Pressure-Based Typing Biometrics User Authentication Using the Fuzzy ARTMAP Neural Network

Chen Change LOY
Adv. Informatics Research Group
MIMOS Berhad
57000 Kuala Lumpur, Malaysia
chenchange.loy@mimos.my

Chee Peng LIM
Sch. of Electrical & Electronic Eng.
University of Science Malaysia, Eng. Campus
14300 Nibong Tebal, Penang, Malaysia
cplim@eng.usm.my

Weng Kin LAI
Adv. Informatics Research Group
MIMOS Berhad
57000 Kuala Lumpur, Malaysia
lai@mimos.my

Abstract—**In spite of the popularity of password-based authentication method, there are many inherent flaws with this approach. To minimize the risk of intrusion, typing biometrics can be used to complement and strengthen this popular authentication method. In this paper we investigate how keystroke pressure is used to strengthen the security of traditional password-based authentication system, and compare its performance with that of the conventional timing-based keystroke technique. The paper also investigates the use of combined keystroke pressure and latency for the verification process. The performances of several classification methods in user authentication, namely Multilayer Perceptron (MLP), Logistic Regression (LR), Fuzzy ARTMAP (FAM) neural networks, and a statistical approach, are studied and compared. Although keystroke latency gives better results than keystroke pressure, a combination of both techniques yields the best performance, with the False Acceptance Rate (FAR) of 0.87% and the False Rejection Rate (FRR) of 4.4%.**

I. INTRODUCTION

Computer systems are now used in almost all aspects of our life. Personal computers have evolved from single-user systems to multi-user networks spanning national and international territories, and even expanding to advanced grid network with large scale sharing of computer resources, such as computational power and databases. Hence, the increasing degree to which confidential and proprietary data can be stolen makes security a foremost concern in today's age of technology.

Password mechanisms have been, and probably will remain, as the primary method of user authentication in web-based or traditional computer access terminals. Ironically, password authentication is an inexpensive, convenient and familiar paradigm that most operating systems support. Unfortunately, static identification and authentication seems to be inadequate to protect computer resources from malicious attacks and intrusions, since passwords can be easily cracked, guessed, stolen or deliberately shared. To remedy these potential security weaknesses, more robust safeguards or strategies are needed against unauthorized access to computer resources.

Typing biometrics is one alternative biometric technology in the computer security arena. As the name implies, it is an automated biometric method that analyzes the way a person types on a keyboard. The concept of typing biometrics we are using here is basically adding another protection layer to the current password system. The premise behind this protection layer is that each person exhibits a distinctive pattern and cadence of typing. Therefore, unless the imposter has the ability to replicate or imitate exactly the authorized user's typing patterns, it would be very difficult for the imposter to gain full access to the computer resources, even if the imposter is able to guess the correct password.

Previous research [1–3] has shown that it is possible to identify a user based on his or her keystroke latency patterns with a high level of accuracy. In this paper we investigate how keystroke pressure (the amount of force exerted on each key pressed) may be used to reinforce the system's security. In order to measure the forces exerted during typing, we developed a pressure-sensitive keyboard system that allows us to acquire keystroke pressure [4].

Based on the data collected from keystroke pressure, we compare the performance among four different classification methods, namely statistical approach, Multilayer Perceptron (MLP), Logistic Regression (LR), and Fuzzy ARTMAP (FAM) neural networks. The results obtained are then compared with that of the conventional keystroke timing-based technique. Apart from that, we investigated the combination of keystroke pressure and keystroke latency in a single profile in order to achieve a higher accuracy rate.

The organization of this paper is as follows. First, an explanation of the procedures for setting up the experiment is provided in Part II. In Part III, the classification methods used in this paper are described. Next, the results are discussed in Part IV. Finally, the paper concludes with some recommendations for further investigation in Part V.

II. Experimental Set-up

A. Data Collection

A special keyboard system with pressure sensors adhered underneath the keys was used to monitor the typing patterns. Analogously, the pressure sensor acts like a variable resistor where its resistance changes in accordance to the amount of force that each user applies when he or she types. A force-to-voltage circuit was used to convert the keystroke pressure to discrete voltage-time signals. The signals were then sent to the computer through data acquisition hardware.

A total of 10 computer-literate users participated in the experiment. Note that the participants were not informed of the data collection strategy. Initially, each participant had to register his or her user name and password with the system. All participants were requested to enter the same password since the objective is to determine if one user could be identified and differentiated from the other, as well as from the whole group. The password was "*try4-mbs*". It was chosen because it is more than 8 characters in length combining symbols, numbers and letters. Participants have been asked to practise typing the password until they are familiar with it prior to the actual experiment several days beforehand. A new user went through a session where he or she typed the password for 15 times under conditions simulating the actual login environment.

Generally, a user's typing pattern will stabilize as his/her fingers become accustomed to the keys of the authentication string (consisting of the account name and password) being entered. The more times a user enters the same password, the less fuzzy the typing pattern becomes [1]. In order to obtain a more consistent typing pattern from the user, all data from the first five trials were eliminated. Only the next ten trials were used.

As a result, a database of 10 user profiles was constructed. Each user profile composed of 10 samples of keystroke latency as well as 10 samples of keystroke pressure. Keystroke latency was measured in milliseconds, whereas keystroke pressure was measured in volts ranging from 0 to 10 volts in the form of time discrete signal.

B. Keystroke Pressure Signal Transformation

Features based on frequency domain analysis play a significant role in this application. From preliminary observations, some user's keystroke pressure may be seen in some frequency band as a higher amplitude level, whereas in others, the amplitude level is higher in the some of the frequency bands.

In our study, the pressure discrete time signals were transformed into the frequency domain by using a Fast Fourier Transform (FFT). The resulting outputs contained both the magnitude and phase information. Only the magnitude information is used for subsequent feature extraction. Note that the pressure signal transformation and feature extraction described in the next section are not applicable to the statistical approach.

C. Feature Extraction

Each of the typing patterns to be classified should have certain features that are unique in order to achieve satisfactory classification accuracy. This plays a very important role in accurately identifying the legitimate user by the classifier. Badly implemented feature extraction or improper features selected will probably lead to poor classification results even by using the best possible classifier. After some careful examination, features to be extracted from the frequency domain signal of keystroke pressure are determined and they are shown in Table I.

TABLE I. Features Extracted From The Frequency Domain Signal Of Keystroke Pressure

No.	Name of Feature	No.	Name of Feature
1	Arithmetic mean	6	Fundamental frequency
2	Root mean square	7	Energy
3	Peak	8	Kurtosis
4	Signal in noise & distortion	9	Skewness
5	Total harmonic distortion		

D. Data Pre-processing

In order to reduce the influence of outliers which are not representative of the user's typing pattern, the data for each profile was pre-processed [1, 2]. Firstly, for both keystroke pressure and latency, the mean and standard deviations of each feature in the profile were computed. Next, each feature was then compared with its respective mean and any measurements that differ by more than T standard deviations from the mean (i.e. outliers), would be discarded and replaced with the mean value instead. The threshold is presently defined as the mean plus 1.5 standard deviations. All features were rescaled so that they fall within the range of 0 and 1, as inputs to both the MLP and FAM need to be in within that range.

III. Classification Methods

A. Statistical Approach

The main purpose of using a simple statistical approach is to provide quick insights into the accuracy that can be obtained. Basically, we have adopted a total of 10 features that can be extracted from the keystroke pressure time discrete signal, i.e. the maximum value, minimum value, range, median, mean, sum, standard deviation, variance, skewness and kurtosis [5]. A reference sample is computed by averaging the features across the samples for each profile. The measurement of similarity is based on the difference between the test sample and the reference sample. A feature from the test sample is considered a good match with the feature from reference sample if it is within a set threshold of the reference feature, i.e. a test feature is valid if,

$$\left|\frac{\textit{reference feature} - \textit{test feature}}{\textit{reference feature}}\right| \times 100\% \leq \textit{threshold} \quad (1)$$

Thus, an attempt is considered valid if more than a certain amount of features in the test sample matches that from the reference template. The matching criterion is presently set at 8, i.e. the user can only be accepted as legitimate user if at least eight features from the test sample match those from the reference sample. Otherwise it is considered as an invalid attempt.

B. *Multilayer Perceptron*

The Multilayer Perceptron (MLP) [6] which is commonly known as the feed-forward neural network, is capable of separating both linearly and non-linearly separable pattern classes.

The network is trained with the backpropagation algorithm. The network used here comprises of nine input nodes, a single hidden layer with five hidden nodes and one output node. After several trials, the hidden unit is set to a *tan-sigmoid* non-linear activation function with a learning rate of 0.1 and momentum of 0.9.

C. *Logistic Regression*

The logit model was introduced by Joseph Berkson in 1944 [7], who also coined the term. The LR is a non-linear transformation of the linear regression. It is useful when the dependent or response variable is restricted to two values, which usually represent the occurrence or non-occurrence of some outcome event - usually coded as 0 or 1, respectively. Given $y = \alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_i x_i$, α is the constant of the equation, and β is the coefficient of the predictor variables, the relationship between the predictor and response variables is defined as

$$p = \frac{1}{1 + e^{-y}} \tag{2}$$

p is used to predict whether the user is legitimate or an imposter. The independent or predictor variables are extracted features of the users' typing patterns. If the value of *p* is less than a pre-defined threshold, the user is legitimate; otherwise the user will be rejected. At present, the threshold value is defined at 0.5.

D. *Fuzzy ARTMAP*

Fig. 1 shows the Fuzzy ARTMAP (FAM) neural network [8]. FAM includes a pair of Adaptive Resonance Theory (ART) modules designated as ART$_a$ and ART$_b$, which create stable recognition categories in response to arbitrary sequences of input patterns. FAM also includes a map field module, F^{ab}, an associative learning network to establish an association between input patterns and target classes. FAM, a generalization of binary ARTMAP, learns to classify inputs by a pattern of fuzzy membership values between 0 and 1 indicating the extent to which each feature is present. This generalization is accomplished by replacing the ART1 modules of the binary ARTMAP system with fuzzy ART modules.

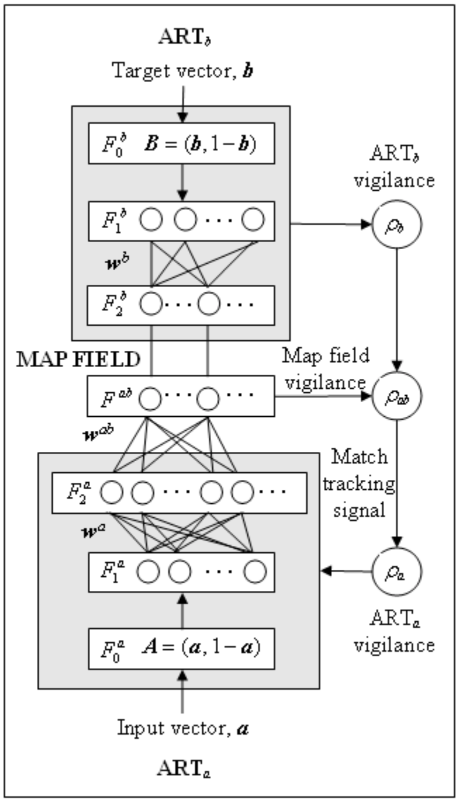

Figure 1. Architecture of Fuzzy ARTMAP (FAM) neural network.

There are two key parameters which influence the FAM performance in pattern classification, i.e. baseline vigilance parameter of ART$_a$, $\bar{\rho}_a \in [0, 1]$; and the learning rate parameter of ART$_a$, $\beta_a \in [0, 1]$. The vigilance parameter, $\rho_a \in [0, 1]$, is a dimensionless parameter of ART$_a$, that determines the level of similarity between the transformed prototype vector and the input vector required before a match is said to occur. At the beginning of each input presentation ART$_a$ vigilance, ρ_a equals a user-defined baseline vigilance parameter $\bar{\rho}_a$. The learning parameter, β_a, determines the learning modes of the network. There are two learning modes: fast learning ($\beta_a = 1$ for all times) and fast-commit slow recode learning ($\beta_a = 1$ for an uncommitted node and $\beta_a < 1$ for a committed node).

A series of experiments have been carried out to find the optimum value of $\bar{\rho}_a$ and β_a. It turned out that $\bar{\rho}_a = 0.77$ and $\beta_a = 0.01$ gave the best performance [4]. Thus, these values were used in the following experiments, unless stated otherwise.

The following is a typical operation in ART$_a$, which also occurs in ART$_b$. All the equations are applicable to ART$_b$ as well. Note that the following is a concise account on FAM, and interested readers can refer to [8] for further details of FAM. Initially, the original input vector $\boldsymbol{a}$ with *M*-dimensional is normalized into a *2M*-dimensional vector, $\boldsymbol{A}$ using a technique called *complement coding*.

$$A = \left(a, a^c\right) \equiv \left(a^1, \ldots, a^M, 1 - a^1, \ldots, 1 - a^M\right) \qquad (3)$$

The complement-coded input pattern $\boldsymbol{A}$ is propagated from F_1^a to F_2^a through a set of adaptive weights. For each input $\boldsymbol{A}$ and j^{th} F_2^a node, the choice function $T_j(\boldsymbol{A})$ is defined as in (4), with $\boldsymbol{w}_j^a$ denoting the weight vector of the j^{th} F_2^a node.

$$T_j(\boldsymbol{A}) = \frac{|\boldsymbol{A} \wedge \mathbf{w}_j^a|}{\alpha_a + |\mathbf{w}_j^a|} \qquad (4)$$

The node with the highest response value, denoted as node J, is selected as the winning node, with N denoting the number of F_2^a nodes.

$$T_j\left(\boldsymbol{A}\right) = \max\left\{T_j\left(\boldsymbol{A}\right) : 1, \ldots, N\right\} \qquad (5)$$

If there is more than one maximum $T_j(\boldsymbol{A})$ exists, the network chooses the category with the smallest index. At the same time, all other nodes $j \neq J$ are deactivated in accordance with the *winner-take-all* strategy. The winning node J then propagates its output back to F_1^a. Resonance occurs if the match function of the chosen category meets the vigilance criterion:

$$\frac{\left|\boldsymbol{A} \wedge \boldsymbol{w}_J^a\right|}{\left|\boldsymbol{A}\right|} \geq \rho_a \qquad (6)$$

If the vigilance test is satisfied, resonance is said to have occurred and the operation proceeds to the learning stage. If the test fails, node J is inhibited from participating in subsequent competitions. The network will carry out a new cycle of hypothesis selection and test until a match is found, - or until all the F_2^a nodes have been exhausted. If none of the existing nodes satisfies the vigilance test, then a new node is formed at F_2^a to code the input pattern.

After resonance has occurred, the map field F^{ab} receives input from both of the ART$_a$ or ART$_b$ category fields. Suppose that $\boldsymbol{y}^b$ is the output of ART$_b$, and $\boldsymbol{w}_J^{ab}$ is the weight vector that links node J in the ART$_a$ and the map field, the F^{ab} output vector $\boldsymbol{x}^{ab}$ is defined as

$$\boldsymbol{x}^{ab} = \boldsymbol{y}^b \wedge \boldsymbol{w}_J^{ab} \qquad (7)$$

After the output of the map field is calculated, a map field vigilance test is conducted to confirm the prediction by comparing the similarity between the map field output and the target vector using (1) with $\rho_{ab} \in [0, 1]$ denoting the user-defined map field vigilance parameter.

$$\frac{\left|\boldsymbol{x}^{ab}\right|}{\left|\boldsymbol{y}^b\right|} = \frac{\left|\boldsymbol{y}^b \wedge \boldsymbol{w}_J^{ab}\right|}{\left|\boldsymbol{y}^b\right|} \geq \rho_{ab} \qquad (8)$$

Failure of map field vigilance test indicates that the winning node, J of ART$_a$ incorrectly predicts the target class, a match tracking procedure is triggered to raise the ART$_a$ vigilance just enough to trigger a search for a new F_2^a coding node.

Since ART$_a$ fails to meet the matching criterion, the current F_2^a winning node is inhibited and the search for another F_2^a node begins. If all the F_2^a nodes in ART$_a$ are inhibited, a new node is created to code the input pattern. A node becomes committed after it is selected for coding. The adaptive weights of winning node in ART$_a$ is then updated according to,

$$\boldsymbol{w}_J^{a(new)} = \beta_a\left(\boldsymbol{A} \wedge \boldsymbol{w}_J^{a(old)}\right) + \left(1 - \beta_a\right)\boldsymbol{w}_J^{a(old)} \qquad (9)$$

Two different operating strategies of FAM were adopted in this paper. The strategies were used to form two types of FAM, namely, voting FAM and average FAM [9].

1) Voting FAM

The voting strategy is introduced based on the observation that different orderings or sequences of training samples will generate different cluster prototypes in ART$_a$. This would subsequently lead to different predictions of target classes, and thus different accuracy scores for each realization of FAM. In order to overcome this problem, a voting FAM network is constructed by training a pool of FAM, each with a random ordering of training samples. Each FAM is considered a voter in predicting an output class. Then, all predictions are combined and the final prediction for the given test pattern is the one with the majority number of votes. In other words, the final decision is the prediction made by more than half of the classifiers. Odd number of classifiers was used so that a final decision could be reached.

2) Average FAM

This method averages the performance metric from a pool of FAM network trained by different sequences of training samples.

IV. Results And Discussion

A comparison of experimental results among statistical approach, MLP, LR and FAM is presented. The primary objective of the experiments is to examine the validity of using keystroke pressure in user authentication. Thus, the experiments and discussions are mainly focused on keystroke pressure. In order to evaluate the performance of different classification approaches, the following performance metrics were used.

- **Accuracy** – the ratio of the number of correct prediction to the total number of cases.

- **Sensitivity** – the ratio of the number of correct legitimate user's samples prediction to the total number of legitimate user's samples.
- **Specificity** – the ratio of the number of correct imposters' samples prediction to the total number of imposters' samples.
- **False Acceptance Rate** (*FAR*) – the rate that an imposter's typing pattern is falsely identified as belonging to a legitimate user,

$$FAR = 100 - Specificity \tag{10}$$

- **False Rejection Rate** (*FRR*) – the rate that a typing pattern is incorrectly identified as belonging to an imposter,

$$FRR = 100 - Sensitivity \tag{11 }$$

A. Performance Evaluation with Cross Validation

The approach of 10-fold cross validation was employed in this investigation [10]. Initially, all keystroke pressure data were divided into 10 profiles where each profile represented a user. Each time a different profile was chosen to be a legitimate user, it was labeled as "Class 0". The rest (9 profiles) were then assumed to be imposters and all of them were labeled as "Class 1" instead. As a result, there were a total of 10 input sequences, in which each input sequence has a different legitimate user profile. For each input sequence, it was randomly partitioned into 90%/10% of training/testing samples. The partitioning procedure was carried out until 100 pairs of training/testing data set have been generated.

Note that for the statistical approach, a slightly different method had been used to generate the training/testing data set. For each input sequence, there were only 10 pairs of training/testing data set. Every time we picked a different test sample from each of the profiles, it was compared against the reference (training) sample computed from Class 0. A reference sample is computed based on 9 samples exclusive to the test sample.

Table II summarizes the experimental results in terms of accuracy, sensitivity, and specificity for different classification methods using the keystroke pressure data.

TABLE II. PERFORMANCE COMPARISON

	Accuracy	Sensitivity	Specificity
Statistical Approach[a]	78.10 ± 4.53 %	31.00 ± 33.48 %	83.33 ± 4.16 %
MLP	89.70 ± 5.86 %	53.60 ± 27.43 %	93.71 ± 3.89 %
LR	94.22 ± 3.79 %	77.50 ± 17.21 %	96.08 ± 2.56 %
Average FAM	93.41 ± 3.68 %	76.20 ± 18.16 %	95.32 ± 2.64 %

a. *Best result with threshold value = 10*

For the statistical approach, the FAR of 16.67% is clearly too high since a reliable identity verifier should have a FAR that is less than 1%. Furthermore, the FRR of 61% is also not acceptable since the users may have to retry many times in order to log onto the computer.

For the MLP, both the FAR (6.29%) and FRR (46.4%) are not suitable for practical application. LR yielded better results as compared to MLP, with FAR of 3.92% and FRR of 22.5%. However, the FAR is still considered high from the target FAR of less than 1%. Similarly, the results achieved for both FAR of 4.68% and FRR of 23.8% for the average FAM is not satisfactory too.

In the experiment using the statistical approach, we also examined the error rate by varying the threshold value. Fig. 2 shows the relationship between the threshold value and the FAR and FRR. As may be seen from these graphs, changing the threshold value decreases one type of error rate while increasing the other type of error. The equal error rate where these two error curves intersect is approximately 40% with a threshold value of 19.

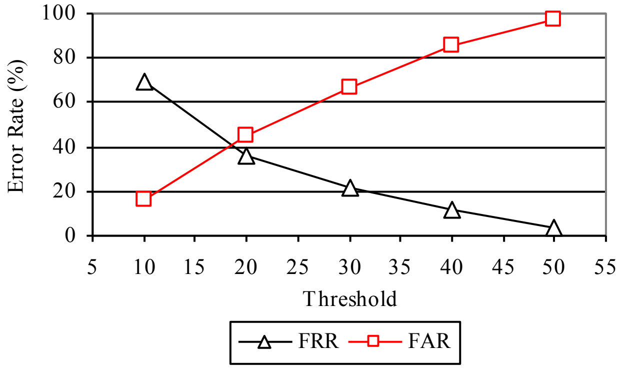

Figure 2. Relationship between threshold value with FAR and FRR.

B. Voting Strategy for FAM

The objective of this experiment is to compare the accuracy of LR, average FAM and voting FAM. In this experiment, nineteen independent classifiers were employed, each with randomized input patterns. Each classifier yielded a set of predictive outcomes, and these were combined to reach a final decision using majority voting. For this experiment, $\overline{\rho}_a$ = 0.1 and β_a = 0.01. Fig. 3 shows the results averaged over twenty runs.

As can be seen from Fig. 3, voting results generally outperform average results and LR results. This suggests that voting may be a good approach to improve classification results.

From this experiment using voting FAM, the highest accuracy achieved was 93.90%. The performance improved as the number of voters increased from just three voters before saturating to 93.80% when this was increased beyond seven voters.

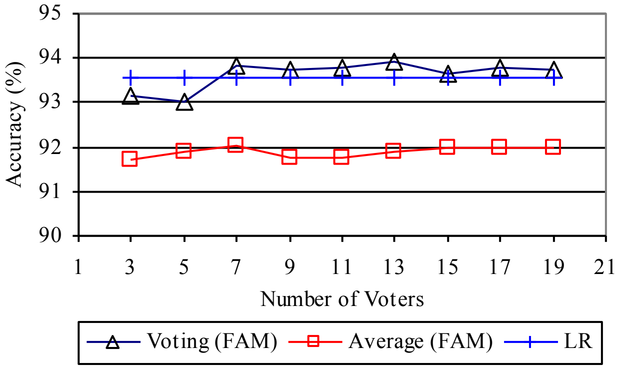

Figure 3. Comparison of performance among LR, average FAM and voting FAM.

C. *Comparison of Keystroke Pressure and Latency*

In this experiment, the main goal is to compare the FAM performance by using keystroke pressure and latency. We also examined the use of both keystroke pressure and keystroke latency as a single profile with the hope that this may give us a better FAM classification accuracy rate. Average FAM was used in this experiment. Table III shows the results in terms of accuracy, sensitivity, and specificity. Better results from keystroke latency as compared with keystroke pressure are obtained. However, the best results are achieved by combining both keystroke pressure and latency as a single profile.

TABLE III. PERFORMANCE USING PLURALITY OF FEATURES

Features	Accuracy	Sensitivity	Specificity
Pressure	93.41 ± 3.68 %	76.20 ± 18.16 %	95.32 ± 2.64 %
Latency	96.17 ± 3.81 %	85.40 ± 15.56 %	97.37 ± 2.83 %
Pressure + Latency	98.78 ± 1.42 %	95.60 ± 5.42 %	99.13 ± 1.18 %

V. CONCLUSIONS AND RECOMMENDATIONS

Instead of using conventional timing-based typing characteristics, the work presented in this paper investigates the applicability of using a relatively novel method to ascertain the identity of a user–the user's typing pressure. A series of experiments have been systematically conducted using different classification algorithms. Another focus of the paper is to explore the use of a voting strategy to improve the performance of FAM. The results were compared with those obtained from experiments using latency patterns. From the experiments, the poor results obtained from the statistical approach may not be surprising, based on its simplicity and ease of use. Nevertheless, it does give us an indication of the baseline accuracy rate that can be expected. In general, performance of MLP was relatively poor, while the performances of LR and FAM were comparably better. However, voting FAM yielded the best performance when compared with either average FAM or LR. In summary, the combination of keystroke latency and pressure yielded the best result, i.e., FAR of 0.87% and FRR of 4.4%, obtained by using average FAM.

The work presented in this paper has revealed the potential of the pressure-based typing biometrics system. However, there are still a number of areas that can be enhanced and pursued as further work. Firstly, the number data samples obtained should be extended to include more participants. Validation and verification work to vindicate the system further with a larger sample size is ongoing. Secondly, the robustness of biometrics refers to the extent to which the characteristic or trait is subject to significant changes over time. Experiments should be conducted to examine the robustness of keystroke dynamics by collecting the user's typing patterns at different phases, where each phase can be a few weeks or a few days apart. In addition to password, typing patterns displayed on entering username may be used as a means to identify a user because the username is also a regularly typed string. Effectively, both password and username can be combined to improve the recognition accuracy. Finally, keystroke duration is the length of time a key is depressed. Obaidat et al. [3] have shown that keystroke durations yielded better classification accuracy than keystroke latencies. It would be interesting to investigate the combination of keystroke duration, latency, and pressure for developing a more accurate typing biometrics-based user authentication system.

ACKNOWLEDGMENT

The authors would like to thank Miss Hanum bte Ibrahim for her efforts in developing the program for the statistical approach experiment.

REFERENCES

[1] R. Joyce, and G. Gupta, "Identity authentication based on keystroke latencies," Comm. ACM, vol. 33, pp. 168–176, February 1990.

[2] F. Monrose, and A. Rubin, "Authentication via keystroke dynamics," Proc. of the Fourth ACM Conference on Computer and Communications Security, Zurich, Switzerland, pp. 48–56, 1997.

[3] M. S. Obaidat, B. Sadoun, "Verification of computer users using keystroke dynamics," IEEE Trans. Syst. Man, and Cybernet., vol. 27, pp. 261–269, 1997.

[4] C. C. Loy, "Development of a pressure-based typing biometrics system for user authentication," Engineering Thesis, USM, 2005.

[5] Milton, Ed. Abramowitz, Handbook of Mathematical Functions, Dover Publications, 1974.

[6] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Parallel Distributed Processing, vol. 1 and 2, Cambridge, MA, The MIT Press, 1986.

[7] J. Berkson, "Application of the logistic function to bio-assay," Journal of the American Statistical Association, vol. 39, pp. 357–365, 1944.

[8] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analogue multidimensional maps," IEEE Trans. on Neural Networks, vol. 3, pp. 698–713, 1992.

[9] C. P. Lim, M. M. Kuan, and R. F. Harrison, "On operating strategies of the Fuzzy ARTMAP neural network: a comparative study," International Journal of Computational Intelligence and Applications, vol. 3, pp. 23–43, 2003.

[10] M. Stone, "Cross-validatory choice and assessment of statistical prediction," Journal of the Royal Statistical Society, vol. 36, pp. 111–147, 1974.