Verification Using LPC Ceptrum and Neural Network*", IEEE Transactions on Systems, Man and Cybernetics—Part B:Cybernetics, Vol. 27, No.1.

13. Martens, R. and Claesen, L., (1997), "*Dynamic Programming Optimization for On-line Signature Verification*", Proc. of the Fourth International Conference on Document Analysis and Recognation 1997, Vol.2, pp. 653-656.

14. Martens, R. and Claesen, L., (1996), "*On-line Signature Verification by Dynamic Time-Warping*", Proc. Of the 13th International Conference on Pattern Recognition 1996, Vol. 3, pp. 38-42.

15. Wirtz, B., (1995), "*Stroke-Based Time Warping for Signature Verification*", Proc. of the Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 179-182.

16. Lee, W.S., (1998), A Neural Network Based On-Line Signature Verification Systems, Dissertation for Doctor of Engineering Degree, University of Detroit Mercy.

17. Mohankrishnan, N., Paulik, M.J. and Khalil, M., (1993), "*Online Signature Verification Using a Nonstationary Autoregressive Model Representation*", International Symposium on Circuits and Systems 1993, pp. 2303-2305.

18. Seong, H.K., Myoung, S. P. and Kim, J.H., (1995), "*Applying Personalized Weights to a Feature Set for Online Signature Verification*", Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 2, pp. 882-885.

19. Nelson, W., Turin, W. and Hastie, T., (1994), "*Statistical Methods for On-line Signature Verification*" International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 749-770.

20. Munich, M.E., "*Application of Hidden Markov Model to Signature Verfication*", California Institute of Technology, 116-81.
21. Penagos, J.D., Prabhakaran, N. and Wunnava, S.V., (1996), "*An Efficient Scheme for Dynamic Signature Verification*", Proc. of IEEE Southeastcon '96, 'Bringing Together Education, Science and Technology', pp.451-457.
22. Lee, L.L., Berger, T. and Aviczer, E, (1996) "*Reliable On-Line Human Signature Verification System*", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.18, No.6, pp. 643-647.
23. Fairhurst, M.C. and Brittan, P., (1994), "*An Evaluation of Parallel Strategies for Feature Vector Construction In Automatic Signature Verification Systems*", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 661 -678.
24. Poyner, R., (1996), "*Wintab^{TM} Interface Specification 1.1:16- and 32- bit API Reference*", LCS/Telegraphics.
25. Yoshimura, I. and Yoshimura, M., (1994), "*Offline Verification of Japanese Signatures After Elimination of Background Patterns*", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 693 -708.
26. Dimuaro, G., Impedovo, S. and Pirlo, G., (1994), "*Component-Oriented Algorithms for Signature Verification*", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 771-793.
27. Myers, C., Rabiner, L.R. and Rosenberg, A.E., (1980), "*Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition*" IEEE Trans. On Acoustics, Speech, and Signal Processing, Vol. ASSP-28, No. 6, pp. 623 – 633.