14

Design and Evaluation of a Pressure Based Typing Biometric Authentication System

MJE Salami, Wasil Eltahir and Hashimah Ali
International Islamic University Malaysia (IIUM)
Malaysia

1. Introduction

Although a variety of authentication devices to verify a user's identity are in use today for computer access control, passwords have been and probably would remain the preferred method. Password authentication is an inexpensive and familiar paradigm that most operating systems support. However, this method is vulnerable to intruder access. This is largely due to the wrongful use of passwords by many users and to the unabated simplicity of the mechanism which makes such system susceptible to unsubstantiated intruder attacks. Methods are needed, therefore, to either enhance or reinforce existing password authentication techniques.

There are two possible approaches to achieve this, namely by measuring the time between consecutive keystrokes "latency" or measuring the force applied on each keystroke. The pressure-based biometric authentication system (PBAS) has been designed to combine these two approaches so as to enhance computer security. PBAS employs force sensors to measure the exact amount of force a user exerts while typing. Signal processing is then carried out to construct a waveform pattern for the password entered. In addition to the force, PBAS measures the actual timing traces, which are often referred to as "latency".

Two approaches to construct user typing pattern have been implemented with PBAS. First approach utilizes a waveform acquired for user keystroke pressure along with time between each keystroke "latency" to create a unique user password typing pattern for authentication. An auto-regressive (AR) classifier is used for the pressure pattern, while a latency classifier is used for the time between keystrokes. The results of both classifiers are combined to authenticate the user typing pattern.

The second approach combines the pressure and latency by creating a pattern of peak keystroke force and latency. By combining the force and time features other classifiers have been tested with PBAS, namely support vector machines (SVM), artificial neural network (ANN), adaptive neuro-fuzzy inference system (ANFIS). Figure 1 illustrates how these classifiers are integrated to develop the system.

As compared to conventional keystroke biometric authentication systems, PBAS has employed a new approach by constructing a waveform pattern for the keystroke password. This pattern provides a more dynamic and consistent biometric characteristics of the user. It also eliminates the security threat posed by breaching the system through online network as the access to the system is only possible through the pressure sensor reinforced keyboard "biokeyboard".

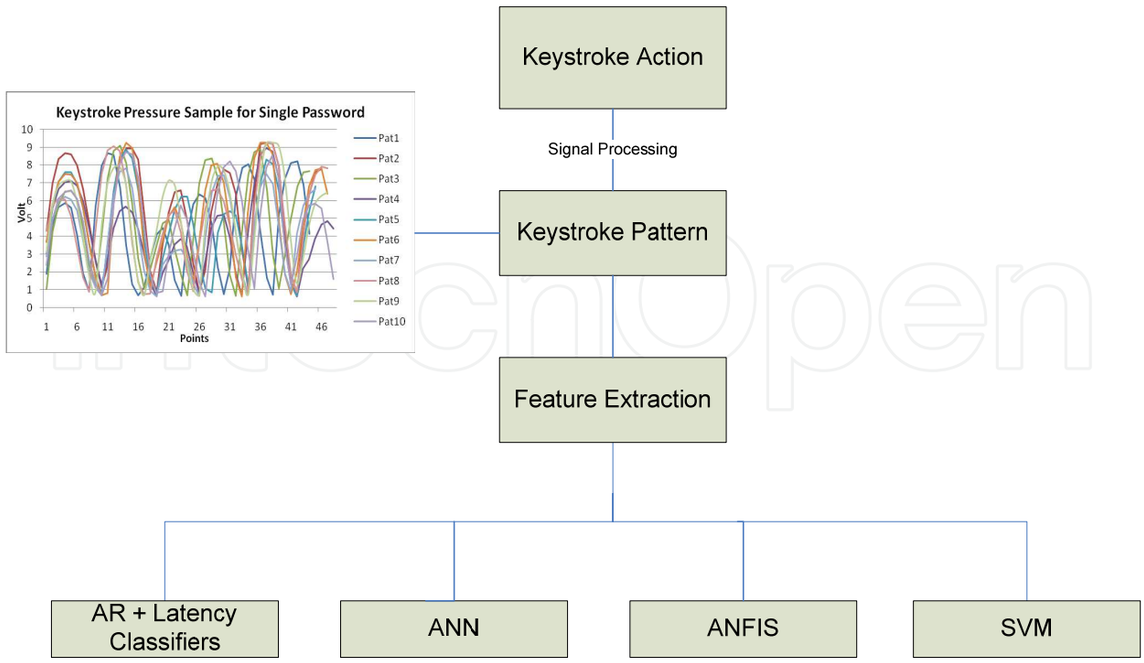

Fig. 1. Keystroke feature extraction and classification.

In order to further reinforce user authentication, PBAS has been integrated with iris system. Figure 2 shows the integrated system which can authenticate users by combining the iris and keystroke authentication mechanisms to achieve optimum results. Preliminary tests on the proposed system have produced promising results.
The detailed design of PBAS system and data classifiers would be subsequently discussed. Some preliminary testing results obtained from the system would also be presented.

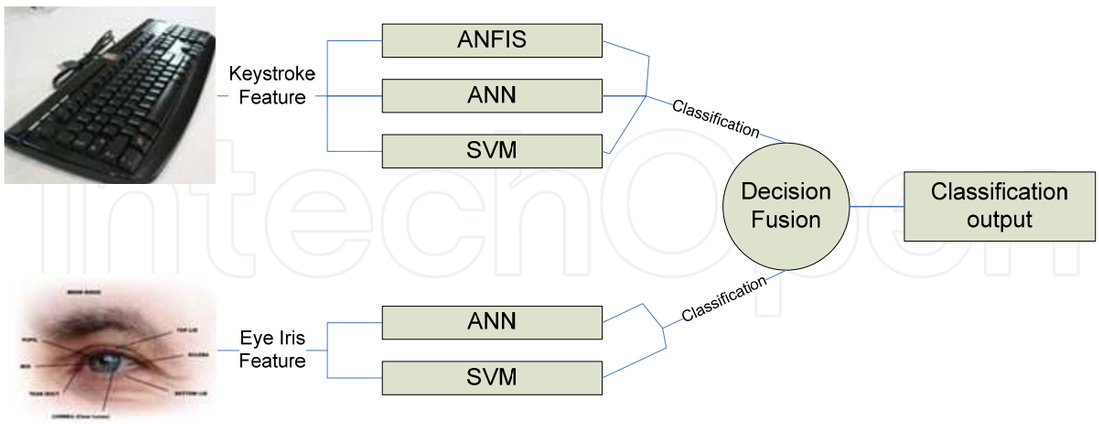

Fig. 2. Decision fusion for eye iris and keystroke classifiers.

2. Pressure based typing biometric authentication system

Keystroke authentication systems available in the market are mostly software-based. This is due to the ease of use as well as the low cost of the mechanism. Any new keystroke

authentication system has to consider these factors in the design. Likewise, the system designed for PBAS uses simplified hardware which minimizes the cost of production.
The system is designed to be compatible with any type of PC. Moreover, it does not require any external power supply. In general, the system components are low cost and commonly available in the market.

2.1 System hardware components

The hardware layout of the system is unique. It consists of a desktop computer equipped with a data acquisition card, and a specially designed alphanumeric keyboard embedded with pressure sensors (Eltahir et al., 2003).
As illustrated in Figure 3, the main hardware components of PBAS are as follows:

1. Alphanumeric keyboard (biokeyboard) embedded with force sensors to measure the keystroke pressure while typing.
2. Data Acquisition System consisting of the following components:
   a. Analogue Interface Box (filtering and amplification of signal).
   b. DAQ PCI card fitted onto the PC.
3. PC/central processing unit (CPU) for running the PBAS program using Windows XP operating system.

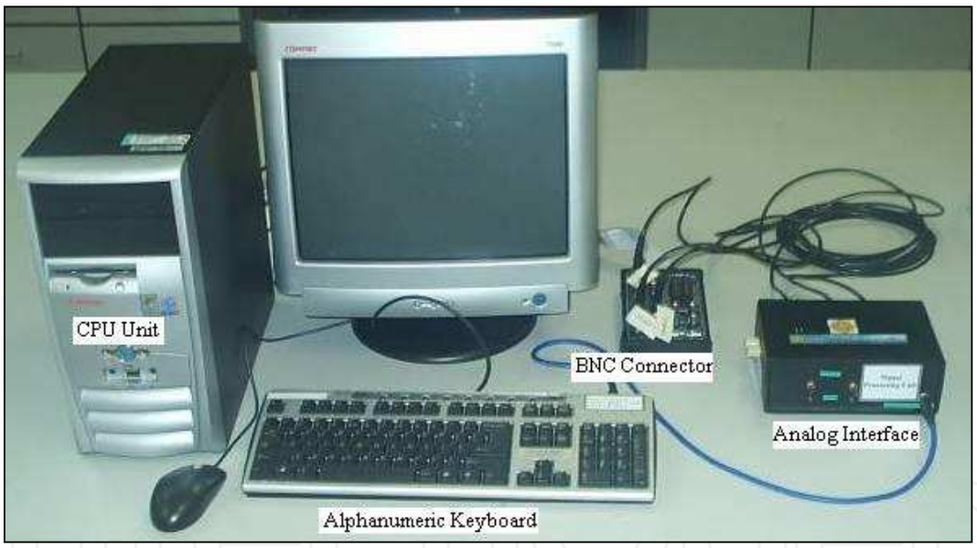

Fig. 3. PBAS system components.

A special keyboard has been manufactured to acquire the alphanumeric password and the keystroke pressure template of the user. The biokeyboard layout is identical to normal commercial keyboard. This is crucial to maintain an intrinsic system that does not alter user typing habits. Figure 4 shows the biokeyboard front, back and side views.
To measure the keystroke pressure, ultra-thin flexible force sensors are fixed below each keyboard key. A plastic spring is fixed between the key and the sensing area to ensure that it does not get dislodged. This is necessary to avoid erroneous readings.
The keyboard operates just as a normal alphanumeric keyboard in addition to measuring keystroke pressure. Thus, the users of this system would not find any differences between this keyboard and the commercial ones.

Fig. 4. Pressure sensitive alphanumeric keyboard (biokeyboard).

2.2 Software layout

The system starts by prompting user to enter his/her user ID and password. The alphanumeric keyboard (biokeyboard) extracts the pressure template for the password entered. At the same time, the system calculates the latency pairs for the entered password and accompanies it with pressure template in a single data file. This data file is transferred to the system's database.

In the learning mode, the user is required to repeatedly key in the password for several times to stabilize his/her keystroke template.

In the authentication mode, the user is requested to enter his/her ID and password. The resulting pressure template and latency vector are compared with those modelled in the database using data classifiers. Depending on the results of this comparison, the user will be either granted or denied access to the system. Figure 5 illustrates the operation of PBAS.

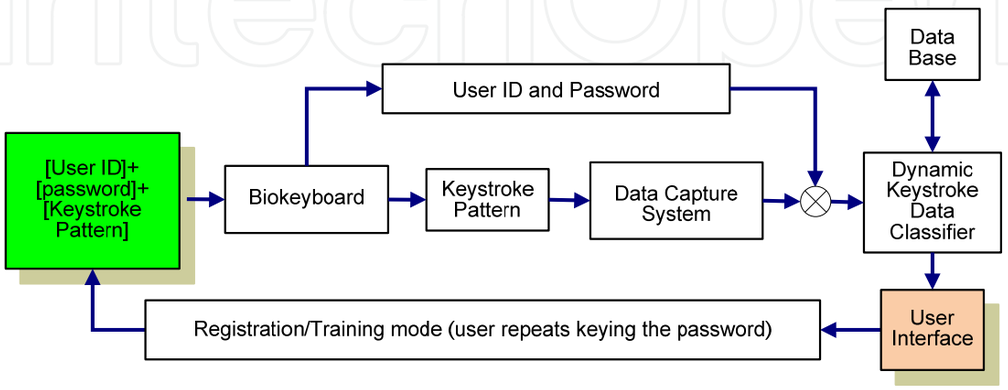

Fig. 5. PBAS block diagram.

2.3 Feature extraction

After acquiring the keystroke pressure template and latency pairs, PBAS applies feature extraction to prepare the pressure template for the specific classifier. The classifiers used in PBAS require two types of pressure templates, while the latency is the same for all classifiers.

A complete template is required for the AR classifier, while ANN, ANFIS, and SVM require a vector of highest pressure score for each keystroke. Figure 6 shows a complete keystroke template for single password entered by the user; this template is saved in the database suitable for the AR classifiers which is based on the signal modelling. Maximum pressure points are recorded and saved separately in the database for ANN, ANFIS and SVM classifiers.

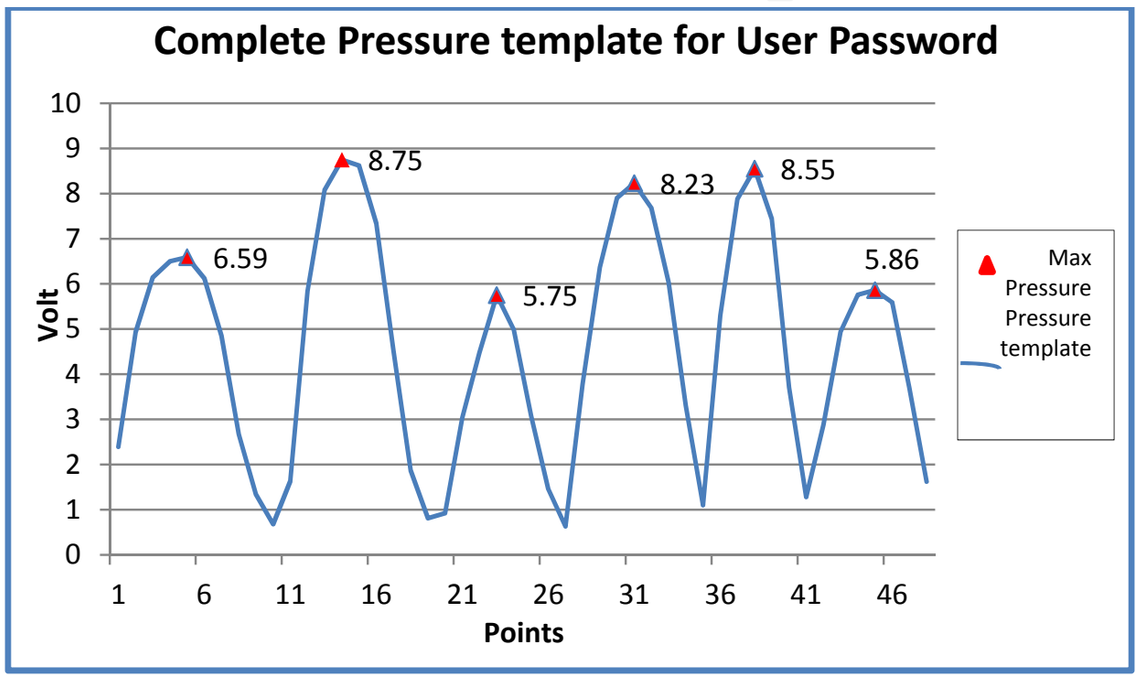

Fig. 6. Complete pressure template for single user password.

2.4 System algorithm and program structure

With the integration of software and hardware, the PBAS algorithm has been designed to have two main operational modes:

1. Training users and creating biometric template profiles; at this stage the user is requested to key-in his/her ID and the user trains his/her password.
2. Authenticating existing users based on their claimed identity; users provide ID and password which are compared with the biometric profiles of the users in the database.

Figure 7 shows the flow graph for the overall PBAS training-authentication processes. The authentication mode consists of two phases:

1. Normal authentication, which involves the password combination and its compliance with the one saved in the database.
2. Biometric authentication, which is done by combining latency with the keystroke pressure classifiers.

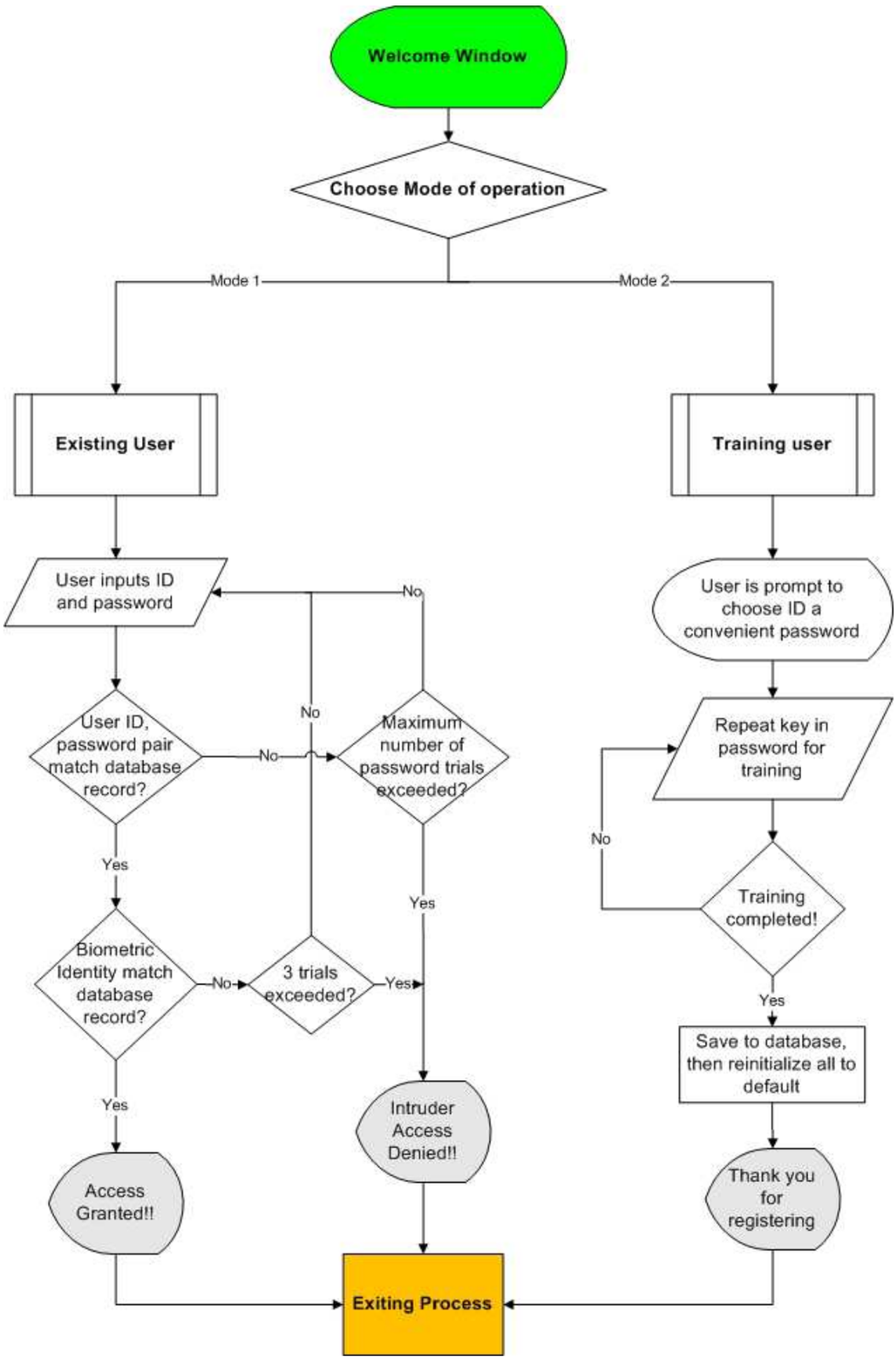

Fig. 7. PBAS main algorithm flowchart.

Firstly, the user will select the mode of operation. In the training mode, the access-control system requests the user to type in the login ID and a new password. The system then asks the user to re-enter the user ID and password to train the new password. The resulting latency and pressure keystroke templates are saved in the database. During training, if the user mistypes the password the system prompts the user to re-enter the password from the beginning. The use of backspace key is not allowed as it can disrupt the biometric pattern. When registration is done, the system administrator uses these training samples to model the user keystroke profile. The design of the user profile is done offline. Finally, the system administrator saves the user's keystroke template model along with the associated user ID and password in the access-control database.

In the authentication mode, the access-control system requests the user to type in the login ID and a password. Upon entering this information the system compares the alphanumeric password combination with the information in the database. If the password does not match, the system will reject the user instantly and without authenticating his/her keystroke pattern. However, if the password matches then the user keystroke template will be calculated and verified with the information saved in the database. If the keystroke template matches the template saved in the database, the user is granted access.

If the user ID and alphanumeric password are correct, but the new typing template does not match the reference template, the security system has several options which can be revised occasionally. A typical scenario could be that PBAS advises a security or network administrator that the typing pattern for a user ID and password is not authentic and that a security breach might be in progress. The security administrator can then closely monitor the session to ensure that the user does nothing un-authorized or illegal.

Another practical situation applies to the automatic teller machine "ATM" system. If the user's password is correct but the keystroke pattern does not match, the system can restrict the amount of cash withdrawn on that occasion to minimize the possibility (and penalty) of theft.

3. Dynamic data classifiers

3.1 Latency classifier

Identity authentication using keystroke latency timing for password digraphs has been investigated thoroughly by many researchers (Joyce & Gupta, 1990; Monrose & Ruben, 1997; De Ru & Eloff, 1997; Ord & Furnell, 2000; Araujo et al, 2005). Many useful methodologies have been presented and are in use with the current latency keystroke authentication systems available in the market.

Joyce and Gupta discussed the design of an identity verifier based on four input strings (login name, password, first name, last name). The verification is done by comparing the mean reference signature M with a test signature T. The norm $\|M-T\|$ is computed and if it is less than the threshold for the user, the attempt is accepted; otherwise it is flagged as an imposter attempt.

Though this approach produces some satisfactory results, it requires a relatively lengthy input string (Joyce & Gupta, 1990). A modified approach has been devised here for PBAS latency authentication. PBAS uses only the password string for latency verification.

3.1.1 Creating mean reference latency vector

1. Registering users are prompt to re-enter their password several times, latency vector for each trial is saved in an individual data file resulting in *(n)* number of files in the database, where *n* is the number of trials.

2. Data treatment is applied on the data files to remove outliers and erroneous values.
3. An average latency vector is calculated using the user trial sample. This results in a single file containing the mean latency vector *(R)* for *n* password trials. This file is used as reference for latency authentication.

3.1.2 Calculating suitable threshold

Thresholding is used to decide an acceptable difference margin between the reference latency vector *(R)* and the latency vector provided by the user upon verification *(V)*. The threshold is computed based on the data files saved in the database. A threshold is set for each user based on the variability of his/her latency signatures. A user that has little variability in his/her latencies would have a small threshold, while another with high variability should have larger threshold. Standard deviation is the variability measure used. The standard deviation between the mean *(R)* latency vector and the user sample is computed. A threshold based on the standard deviation is calculated by the following equation:

$$\sum_{k=1}^{m-1} \left| R_k - V_k \right| \le c * d \tag{1}$$

where m is the password length, R_k is the *kth* latency value in the reference latency vector, V_k is the *kth* latency value in the user-inputted latency vector, c is an access threshold that depends on the variability of the user latency vector, d is the distance in standard deviation units between the reference and sample latency vectors.

In order to classify user attempt, the latency score S_L for the user attempt is defined as

$$S_L = \frac{\sum_{k=1}^{m-1} \left| R_k - V_k \right|}{c * d} \tag{2}$$

Therefore, depending on the value of S_L the classifier output can be expressed as

$$S_L \begin{cases} \le 1, & accept\ \ template \\ > 1, & reject\ \ template \end{cases} \tag{3}$$

Table 1 shows the reference latency vector for user "MJE1" which was calculated by the above mentioned method for a sample of 10 trials. Five latency vectors are used to test the threshold c for this reference profile (Table 1). The standard deviation was calculated to be S_y=*46.5957 m.sec* and a threshold of two standard deviations above the mean (c=*2.0*) resulted in the following variation interval *253.9748* ≥ *R-V* ≥ *67.83189*.This threshold takes in all 5 trials of the user. However, this is a relatively high threshold value and in many practical situations such values would only be recommended for unprofessional users who are usually not very keen typists. The user here is a moderate typist. This is evident by his/her relatively high standard deviation. High standard deviation is also a measure of high variability in the users' typing pattern; this usually indicates that the user template has not yet stabilized, perhaps due to insufficient training.

Table 2 shows the variation of threshold values *c* (from 0.5 to 2.0) and its effect on accepting the user trials. For this user, a threshold value that is based on standard deviation of 2.0 provides an acceptance rate of 100% (after eliminating outliers). However, a high threshold value would obviously increase the imposter pass rate. Therefore for normal typists, the threshold values should only be within the range of 0.5 to 1.5.

An experiment was conducted to assess the effect of varying the latency threshold value on the false acceptance rate (FAR) and the false rejection rate (FRR). In this experiment an ensemble for 23 authentic users and around 50 intruders were selected randomly to produce authentic and intruder access trials. Authentic users were given 10 trials each while intruders were given 3 trials per account. All trials were used for the calculations and no outliers were removed.

Password (asd123)	Reference Vector	T 1	T 2	T 3	T 4	T 5
a-s	201.9231	203	187	172	203	235
s-d	162.8571	156	188	171	156	93
d-1	316.2308	235	187	219	188	250
1-2	185.2000	187	203	203	187	203
2-3	108.5714	110	110	110	94	79

Table 1. Reference latency tested against 5 authentic user trials.

Standard Deviation	Upper Limit	Lower Limit	Acceptance Percentage
0.5	184.1712	137.6355	40
1.0	207.439	114.3676	60
1.5	230.7069	91.09975	80
2.0	253.9748	67.83189	100

Table 2. Effect of threshold value on user acceptance rate.

Figure 8 shows that the equal error rate (EER) for the FAR and the FRR rates was 24% and it occurred at a threshold value of 2.25. This relatively high FRR rate is expected since the password strings used were mainly short in length and weak in strength.

3.2 AR-burg classifier

The AR algorithm uses the notion of signal analysis to reproduce the users' keystroke pressure template which is then compared with the keystroke template produced by the alleged intruders. Based on this comparison an authentication decision is made.

A signal model approach is advocated here since the pressure template points are interrelated across time. The AR signal model is expressed as

$$y(n) + a(1)y(n-1) + a(2)y(n-2) + \ldots + a(p)y(n-p) = x(n) \tag{4}$$

where n is the time index, $y(n)$ is the output, $x(n)$ is the input, and p is the model order.

For signal modelling $y(n)$ becomes the signal to be modelled and the $a(i)$ coefficients need to be estimated.

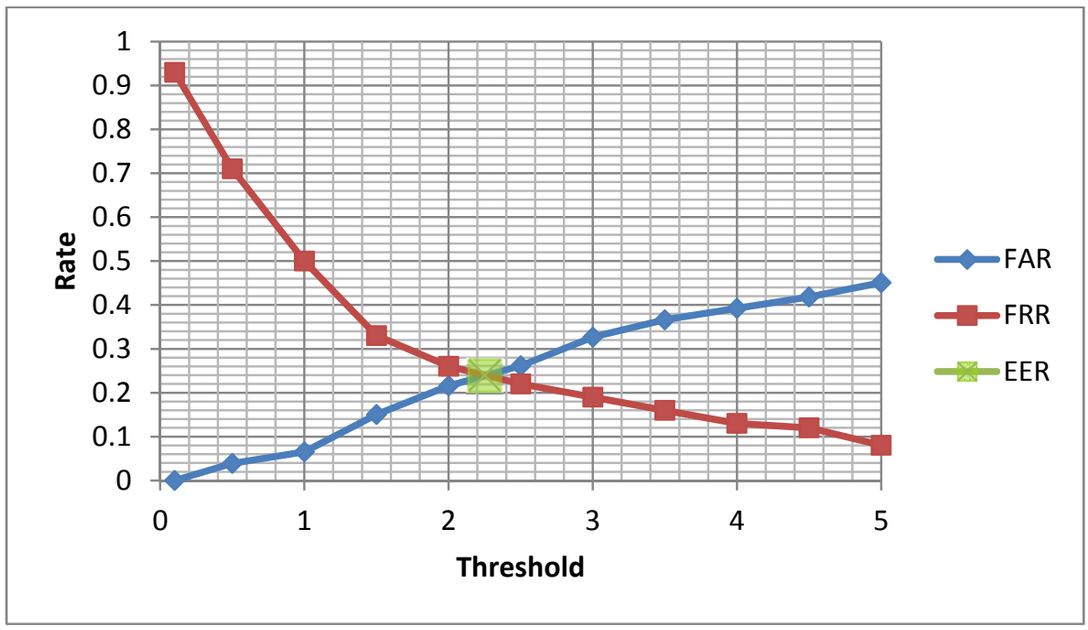

Fig. 8. Latency threshold versus FAR and FRR rates.

Equation (4) can be used to predict future values of the signal $\hat{y}(n)$ *where*

$$\hat{y}(n) = -a(1)y(n-1) - a(2)y(n-2) - \ldots - a(p)y(n-p) \tag{5}$$

Denoting the model error as $e(n)$ then

$$y(n) + a(1)y(n-1) + a(2)y(n-2) + \ldots + a(p)y(n-p) = e(n) \tag{6}$$

The Total Squared Error *(TSE)* for the predicted signal is

$$TSE = \sum_{n=1}^{N-1} e_n^2 \tag{7}$$

The AR model is used most often because the computation of its parameters is simpler and more developed than those of moving average *(MA)* and autoregressive moving average *(ARMA)* models (Shiavi, 1991; Manson, 1996).

Burg method has been chosen for this application because it utilizes both forward and backward prediction errors for finding model coefficients. It produces models of lower variance (S_p^2) as compared to other techniques (Shiavi, 1991).

Authentication is done by comparing the *TSE* percentage saved in the users' database with that generated by the linear prediction model. Previous experiments proved that authentic users can achieve TSE margin of less than 10% (Eltahir et al., 2004).

3.2.1 Identifying optimum pressure template for AR model

An algorithm has been developed in MATLAB to identify the best pressure template among the user sample. This pattern is used for estimating the AR model parameters for the user keystroke pressure pattern. The algorithm uses the correlation technique to calculate the

accumulative correlation index "ACI", which is the accumulation of the correlation between each pressure pattern and the whole sample. The pattern with the highest ACI is selected for the model.

3.2.2 Identifying the optimum TSE acceptance margin

The TSE relative prediction error *RPE* is calculated according to

$$\text{Relative Prediction Error} = \left| \frac{TSE_m - TSE_s}{TSE_m} \right| \tag{8}$$

where TSE_m is the *TSE* calculated for the user's AR-Burg model in database. TSE_s is the *TSE* for the pressure pattern of the user.

Classification of user attempt is done by comparing *RPE* to threshold *T* according to the following equation:

$$RPE \begin{cases} \leq T, \ accept \ template, \\ > T, \ reject \ template, \end{cases} \qquad where \ 0 < T \leq 1 \tag{9}$$

Based on previous research experiments it has been reported that authentic users can achieve up to 0.1 *RPE* while intruders exhibiting unbounded fluctuation can have *RPE* above 3 (Eltahir et al., 2004).

An experiment was conducted to assess the effect of varying the TSE threshold value on the FAR and FRR rates. In the experiment an ensemble of 23 authentic users and around 50 intruders were selected randomly to produce authentic and intruder access trials. Authentic users were given 10 trials each while intruders were given 3 trials per account. All trials were used for the calculation of results with no outliers removed. Figure 9 shows the variation of FAR and the FRR against the TSE threshold values. The EER was 25% and it was recorded at TSE of 37.5%. When compared to latency, TSE has lower FRR spread out as the threshold is increased.

The AR modelling algorithm has been implemented in the following order:

1. The user is prompted to enter the password several times.
2. The optimum pattern for modelling the user pressure template is identified using the ACI values obtained from the sample.
3. The best AR model order is determined based on the final prediction error (FPE) and the Akaike's information criteria (AIC) (Akaike, 1970).
4. The AR model is constructed and the model coefficients are saved for user verification.
5. When a user enters a password, the linear prediction model is constructed to create a pressure template from the users' keystroke. TSE_s is calculated for this template.
6. From the system database, TSE_m is used to calculate the RPE score.
7. If $RPE \leq T$ user is authentic, whereas if $RPE > T$ then user is an intruder.

3.3 Combined latency and AR classifiers

The receiver operating characteristic curve (ROC) is used to assess the effect of the threshold value on the FAR and FRR rates. ROC curve assesses the trade-off between low intruder

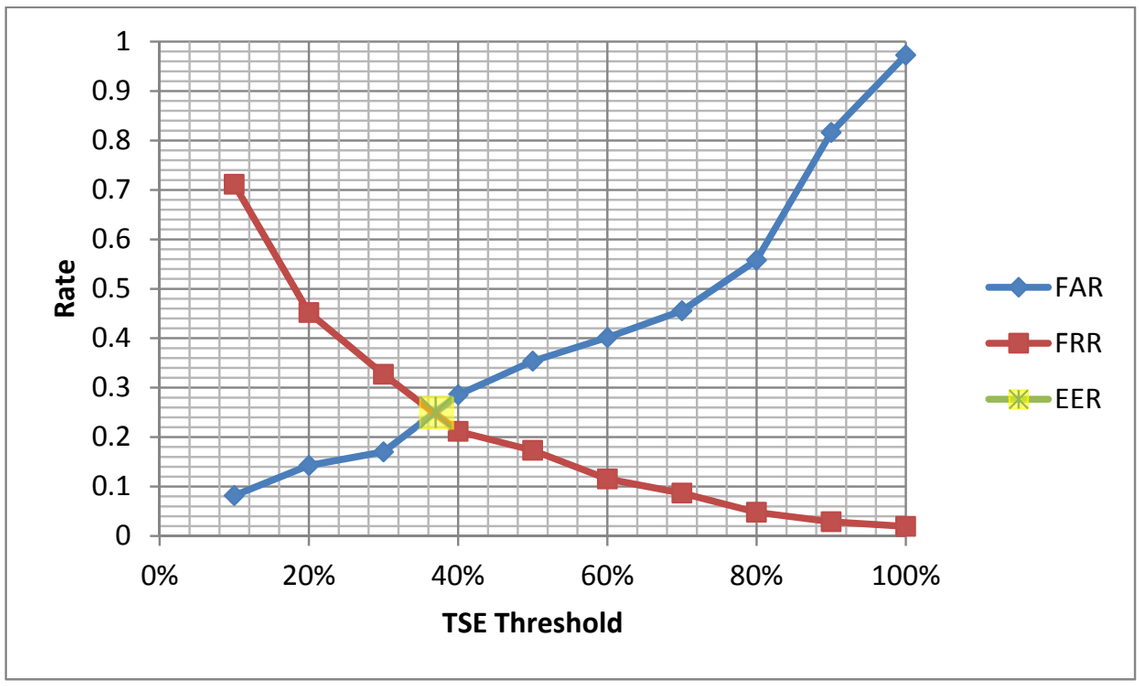

Fig. 9. TSE threshold versus FAR and FRR rates.

acceptance rate (IAR) and high authentic acceptance rate (AAR) as the decision threshold value varies. Figure 10 shows that the latency classifier has slightly better separation than the AR classifier. In addition, the latency classifier has better intruder rejection rate whereas AR classifier has a higher true pass rate. This graph also shows that the performance of both

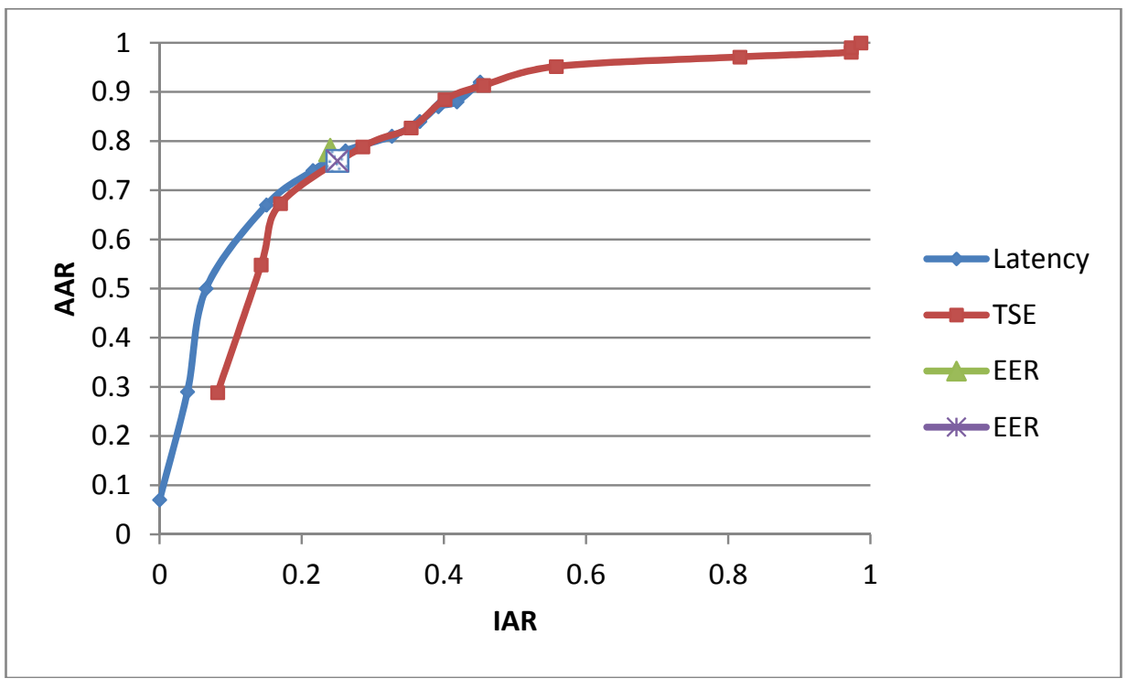

Fig. 10. ROC showing performance of latency and TSE.

classifiers at the EER points is very similar; therefore it is expected that by combining both algorithms the overall system performance would be improved.
The threshold used for the TSE classifier is $T = 0.4$ based on the calculated EER. As for the latency threshold c, it is recommended to use a threshold value of 2.0 to 2.25 for unprofessional typists and 1.0 to 1.5 for professional typists.
An experiment has been conducted with the combined latency and AR classifiers, whereby 23 users participated in the data collection. All results were calculated online. The results showed FRR of 3.04 and FAR of 3.75 (Eltahir et al. 2008).

3.4 Artificial Neural Network (ANN)

In general, an ANN is characterized by its architecture, learning algorithm, and activation function. The architecture describes the connections between the neurons and consists of an input layer, an output layer and at least one hidden layer.

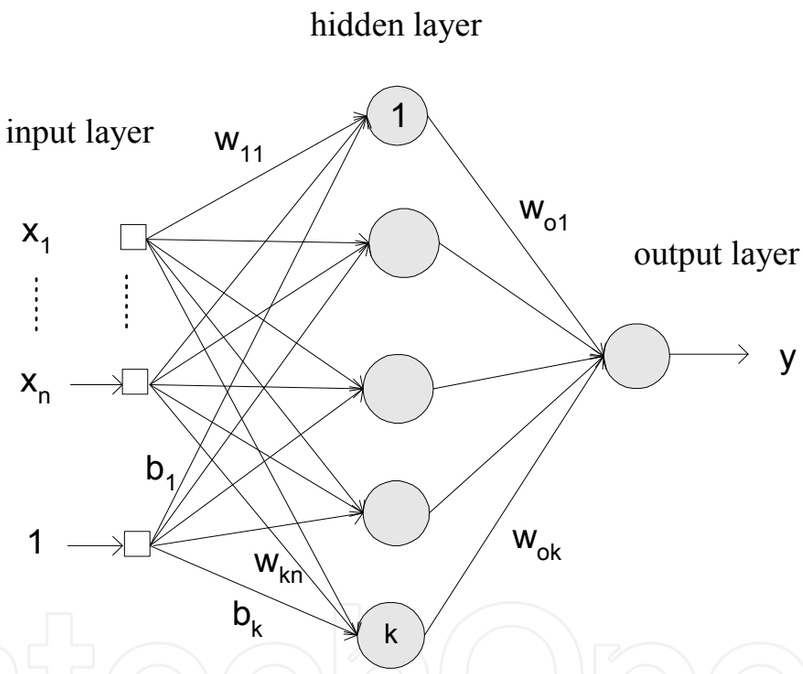

Fig. 11. A Typical multilayer feedforward neural network structure.

Among the available architectures of the ANN, Multilayer Feedforward Network (MFN) is a very popular network. The *learning algorithm* or *adaptation rule* refers to the way the weights are changed in order to achieve a desired mapping between input and output. The backpropagation (BP) learning algorithm is one of the most important historical developments in neural network (Lin & George, 1995). It is an iterative gradient descent algorithm designed to minimize the *mean square error* (*MSE*) between the actual output of a backpropagation neural network (BPNN) and the desired output (Jang et al., 1996). MFN trained with backpropagation algorithm has been widely used in many engineering applications due to its capability as a general approximation. Figure 11 shows the MFN structure with input $x_1, x_2, \ldots, x_n$ and an output y. The connections between the neurons of the different layers are called weight and bias.

3.4.1 ANN keystroke model

Using multilayer feedforward network and the backpropagation algorithm, the process of developing keystroke models based on the typing biometric data can be summarized in the following steps:

1. The weight of the network is initialized with random values and the number of iteration *(n)* is set to zero.
2. Enter the input vector *p* from the training dataset to the network for the first time.
3. Send the input vector *p* through the network to get an output

$$o_{pk} = f\left(\sum w_{jk} f\left(\sum_m w_{mj} f\left(\ldots f\sum_i w_{il} x_i\right)\right)\right) \tag{10}$$

4. Evaluate the error value by computing the difference between the real output and the desired output: calculate also the sum of the squared error and increment *n*.
5. Carry out the back-propagation of output layer error to all elements of network.
6. For every unit of output k, calculate,

$$\delta_k = \left(O_k - y_k\right) f'\left(net_k\right) \tag{11}$$

7. For every hidden unit j, calculate,

$$\delta_j = f'\left(net_j\right)\sum_k \delta_k w_{jk} \tag{12}$$

8. Update the weights of all elements between output and hidden layer and then between all hidden layers moving toward the input layer. Changes of the weights can be obtained as

$$\Delta w_{ij}^{(n+1)} = \eta \delta_j o_i + \alpha \Delta w_{ij}^{(n)} \tag{13}$$

   where α is a constant called momentum which serves to improve the learning process.
9. Repeat the step 2-8 with the next input vector until the error becomes small or maximum number of iterations is reached.

3.4.2 Training and evaluating the ANN based model

To evaluate the effectiveness of the proposed classifier, a group of five (5) participants was used; the collected data size was 1000. Three (3) of the participants were considered as authorized persons while the other two (2) were considered as imposters. Each person was required to type a password of six characters of his/her own choice for 200 times, 100 data were used for training whereas the rest were used for data testing. Each typed character of the password had maximum pressure and latency features. For example, password "asd123" consists of five latency and six maximum force features, resulting in eleven features for each trial. The performance of the classifier was assessed by FRR and FAR rates.

Table 3 shows the result of training the user data. User 1 and user 3 have single hidden layer with fifteen and six neurons respectively. Meanwhile, user 2 has two hidden layers consisting of seven (7) and six (6) neurons for the first and the second hidden layers respectively. The combined features (maximum pressure and latency) gave perfect classification rates for all users. Furthermore, the MFN model can be trained in less than one second.

Model	Authorized User	Training Time(sec)	Classification Rate (%)
[15 1]	User 1	0.9062	100
[7 6 1]	User 2	0.8125	100
[6 1]	User 3	0.5312	100
	Average	0.7499	100

Table 3. Training performance of ANN user model.

Table 4 shows the user verification results for the user model. The proposed system gives good average FRR which is about 0.67.

Model	Authorized User	FRR (%)	FAR (%) Close Set	FAR (%) Open Set
[15 1]	User 1	1	0	5
[7 6 1]	User 2	1	0	38
[6 1]	User 3	0	0	0.5
	Average	*0.67*	*0*	*14.5*

Table 4. Verification performance of ANN user model.

3.5 Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS is an architecture which is functionally equivalent to a Sugeno type fuzzy rule base (Jang et al., 1996). It is a method for tuning an existing rule base with a learning algorithm based on a collection of training data.

ANFIS is an approach that integrates the interpretability of a fuzzy inference system with the adaptability of a neural network. Moreover, because ANFIS has much less tuneable parameters than traditional neural networks, it has the advantage of being significantly faster and more accurate than many ANN-based methods.

An ANFIS architecture corresponding to a Sugeno fuzzy model for two inputs and single output is shown in Figure 12. It is a feedforward network consisting of 5 layers. To simplify

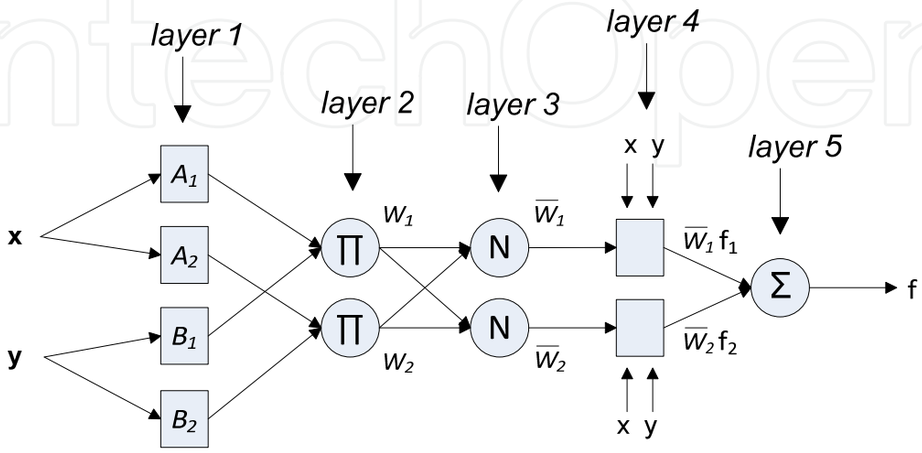

Fig. 12. ANFIS architecture (Jang et al., 1997).

the notation, consider the two inputs, x and y and one output f. A rule in the first order Sugeno fuzzy inference system has the form:
Rule 1: If x is A_1 and y is B_1, then

$$f_1 = p_1x + q_1y + r_1$$

Rule 2: If x is A_2 and y is B_2, then

$$f_2 = p_2x + q_2y + r_2$$

The parameters p_1, p_2, q_1, q_2, $r_1$and r_2 are linear, whereas A_1, A_2, B_1 and B_2 are nonlinear.
The five layers in ANFIS are fuzzification, production, normalization, defuzzification, and aggregation layer arranged in this order. The following are the input and output relationships for each layer:
Layer 1(Fuzzification layer): A_1, A_2, B_1 and B_2 are the linguistic expressions which are used to distinguish the membership functions (MFs). The relationships between the input-output and MFs are:

$$O_{1,i} = \mu_{A_i}(x) \text{ , for } i\text{= 1, 2} \tag{14}$$

$$O_{1,j} = \mu_{B_j}(y) \text{ , for } j\text{= 1, 2,} \tag{15}$$

where $O_{1,i}$ and $O_{1,j}$ denote the output function of the first layer, $\mu_{A,i}(x)$ and $\mu_{B,j}(y)$ denote MFs so that

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \tag{16}$$

where $\{a_i, b_i, c_i\}$ is the parameter set. As the values of these parameters change, the bell-shaped functions vary accordingly. In fact, any continuous and piecewise differentiable functions such as trapezoidal or triangular–shaped functions are also qualified for use as MF. Parameters in this layer are referred to as *premise parameters*.
Layer 2 (Production layer): Every node in this layer is marked as symbol Π the outputs are w_1 and w_2 , the weight functions of the next layer are

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y) \text{ , } i\text{=1, 2.} \tag{17}$$

where $O_{2,i}$ is the output function of the second layer.
Layer 3 (Normalization layer): The node is marked as **N**, and it is used to normalize the weight functions, that is

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2} \text{ , } i\text{=1, 2.} \tag{18}$$

where $O_{3,i}$ is the output function of the third layer. For convenience, outputs of this layer are called normalized firing strengths.
Layer 4 (Defuzzification layer): Being an adaptive node, $\overline{w_i}$ is the output and $\{p_i, q_i, r_i\}$ is the parameter set in this layer. The relationship between input and output is

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{19}$$

where $O_{4,i}$ is the output function of the fourth layer. Parameters in this layer are referred to as consequent parameters.

Layer 5(Total output layer): The single node in this layer is an output corresponding to the aggregate of all inputs so that the overall output is

$$O_{5,1} = \sum_i \overline{w_i} f_i = \frac{\sum_i \overline{w_i} f_i}{\sum_i \overline{w_i}} \tag{20}$$

where $O_{5,i}$ is the output function f of the fifth layer.

3.5.1 ANFIS keystroke model

The steps for creating an ANFIS model for user typing pattern are as follows:

1. User retypes password until sufficient data is acquired for feature extraction.
2. Determination of the premise parameters.
3. Training of the ANFIS using the input pattern and the desired output to obtain the consequent parameters.
4. Validation of the trained ANFIS system.

3.5.2 Training and evaluating the ANFIS based model

In order to evaluate the user ANFIS typing model, an experiment was carried out with a group of seven (7) persons with data sample size of 1400. Five (5) persons were considered as authorized users, while the other two (2) persons were assumed as impostors. Each person was required to type a password of six characters of his/her own choice for 200 times. 100 samples were used for training, whereas the rest were used for data testing. Both impostors had 200 trials to break into the system. Table 5 shows the training results using a cluster radius of 0.25. It is noted that the training time was relatively long (about 45 seconds).

Model	Authorized User	Training Time(sec)	Classification Rate (%)
ANFIS 1	User 1	47.4688	100
	User 2	45.9686	100
	User 3	45.6406	100
	User 4	43.5781	100
	User 5	43.2344	100
	Average	45.1781	100

Table 5. Training performance of ANFIS of radius =0.25.

By increasing the cluster radius to 0.5, the training time was reduced dramatically to 1.75 seconds. Furthermore, classification rate was maintained at 100% except for user 4 which dropped slightly to 98%.

Furthermore, increasing the radius cluster to 1, training time was further reduced to 1.6 seconds while maintaining the average classification rate at 99.6% as shown in Table 6.

Model	Authorized User	Training Time(sec)	Classification Rate (%)
ANFIS 3 (R=1)	User 1	1.5469	100
	User 2	1.5625	99
	User 3	1.5625	100
	User 4	1.5781	99
	User 5	1.5469	100
	Average	1.5594	99.6

Table 6. Training performance of ANFIS of radius =1.

It is observed that the training time varies significantly for different cluster radius. A smaller radius leads to a longer training time due to the formation of smaller clusters in the data, this creates more rules and a larger number of consequent parameters to be tuned and optimized in ANFIS system. Hence it can be concluded that a larger radius is preferable to achieve shorter training time. However, larger radius tends to give lower accuracy as compared to smaller radius.

Table 7 shows performance results for a radius of 0.25. The FRR is about 9%, FAR for closet set was 3.2% which is relatively low. However, FAR for open set is relatively very high for user 2 and user 5.

Authorized User	FRR (%)	FAR (%) Close Set	FAR (%) Open Set
User 1	0	1.5	1
User 2	8	1.75	39.5
User 3	17	0.5	0.5
User 4	16	10	14
User 5	4	2.25	47.5
Average	9	3.2	20.5

Table 7. Testing performance of ANFIS of radius = 0.25.

Table 8 shows the results obtained when the radius was increased to 0.5 in which the FRR results were significantly improved. The average value for FRR dropped to 3.2 as compared to 9 in previous results. The FAR for closet set is greatly improved with an average value of 0.45. However, FAR for open set shows only slight improvement.

Authorized User	FRR (%)	FAR (%) Close Set	FAR (%) Open Set
User 1	1	0	1
User 2	12	0.25	31.5
User 3	1	0.5	3
User 4	1	0.5	0
User 5	1	1	48
Average	3.2	0.45	16.7

Table 8. Testing performance of ANFIS radius = 0.5.

By selecting a cluster radius of 1, the performance of the ANFIS model improves further as shown in Table 9. The average FRR dropped to 1.2, while FAR for the close set dropped to an average value of 0.15. However, FAR of the open set is still relatively high with an average of about 15.7. Therefore, further improvement is needed to enable the proposed system to produce a small FAR (of the open set), that is smaller than 1.

Authorized User	FRR (%)	FAR (%) Close Set	FAR (%) Open Set
User 1	0	0	1
User 2	5	0	29
User 3	0	0.5	0
User 4	0	0	0
User 5	1	0.25	48
Average	1.2	0.15	15.7

Table 9. Testing performance of ANFIS radius = 1.

In terms of FRR and FAR, it is observed that a radius of 1 produced the best results here.

3.6 Support Vector Machines based keystroke model

Support Vector Machines (SVMs) is a relatively new learning machine technique, which is based on the principle of structural risk minimization (minimizing classification error). SVM is a binary classifier that optimally separates two classes of data (Burges, 1998). Figure 13 shows the flowchart of the SVM algorithm.

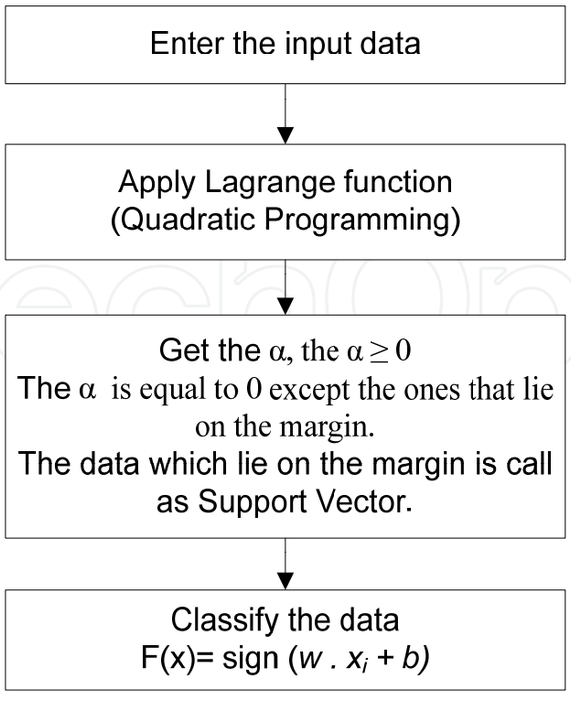

Fig. 13. Flowchart of SVM algorithm.

3.6.1 Training and evaluating the SVM based model

In order to evaluate the user SVM typing model, an experiment was carried out with a group of seven (7) persons. The sample size was 1400. Five (5) of the participants were considered as authorized users, while the other two (2) participants were considered as impostors. Each user was required to type a password of six characters of his/her own choice for 200 times, 100 trials were used for training whereas the rest were used for data testing. Both impostors had two hundred (200) trials to break into the system.

SVM with fifth-order polynomial kernel function was used for developing keystroke typing biometric models. Each authorized user had his/her individual SVM–based model characterized by a set of support vectors. The support vectors are obtained by developing quadratic algorithm for use in the MATLAB toolbox. A penalty term C of 100 is used to anticipate misclassified data.

Table 10 shows the training time and the classification results obtained, it is observed that the entire SVM–based models of authorized users gave perfect classification of 100%, i.e. there were no errors in identifying the authorized users. Furthermore, all the SVM–based models of authorized users can be trained in a relatively short time of less than 0.5 second on average. Therefore, it can be concluded that the SVM models produce promising results when used in the proposed system.

Authorized User	Training Time(sec)	Classification Rate (%)
User 1	0.0625	100
User 2	0.6563	100
User 3	0.4687	100
User 4	0.6250	100
User 5	0.2815	100
Average	0.4188	100

Table 10. Training performance of SVM.

By examining the results shown in Table 11 it is observed that the average FRR rate is 5.6% which is acceptable as compared to the previous results. The average FAR for close set was 1.75% which is relatively good. However, the FAR for open set is high, user 5 has FAR of 48% which suggests that his password typing pattern is weak and easy to imitate, whereas user 4 has a strong password pattern that is hard to imitate. Further improvements are needed as the average value for the FAR is 14.7.

Authorized User	FRR (%)	FAR (%) Close Set	FAR (%) Open Set
User 1	2	0	10
User 2	7	1.75	14
User 3	19	0.5	1.5
User 4	0	0.75	0
User 5	0	1.75	48
Average	5.6	0.95	14.7

Table 11. Testing performance of SVM.

Many experiments have been conducted which indicate that the proposed system, involving the combined maximum pressure and latency features could produce low FRR and FAR values.

4. IRIS recognition

The number of features in human iris is large. Its complex pattern can contain many distinctive features such as arching, ligaments, furrows, ridges, crypts, rings, corona, freckles and zigzag collarette. Iris pattern can have up to 249 independent degrees of freedom; therefore it is very difficult to falsify an iris pattern. As a result of this, iris pattern is a very attractive feature for user verification (Daugman, 2003; Masek, 2003; Wildes, 1997).

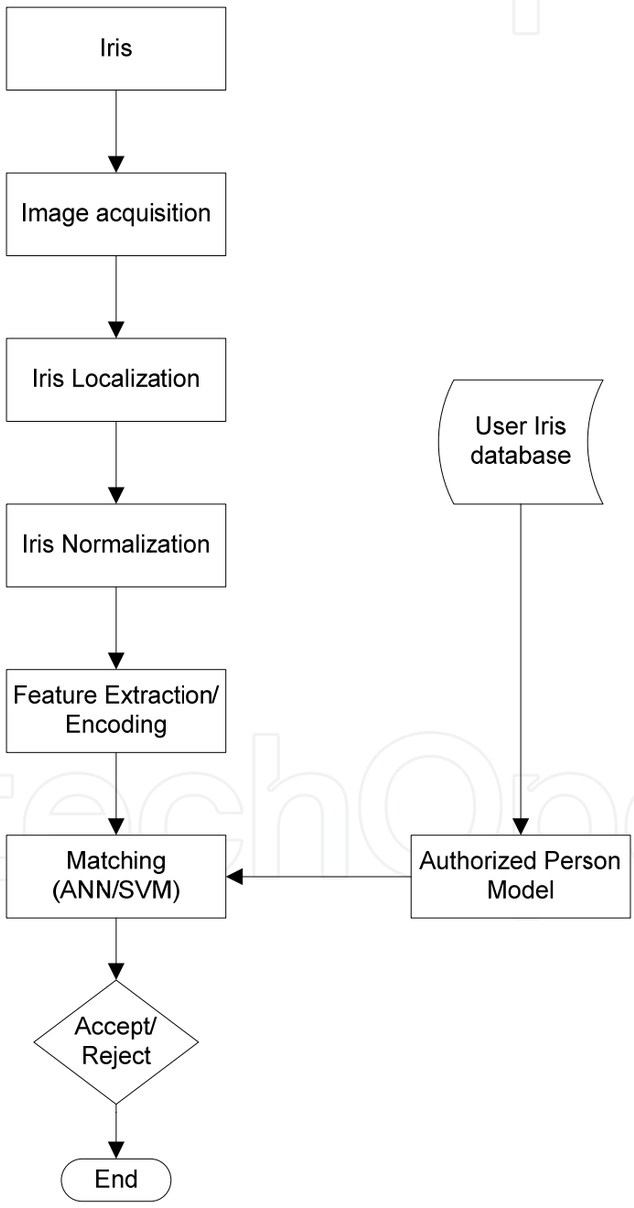

Fig. 14. Basic structure of iris–based verification system.

The first stage of the iris recognition process is to isolate the actual iris region in a digital image so as to localize an acquired image that corresponds to the iris.
Figure 14 shows the complete diagram of iris authentication system. ANN and SVM techniques are used for pattern matching and classification in order to recognize the authorized user.

4.1 ANN based IRIS model

Multilayer Feedforward Network (MFN) architecture is also used in developing the iris – based model. The process of developing iris model through extracted features using MFN is as follows:

1. Acquisition of iris image which is then analysed to obtain the extracted features.
2. Determination of the structure of the ANN based on the inputs, hidden layers and activation function.
3. Training of the ANN using the input pattern and desired output.
4. Validation of the iris model.

Each authorized user has a unique ANN–based model. The initial weights and biases of the MFN are randomly selected. The target of the MSE is set at first as 10^{-7} and log-sigmoid is used as the activation function for all layers. The ANN is trained using backpropagation learning algorithm with a learning constant of 0.05.

4.2 SVM based iris model

In developing user model based on iris code, a SVM with polynomial kernel function of order eight (8) is used. Each authorized user has his/her individual SVM–based model characterized by a set of support vectors. By using quadratic programming in the MATLAB environment, appropriate support vectors are determined. The penalty term C was set at 10^{15} to anticipate misclassified data.

4.3 Iris based authentication system

Figure 15 shows the flow chart of the proposed system. When trying to access the system the user is required to enter his/her ID to associate it with his/her iris image being acquired. The captured iris image is then processed and compared with the person model to verify his/her identity claim. The iris testing phase has a decision process in which the system decides whether the extracted features from the given iris image matches with the model of the claimed identity.
In order to give access to a user or reject an intruder, a threshold is set. If the degree of similarity between a given iris image and its model is greater than a given threshold, then the user will access the system, otherwise the user will be rejected.

5. Integrated biometric system

In order to reinforce the user authentication system, PBAS has been integrated with iris (keystroke and iris features) to improve the FAR and FRR rates. Figure 16 shows how the system is integrated.
The algorithm for the system is similar to what was discussed earlier. In the validation mode, a user is given a prompt to enter his/her ID together with password and iris image to validate whether there is an existing profile in the database or not. After the user enters the

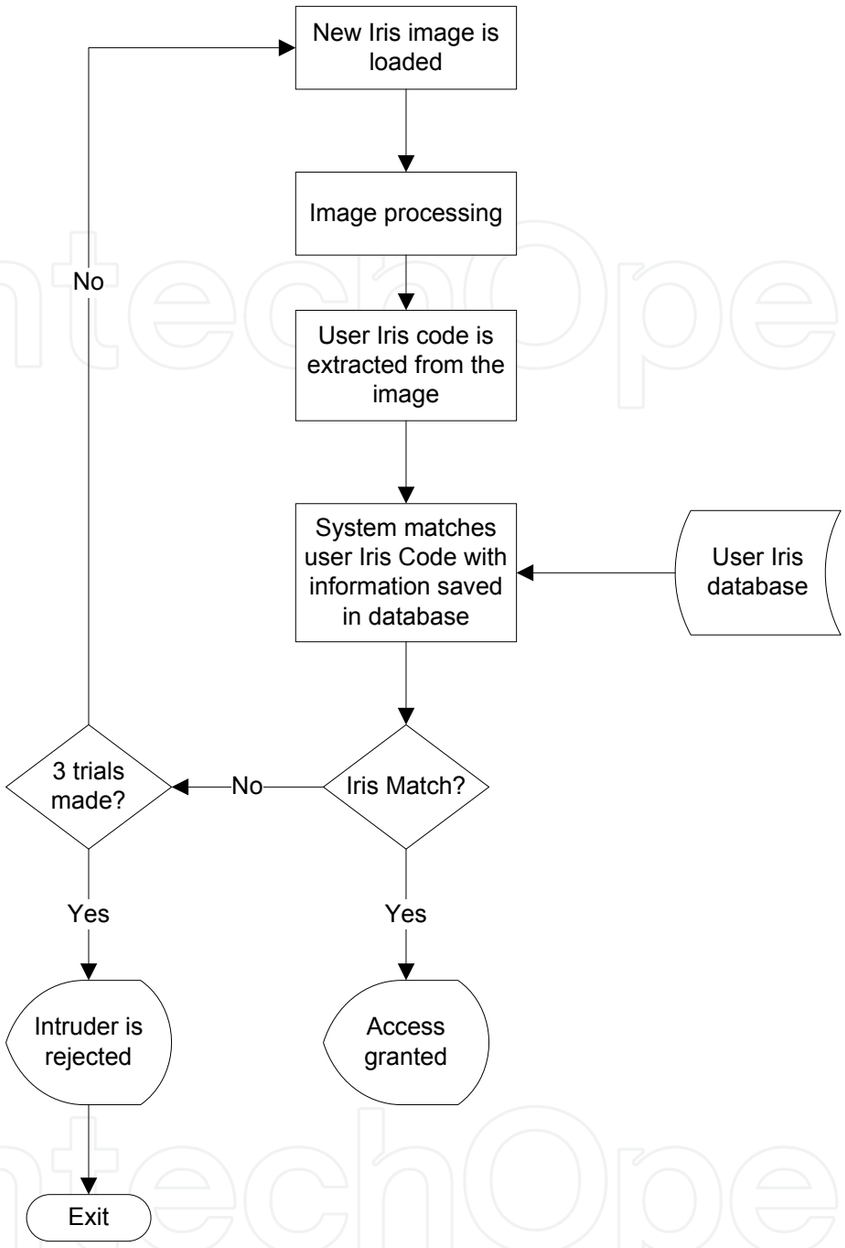

Fig. 15. Flowchart of iris-based authentication system.

password, maximum pressure and latency would be calculated and transferred to the classifiers (ANN, ANFIS and SVM). At the same time, an iris code is extracted from acquired image and transferred to the classifiers (ANN and SVM). The classifiers use MATLAB based algorithms to process the maximum pressure, latency and iris code and verify its compatibility (or level of matching) with the model in the database. If both keystroke template and iris of the user are nearly identical with the classifier models, then the system grants access to the user, otherwise access is denied after three (3) trials.

An experiment has been conducted to evaluate the effectiveness of the proposed system. The databases containing dynamic keystrokes and iris images were used in this experiment.

The dynamic keystroke data were collected from PBAS and stored in database 1. On the other hand, database 2 contains the iris images acquired from the Chinese Academy of Sciences–Institute of Automation (CASIA) eye image database.
A group of seven (7) persons was used for this experiment. Five (5) persons were considered as authorized users while the other two users were considered to be impostors.

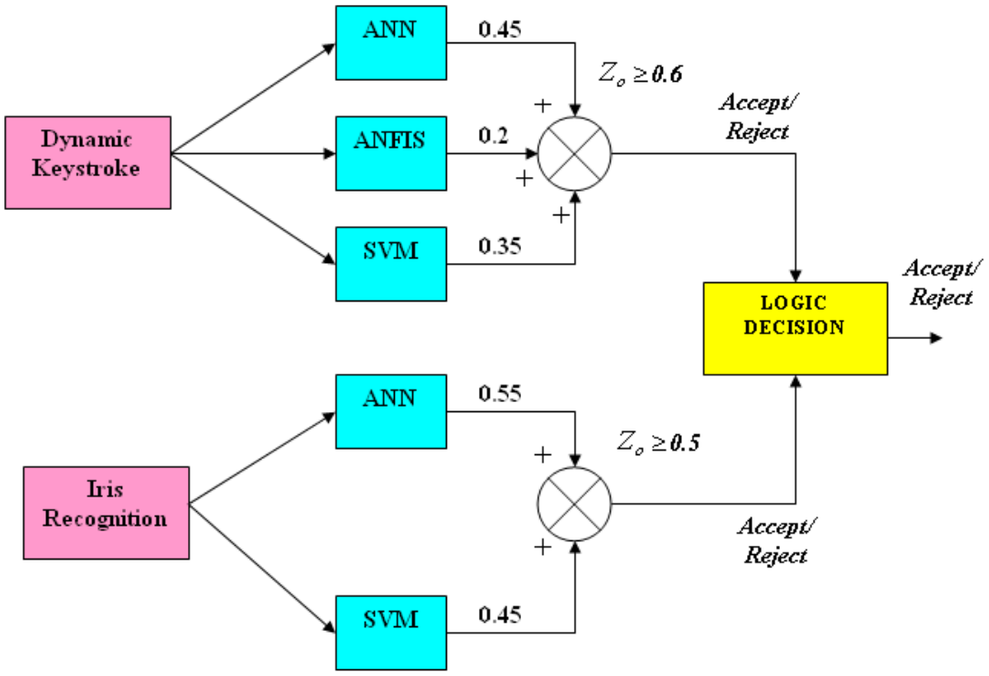

Fig. 16. Architecture proposed on integrating biometric system: first trial.

5.1 Training and testing performances of the integrated biometric system

In the first trial experiment, shown in Figure 16, the weight coefficients are assigned to each classifier experimentally by applying nonlinear programming. The ANN, ANFIS and SVM classifiers operate in such a way that every weight coefficient will multiply the classifier output to produce the best single classifier output which can be expressed as

$$\alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 \geq Z_o \tag{21}$$

where

$$\sum_{i=1}^{3} \alpha_i = 1$$

The outputs of the classifiers are compared with their respective threshold values. If the result is greater than the threshold value, then the decision unit would accept the user, otherwise the user would be denied access. A logic AND operator is used for the final decision, thus, the system only gives access to the user when "accept" condition for both keystroke and iris recognition are met.
Table 12 and Table 13 illustrate the training and verification results for the users. The weight coefficients and thresholds for the system are shown in Figure 16. Results show that all users

had perfect classification rates and the average training time for the system was less than 15 seconds.

Table 13 shows excellent FAR results for both close and open set. From security point of view, one can say that the system is fully protected from impostor attacks. However, the FRR values are relatively high with an average of 39.6 which would not be suitable for robust systems.

In order to reduce the FRR values, a second experiment was conducted by changing the values of α_1 and α_3 to 0.4 and 0.4 respectively and the threshold value for dynamics keystroke was changed to 0.7. All other values remain unchanged as shown in Figure 16.

The training performance based on the second trial is represented in Table 14. An average training time of less than 12 seconds is obtained in this experiment. Furthermore, there is no error in classifying the users as the results exhibit perfect classification for all people.

Authorized User	Training Time(sec)	Classification Rate (%)
User 1	9.250	100
User 2	7.781	100
User 3	13.07	100
User 4	17.00	100
User 5	22.48	100
Average	13.92	100

Table 12. Training performance based on the first trial.

Authorized User	FRR (%)	FAR (%) Close Set	FAR (%) Open Set
User 1	33	0	0
User 2	66	0	0
User 3	66	0	0
User 4	33	0	0
User 5	0	0	0
Average	39.6	0	0

Table 13. Testing performance based on the first trial.

Authorized User	Training Time(sec)	Classification Rate (%)
User 1	13.94	100
User 2	4.625	100
User 3	8.906	100
User 4	12.56	100
User 5	19.57	100
Average	11.92	100

Table 14. Training performance based on the second trial.

Furthermore, Table 15 shows that FAR results are excellent for both close and open set. The FRR average value has been reduced to 19.8; nevertheless, it is still relatively high and should be reduced further.

Authorized User	FRR (%)	FAR (%) Close Set	FAR (%) Open Set
User 1	33	0	0
User 2	66	0	0
User 3	0	0	0
User 4	0	0	0
User 5	0	0	0
Average	19.8	0	0

Table 15. Testing performance based on the second trial.

In the third trial all the weight coefficients were fixed, while the threshold value for iris recognition was changed to 0.45 whilst keeping other coefficients used in the second experiment constant.
Table 16 shows the perfect classification for all users in the third trial experiment. The proposed system produced no error in the classification of the individual users and its training time was about 17 seconds.

Authorized User	Training Time(sec)	Classification Rate (%)
User 1	20.45	100
User 2	15.62	100
User 3	19.44	100
User 4	21.42	100
User 5	7.718	100
Average	16.93	100

Table 16. Training performance based on third trial.

This experiment produces the best performance as illustrated in Table 17. The proposed integrated biometric system has been able to produce the best results for both FRR and FAR.

Authorized User	FRR (%)	FAR (%) Close Set	FAR (%) Open Set
User 1	0	0	0
User 2	0	0	0
User 3	0	0	0
User 4	0	0	0
User 5	0	0	0
Average	0	0	0

Table 17. Testing performance based on third trial.

Consequently, one can conclude that with the proper choice of threshold values the performance of the system can be optimized to achieve the best FRR and FAR rates.
The performance of the proposed system, especially with larger number of participants and sample size is still under investigation.

6. Conclusion and recommendation

Several commercial keystroke biometric algorithms rely on the use of the time-information (latency) alone and thus utilize only one information aspect of the keystroke action. It neglects the force applied during the keystroke action. PBAS has successfully acquired both time-frame and applied force information to develop varied biometric systems. Experiments have proved that force is highly distinctive to users typing keystroke. While some users may have similar latency profiles, however their keystroke pressure templates are easily discriminated.
The combination of AR and latency classifiers allows for an increase in latency threshold value to decrease the FRR rates. This increase does not have great effect on the FAR rates as the AR classifier would reject intruders based on their pressure templates, but it will make the system more user friendly without compromising the security.
By utilizing maximum pressure for keystroke template it is possible to merge the force and latency as a single feature for users. Multiple classifiers consisting of ANN, ANFIS and SVM are employed to authenticate personal identity based on extracted features. The combined features (maximum pressure and latency) are considerably more effective in verifying the authorized users. Results of experiments have shown that the use of combined features are more effective as compared to that of individual feature such as maximum pressure or latency as it gives better FRR and FAR for both open and close set conditions.
A multimodal biometric system uses multiple applications to capture different types of biometrics. This allows the integration of two or more types of biometric recognition and verification systems in order to meet stringent performance requirements. Furthermore, multimodal biometric systems also provide anti-spoofing measures by making it difficult for impostor to spoof multiple biometrics simultaneously. In this chapter, an integrated biometric system, which integrates keystroke pressure–based typing biometric system and iris verification has been presented. The proposed integrated biometric system is designed to operate in parallel mode and the integration process is made at the decision level. Integrated biometric fusion at the decision-level has shown to have good flexibility, high level of robustness and a great potential for improved accuracy.

7. References

Akaike, H. (1970). Statistical predictor identification, *Annals of the Institute of Statistical Mathematics*, Vol. 22, No. 2, (1970) pp. 203-217, ISSN 0020-3157.

Araujo, L.C.F.; Sucupira, L.H.R.; Lizarraga, M.G.; Ling, L.L. &Yabu-Uti, J.B.T. (2005). User Authentication through Typing Biometrics Features. *IEEE Transactions on Signal Processing*. Vol. 53, No. 2, (February 2005), pp. 851 – 855, ISSN1053-587X.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery,* Vol. 2, No. 2, (1998), pp. 121-167, ISSN 13845810.

Daugman, J. (2003). The importance of being random: statistical principles of iris recognition. *Pattern Recognition, Vol. 3, No.* 2, (February 2003), pp. 279-291, ISSN 00313203.

De Ru, W. & Eloff, J. (1997). Enhanced Password Authentication through Fuzzy Logic, *Intelligent Systems and Their Applications,* Vol. 12, No. 6, (November 1997), pp.38-45, ISNN 0885-9000.

Eltahir, W.S.; Salami, M.J.E.; Ismail A.F. & Lai, W.K. (2008). Design and Evaluation of a Pressure-Based Typing Biometric Authentication System, EURASIP *Journal on Information Security*, Vol. 2008, No. 1, (2008), doi:10.1155/2008/345047,Available from http://www.hindawi.com/journals/is/2008/345047/cta/

Eltahir, W.S.; Salami, M.J.E.; Ismail A.F. & Lai, W.K. (2004). Dynamic Keystroke Analysis Using AR Model, *Proceedings of the International Conference on Industrial Technology (IEEE-ICIT04),* pp. 1555 – 1560, ISBN 0-7803-8662-0, Hammamet, Tunisia, 08-10 December, 2004.

Eltahir, W.E.; Lai, W.K.; Salami, M.J.E. & Ismail, A.F. (2003). Hardware Design, Development and Evaluation of a Pressure-based typing Biometric Authentication System. *Proceedings of the Eighth Australian and New Zealand Intelligent Information Systems Conference(ANZIIS)*, pp. 49-54, *ISBN*1-74107-043-0, Sydney, Australia, December 10-12, 2003.

Jang, J. S. R., Sun, C. T. & Mizutani, E. (1996). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence.* Prentice Hall, ISBN 978-0132610667, USA.

Joyce, R. & Gupta, G. (1990). Identity Authentication Based On Keystroke Latencies. *Communications of the ACM*, Vol. 33, No. 2, (1990), pp. 168-176, ISSN 0001-0782

Lin, C. T. & George, C. S. (1995).*Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems.* Prentice-Hall, ISBN 978-0132351690, USA.

Manson, H. H. (1996). *Statistical Digital Signal Processing and Modelling*. John Wiley & Sons, ISBN 0-471-59431-8, USA.

Masek, L. (2003). Recognition of Human Iris Patterns for Biometric Identification, In: *Measurement*, 01.11.2009, Available from
http://www.csse.uwa.edu.au/~pk/studentprojects/libor/index.html.

Monrose, F. & Rubin, A. (1997).Authentication via keystroke dynamics. *Proceedings of 4th ACM Conference on Computer and Communication security*, pp. 48-56, ISBN 0-89791-912-2, Zurich, Switzerland, April 01 - 04, 1997.

Ord, T. & Furnell, S. (2000). User Authentication for Keypad-Based Devices Using Keystroke Analysis. *Proceedings of the Second International Network Conference (INC)*: pp. 263-272, ISBN 1 84102 066 4, Plymouth, UK, 03-06 July, 2000.

Shiavi, R. (1991). *Introduction to Applied Statistical Signal Analysis*. Aksen associates incorporated publishers, ISBN0256088624, Homewood, Illinois, USA.

Wildes, R.P. (1997). Iris Recognition: An Emerging Biometric Technology. *Proceedings of the IEEE,* Vol. 85, No. 9, (August 2002), pp. 1348-1363, ISSN 0018-9219.

Biometric Systems, Design and Applications

Edited by Mr Zahid Riaz

ISBN 978-953-307-542-6
Hard cover, 262 pages
Publisher InTech
Published online 21, October, 2011
Published in print edition October, 2011

Biometric authentication has been widely used for access control and security systems over the past few years. The purpose of this book is to provide the readers with life cycle of different biometric authentication systems from their design and development to qualification and final application. The major systems discussed in this book include fingerprint identification, face recognition, iris segmentation and classification, signature verification and other miscellaneous systems which describe management policies of biometrics, reliability measures, pressure based typing and signature verification, bio-chemical systems and behavioral characteristics. In summary, this book provides the students and the researchers with different approaches to develop biometric authentication systems and at the same time includes state-of-the-art approaches in their design and development. The approaches have been thoroughly tested on standard databases and in real world applications.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Momoh J. E. Salami, Wasil Eltahir and Hashimah Ali (2011). Design and Evaluation of a Pressure Based Typing Biometric Authentication System, Biometric Systems, Design and Applications, Mr Zahid Riaz (Ed.), ISBN: 978-953-307-542-6, InTech, Available from: http://www.intechopen.com/books/biometric-systems-design-and-applications/design-and-evaluation-of-a-pressure-based-typing-biometric-authentication-system

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen