Feature Fusion Approach on Keystroke Dynamics Efficiency Enhancement

Pin Shen Teh[1], Shigang Yue[2]
School of Computer Science
University of Lincoln
Lincoln, United Kingdom
pteh@lincoln.ac.uk[1], syue@lincoln.ac.uk[2]

Andrew B.J. Teoh[3]
School of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
bjteoh@yonsei.ac.kr[3]

ABSTRACT

In this paper we study the performance and effect of diverse keystroke feature combinations on keystroke dynamics authentication system by using fusion approach. First of all, four types of keystroke features are acquired from our collected dataset, later then transformed into similarity scores by using Gaussian Probability Density Function (GPD) and Direction Similarity Measure (DSM). Next, three fusion approaches are introduced to merge the scores pairing with different combinations of fusion rules. Result shows that the finest performance is obtained by the combination of both dwell time and flight time collectively. Finally, this experiment also investigates the effect of using larger dataset on recognition performance, which turns out to be rather consistent.

KEYWORDS

Keystroke Dynamics; Keystroke Feature; Security; Authentication; Biometrics; Fusion

1 OVERVIEW

Keystroke dynamics biometrics is a data processing technique that analyzes the way a user types by monitoring the keyboard inputs in attempt to identify them by their habitual typing patterns [1]. As compared to other physical and behavioral biometrics, keystroke dynamics biometrics falls short to be a sole biometrics authenticator. Conversely, by integrating keystroke dynamics biometrics into the existing password authentication system, even if the impostor is able to present the correct login information, either by hacking, key logger or shoulder spoofing, without the right typing pattern, they will be denied access. In contrast, sole password authentication will guarantee access to any user as long as the login credential received is correct not considering if the user is legitimate.

1.1 Background Study

Among the earliest researches on keystroke dynamics had been conducted by [2]. The experiment involved 6 professional secretaries as subjects. Subjects were required to type three passages consisting of 300 to 400 words each, separated by two sessions spanning across four months. The time between each pair of consecutive keystrokes were calculated and recorded from the experiment. Although the study was able to acquire good result of 0% False Acceptance Rate (FAR) and 4% False Rejection Rate (FRR) by using statistical t-test, it was impractical in real cases due to the massive amount of input required.

Joyce et al. [3] reported some encouraging result of 0.17% FAR and 13.3% FRR. Their experiment involved 33 users. The mean reference feature was computed from eight sets of the users' keystroke patterns consisted of username, password, first name, and last name. They then computed the norm of difference between the test keystroke feature and mean reference feature used for authentication. Meanwhile [4] employed fuzzy logic to measure 29 users' typing biometrics. Their experiment achieved a moderately low FAR and FRR of 2.79% and 7.379% respectively. All of the above mentioned researches engaged small number of experimental subjects. Therefore, the results doubtfully drew to a strong conclusion.

Most research works done by far were focusing on extracting keystroke timing latency as feature data and mainly focused on one or two types of keystroke features at a time. For example [5] proposed a simple statistical method in which duration of each key press and the time duration between each different key press were considered. The experimental result recorded an unfavorable FRR of 24%. The author asserted that the poor performance was partly caused by the poor typing skill of the users involved.

On the other hand, [6] implemented a probabilistic model used to characterize each user's password by means of continuous Hidden Markov Models (HMM), where else [7] introduced Gaussian Mixture Models (GMM) in keystroke identification task. The researchers argued that keystroke pattern was harder to duplicate as compared to written signature. This was because an intruder has limited number of trials, as most authentication systems will block further access if an erroneous verification attempt exceeds three times. A total of 8 subjects were enrolled into their system by typing their full names ten times. Keystroke duration and latency were extracted from the user samples. The experiment produced a good FRR of 2.4% and a FAR of 2.1%. The advantage of their method was the ability to update user template upon each successful authentication.

Monrose et al. [8] stressed that keystroke recognition based on fixed-text was more desirable than free-text. This was due to contributing factors such as uncontrolled environmental parameters, unconstrained inputs, and uncooperative user which imposed restriction on the usage of free-text recognition. The author used Euclidean distance and Bayesian alike classifier as the classification techniques in their study. The keystroke features extracted were keystroke duration and keystroke latency (time between a key is released and the next key is pressed). However, the performance result presented was not complete as the result only reported in FRR of 16.78% and 7.83% for Euclidean distance and Bayesian classifier respectively.

While most research works on keystroke dynamics have been conducted on conventional timing-based typing characteristics, [9] looked into the prospect of using typing pressure as keystroke feature. A conventional keyboard was customized into a pressure sensitive version by inserting special force detection sensors underneath the keyboard matrix. ARTMAP-FD neural network was used for keystroke pattern classification. They fed the network with keystroke pressure, keystroke latency, as well as the combination of both. The performances of the classifier using different sets of aforementioned keystroke features were recorded at an EER of 16.50%, 14.94%, and 11.78% respectively. Although the inclusion of keystroke pressure feature showed improvement in performance, the error rate was still higher than the other traditional keystroke features based methods. Furthermore the practicability of such customized keyboard in large scale implementation was called into question due to limited availability.

Another group of researchers who tried to exploit the pressure feature in human typing sequence for identity verification was [10]. In fact they combined global features of pressure sequences and dynamic time warping with traditional keystroke features in their investigation. The three methods produced individual scores which were then combined by using weighted sum rule to obtain a final score. Their experiment involved 100 users with 50 samples each. They were able to obtain an EER of 1.41% with the combination of pressure features. Nevertheless, the percentage of improvement after using pressure sensitive features was fairly insignificant as compared to the increase of cost for the pressure sensitive keyboard.

There were also attempts on using simple fusion method to increase the performance of keystroke recognition [11]. The authors studied on the possibility of combining three different methods. The normalized scores produced by each method were fused by using weighted sum rule into a final score. The performance of the experiment was recorded at an EER of approximately 5%. Although result improvement was remarkable after fusion, more comprehensive fusion approaches would be desired to validate these findings.

1.2 Motivation and Contribution

Our study focuses on static authentication based keystroke dynamics recognition system. Static authentication is favorable in strengthening the existing password authentication system, especially in desktop, web, and mobile applications. Existing works in the literature of keystroke dynamics mainly focus on one or two types of keystroke features at a given instance. Another observation is the lacking of information on which keystroke feature performs best. Specifically, this paper extends our previous study [12] in several ways. First, we investigate the performance difference of a larger data set of 100 subjects compared to 50 from the previous study. Next, it extends the study on fusion by introducing greater dimension in fusion approaches, instead of only weighted sum rule and sum rule. Finally, a full scale combination of keystroke features is tested and compared.

The main contributions of this work include:

i. The extraction of four different kinds of keystroke features and the study on which is most effective in keystroke dynamics domain.

ii. The study on the possibility of obtaining a better result by performing a full scale combination on the four keystroke features.

iii. To study the influence on the performance between small and large dataset.

2 METHODOLOGY

2.1 Feature Extraction

Keystroke features can be extracted in terms of:

- Dwell Time (DT) [13],[14],[15],[16],[17],[18]
- Flight Time (FT) [19],[20],[21],[22],[23],[24]
- Difficulties of typing phrase [4]
- Pressure of keystroke [25],[26],[27],[28],[29]
- Typing rate [30],[31],[32]
- Linguistic style [33]
- Sound of typing [34]
- Frequency of word errors [30],[14]

Nevertheless, not all of the above features are favorable. For example, in order to acquire keystroke pressure feature, dedicated pressure sensitive keyboard is essential, which contradicts with the main advantage of keystroke dynamics biometrics. Frequency of word errors, typing rate, and difficulties of typing phrase are merely practical for text with large number of characters. Where else, there is a high concern with the noise associates with the acquisition devices used to record sound of typing.

In this experiment, we extract *Dwell Time*, timing interval between keystroke actions of the same key (also known as duration, press or hold time) and *Flight Time*, timing interval between keystroke actions of different keys (also known as latency). Eventually, if we try to break down flight time further; we notice that it can be sub divided into three types (D_2, D_3, and D_4) as in Figure 1. Explanation and method of calculation for each of these keystroke features based on example are given as follow.

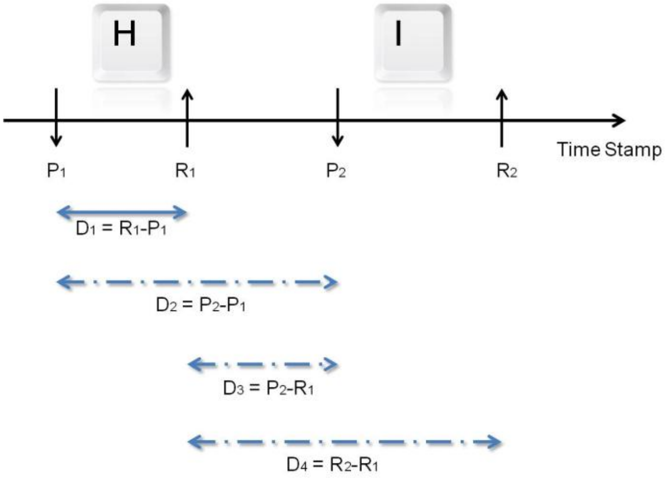

Figure 1. Example of four keystroke features extracted in this experiment.

Dwell Time (D_1): The time interval between a key pressed until the key is released.

$$D_1 = R_1 - P_1 \tag{1}$$

Flight Time (D_2): The time interval between a key press and the next key press.

$$D_2 = P_2 - P_1 \tag{2}$$

Flight Time (D_3): The time interval between a key release and the next key press. Negative value may occur if the next key is pressed before the previous key release.

$$D_3 = P_2 - R_1 \tag{3}$$

Flight Time (D_4): The time interval between a key release and the next key release.

$$D_4 = R_2 - R_1 \tag{4}$$

At this stage we propose two methods: (1) Gaussian Probability Density Function (GPD) and (2) Direction Similarity Measure (DSM). The outputs of both methods are both a similarity score, *S*, where $0 \le S \le 1$. The score is then compared against a predefined threshold, *thr*. If the score is larger than the threshold, then we declare the user as a genuine user, and vice versa.

$$Genuine = \begin{cases} yes & if \quad S > thr \\ no & otherwise \end{cases} \tag{5}$$

Since it is not the main objective of this paper to discuss about the classifier employed, we would redirect readers to [35], which provides comprehensive details and examples. However, a brief description of these two methods will be summarized in the following section.

2.2 Matching

2.2.1 Gaussian Probability Density Function

GPD is used to calculate the similarity score between a reference template and a claimant template. The nearer a score to the value of one indicates a higher probability that a claimant template belongs to a genuine user and vice versa. Generally GPD score has the form of

$$S_{GPD} = \frac{\sum_{i=1}^{k} -\left(\frac{(t_i - \mu_i)^2}{2\sigma_i^{\ 2}}\right)}{k} \tag{6}$$

where *t* is the timing latency of a particular character of the claimant, *k* is the total number of keystroke feature vector in a phrase, μ and σ are the mean and standard deviation of a reference template respectively.

2.2.2 Direction Similarity Measure

DSM is an uncomplicated yet discriminative approach to compare user keystroke typing patterns. The idea behind this method is to determine the consistency of the users' typing patterns. The formula of calculating DSM score is defined as follow

$$S_{DSM} = \frac{m}{c-1} \tag{7}$$

where *c* is the total number of characters in a phrase. Let Δd represents the dissimilarity of direction in two successive keystrokes. We observe the change of signs in Δd between a reference template and a test data template. If both signs are identical, we increase the counter *m*, and vice versa.

2.3 Fusion Approach

2.3.1 Single Layer Single Expert (SLSE)

In this category, only one matching function is used to find the matching score of two different combination types of keystroke features resulting in two different matching scores. After each matching component produces an intermediate score, it will be transferred to the fusion component to calculate a final score by using fusion rules. Lastly, a decision is generated using the final score whether to accept or reject the user. Figure 2 illustrates the idea of SLSE.

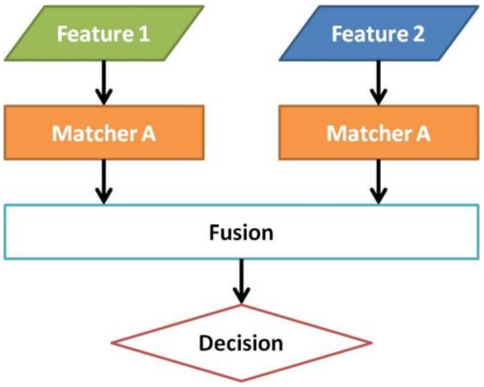

Figure 2. Block diagram illustrating the concept of SLSE.

2.3.2 Single Layer Multiple Expert (SLME)

Basically SLME is the exact inverse of SLSE where identical keystroke features are passed to different matching function. The other process flow remains the same in this approach. Figure 3 shows the concept of SLME.

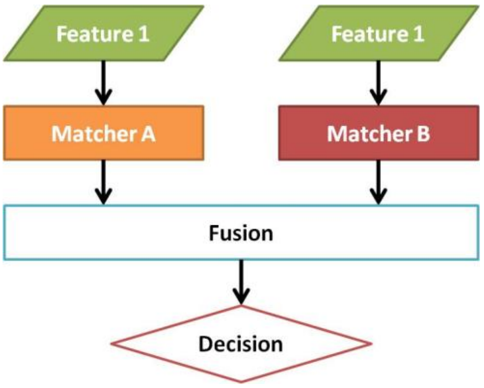

Figure 3. Block diagram illustrating the concept of SLME.

2.3.3 *Multiple Layer Multiple Expert (MLME)*

MLME, on the other hand, is a more in depth fusion approach which is derived from SLME. The idea is to merge the final scores produced by individual SLME models before final decision making. The number of SLME models involved depends on the number of keystroke features intended for fusion. Distinct keystroke feature is used within each individual SLME model itself with two identical matching function combinations, while among each SLME model uses different keystroke features. Figure 4 depicts the general concept of MLME.

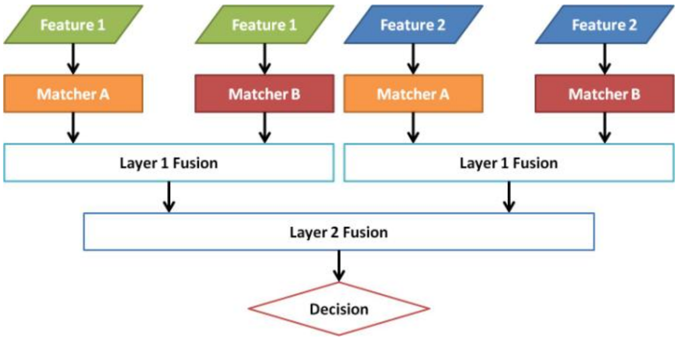

Figure 4. Block diagram illustrating the concept of MLME.

2.4 Fusion Rules

We employ six fusion rules in our fusion approaches. Apart from simplicity, the key advantage of these rules is that score normalization is not necessary since the output score from each matcher is already within the range of 0 to 1. Apart from that, they also neither incur system overhead nor require any additional changes to the existing component. Table 1 shows the formula for each fusion rule.

Table 1. Summary of the six fusion rules.

Fusion Rule	**Formula**
Sum	$S_f = \frac{s_1 + s_2}{2}$
Weighted Sum	$S_f = w_1 s_1 + w_2 s_2$
Product	$S_f = s_1 \times s_2$
Max	$S_f = MAX\left(s_1, s_2\right)$
OR Voting	$genuine = \begin{cases} reject & if \quad S_1 < thr, S_2 < thr \\ accept & \quad otherwise \end{cases}$
AND Voting	$genuine = \begin{cases} accept & if \quad S_1 > thr, S_2 > thr \\ reject & \quad otherwise \end{cases}$

where S_f represents the final score after fusion of partial scores S_1 and S_2.

3 EXPERIMENTAL SETUP

A total of one thousand keystroke timing data on a fixed phrase has been collected from one hundred users. This database collection process was conducted in two phases separated by an interval of four months apart. Phase I has 50 users while phase II consist of 100 users (inclusive of users from phase I and another new fifty users). All the users are people from university population, where by 37 of them are academic staffs followed by 45 undergraduate and postgraduate students while the remainder consists of technical or administrative staffs. Users are within the age of 18 to 40 years old and have a gender distribution of 59 males and 41 females.

Users have the option to perform the experiment in our pre-allocated desktop computer or by installing the programme into their preferable computer. Users will be prompted to type a fixed line of text “the brown fox” for ten consecutive times without typing error. Users are requested to type casually to reflex their normal typing behavior. Any erroneous phrase detected by the programme will not be recorded and the user is requested to retype the particular phrase. Some extra information such as their overall typing speed and how frequent they use a computer will also be collected for future analysis. The overview of our database setting is summarized in Table 2.

Table 2. Summary of our experimental dataset setting.

Property	**Description**
Input String	“the brown fox”
Population	50(Phase I), 50(Phase II)
Input Repetition	10
Total Sample	1000

Error Correction	Not Allowed
Supervision	No
Outlier Removal	No
Input Device	QWERTY keyboard
Device Freedom	Yes

Among the ten samples of text phrases collected, seven were used for training while the remainders were reserved for testing. The training sample sets were transformed into a user template. For the FAR (False Acceptance Rate) test, the first keystroke testing sample of each user in the testing set was compared against all the other users' keystroke templates. The identical matching process was repeated for all subsequent keystroke testing samples, which resulted in 29700 $(3\times[100-1]\times100)$ impostor attempts. As for FRR (False Rejection Rate) test, all the three testing keystroke samples of a user were matched against the keystroke template of the same user. The same matching process was repeated for all subsequent users, resulting in 300 (3×100) genuine attempts. Our experiment was repeated 10 times with the randomly selected combination of 7 training versus 3 testing data, and then the final result was averaged. All the results discussed in the later section will be portrayed with the average of FAR and FRR, the EER (Equal Error Rate).

4 RESULT DISCUSSIONS

4.1 Non Fusion Approach

In this subsection, comparisons have been made between four different keystroke features (D_1, D_2, D_3, D_4) used upon two proposed methods (GPD and DSM) without involving any fusion approaches. By observing Figure 5 we can make a performance comparison between different types of keystroke features used. The arrow in the graph shows the best result among all feature comparisons. We notice that by using D_1, we are able to obtain a better result compared to other keystroke features (D_2, D_3, D_4), particularly noticeable while using GPD. The complete EER of GPD and DSM performed on four different types of features are showed in Table 3.

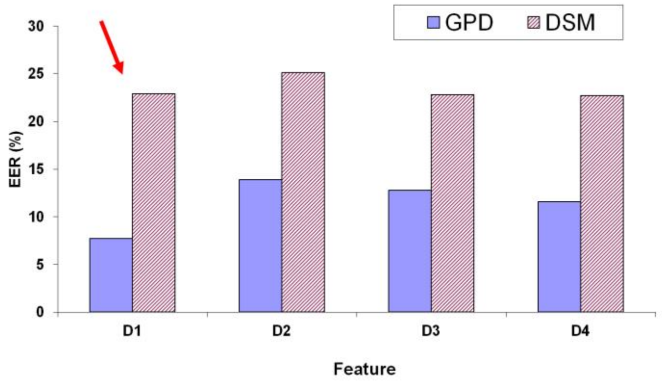

Figure 5. Comparison between the performance of four different keystroke features on GPD and DSM.

Apart from that we performed the experiment on two stages, one on a data sample of 50 users (data collection Phase I) and the other on 100 users (data collection Phase II). Experiment shows that as more data samples involved, the result remains consistent (Table 3), this shows the scalability of our method. Hence, all the experimental results shown for the subsequent fusion approaches are performed on 100 users.

Table 3. Performance comparison of four different keystroke features on different data size.

Method	Feature	50 Users	100 Users
GPD	D_1	7.9714	7.7199
	D_2	14.353	13.875
	D_3	12.554	12.776
	D_4	10.706	11.602
DSM	D_1	22.395	22.908
	D_2	22.946	25.112
	D_3	21.152	22.822
	D_4	21.382	22.744

4.2 Single Layer Single Expert (SLSE)

SLSE is among the first fusion approaches that we proposed, whereby the information of two different keystroke features is combined by using sum rule. As D_1 shows the best result among the four keystroke features, it can also be further proven in SLSE fusion approach. Experiment results show that the combination of D_1 with any other three keystroke features yield a better result as compared to other combinations without D_1 regardless of which method used (GPD or DSM). We note in Figure 6 and Figure 7 that D_1 combines with D_3 lead the best result of 5.13% and 14.879% EER for both GPD and DSM respectively. We see that an improvement is achieved for both methods in SLSE fusion approach. It is also worth noticing that fusion approach is appealing especially when we observe Figure 8, all the results obtained by using combination of two keystroke features are generally better compared to using only one keystroke feature.

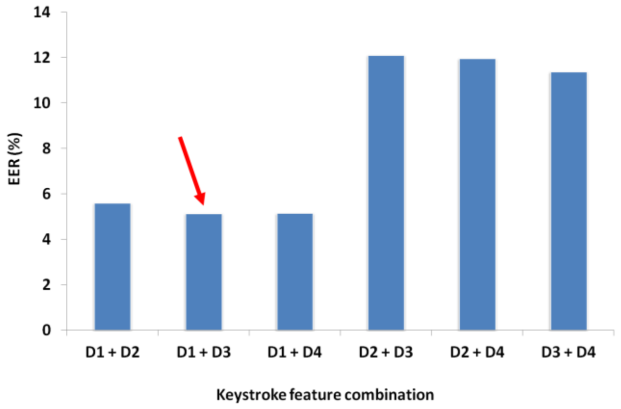

Figure 6. Performance of six different combinations of different keystroke features performed on GPD.

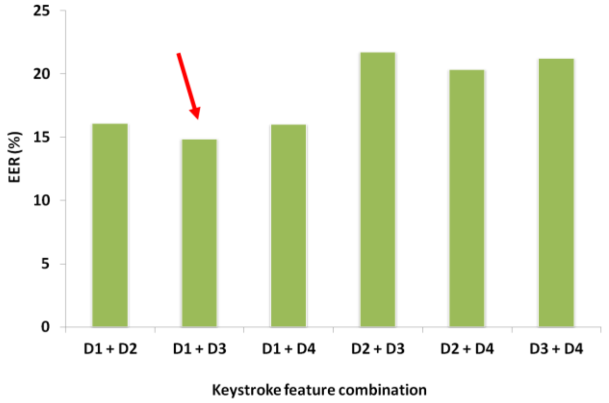

Figure 7. Performance of six different combinations of different keystroke features performed on DSM.

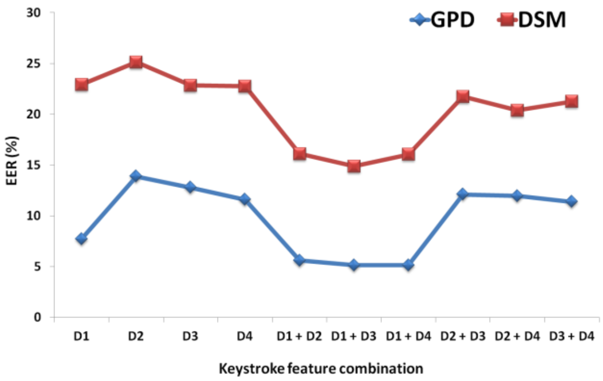

Figure 8. Comparison between the performances of four keystroke features used independently against the six different combinations performed on GPD and DSM.

4.3 Single Layer Multiple Expert (SLME)

In SLME, identical set of keystroke features is used on GPD or DSM. The output of both GPD and DSM (two matching scores) will then be fused together by using six fusion rules, which are sum rule, weighted sum rule, product rule, min-max rule, OR voting rule, and AND voting rule separately.

As discussed in the section above on non fusion approach and SLSE, using D_1 allows us to get a better result compared to the other three (D_2, D_3, D_4,). This claim is reinforced in SLME, as we can see from Figure 9 for most of the fusion rules used along D_1 result in a better EER. A detailed breakdown of EER of every combination of methods and features used in SLME is shown in Table 4. We can see that by using AND rule for fusion coupled with D_1 as keystroke feature produced an EER of 2.791%, which is the best result compared to other rules and features used.

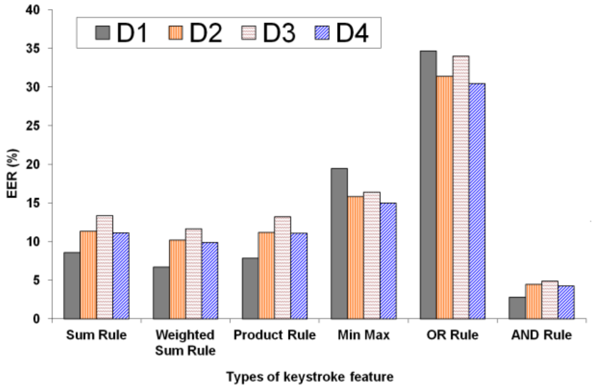

Figure 9. Comparison between the performances of four keystroke features used on six fusion rules to fuse score from GPD and DSM.

Table 4. Breakdown result of SLME fusion approach.

Fusion Rule	**Feature**			
	D_1	D_2	D_3	D_4
Sum	8.593	11.367	13.404	11.146
Weighted Sum	6.723	10.213	11.652	9.892
Product	7.847	11.201	13.232	11.071
Max	19.447	15.856	16.392	14.985
OR Voting	34.669	31.424	34.007	30.456
AND Voting	**2.791**	4.474	4.897	4.274

4.4 Multiple Layer Multiple Expert (MLME)

MLME fusion approach is the expansion of SLME, whereby the resulting fused scores from two or more sets of SLME with different latencies are fused together once more to get a final score. Once again the six fusion rules will be used as one of the fusion rules coupled with sum rule to form a two-layer fusion approach.

It is interesting to note that when D_1 is absent from any keystroke feature combination, the result is not as good as when D_1 is present as shown in Figure 10. This observation holds for almost all feature combinations against every fusion rule as shown in Table 5. To be more specific $D_1 + D_3$ yields the best result regardless which fusion rule is used. It is not surprising that those involving D_1 as part of the feature combinations outperform the others since this is also noticed in SLME as discussed in the above section.

Table 5 illustrates details of the performance comparisons of different fusion rules and keystroke feature combinations in MLME fusion approach. The best result was achieved at an EER of 1.404% in MLME by using sum rule coupled with AND rule as the fusion rule with $D_1 + D_3$ as keystroke feature. The obvious reason for the superior performance of MLME as compared to SLNF, SLSE, and SLME is because MLME involves two layers of fusion. Layer one which fuses the matching scores of two matching function (GPD and DSM), while layer two fuses the information of different keystroke feature combinations respectively. Hence, we can conclude that generally as more information are combined the higher accuracy can be obtained when distinguishing genuine and impostor.

It is noticeable that there is a big performance gap between OR and AND voting rule. The probable explanation may due to the individual performance of the two methods GPD and DSM. As we can see from Figure 5 in the previous section, GPD is able to perform better than DSM, in other words the chances of GPD accepting an impostor is low while DSM is higher. Assume a case when GPD rejects an impostor, while DSM has a higher chance to wrongly accept the impostor. If OR voting rule is used, the final decision will be to wrongly accept the impostor. Thus, this results to an overall degradation of the performance. On the other hand, if AND voting rule is used, the final decision will be accepted only when both GPD and DSM accept a user. Therefore, a stricter condition reduces the chance to wrongly accept an impostor hence increases the overall performance

The second best result we obtain among the six fusion rules is weighted sum rule at 3.733% EER. During our experiment the bias weight w_{GPD} and w_{DSM} were tested from the range of 0 to 1 with the step size of 0.1, we noticed that the best result obtained was when $w_{GPD} = 0.7$ and $w_{DSM} = 0.3$. Since GPD outperforms DSM when used individually, so it can be anticipated that weighted sum rule performs better than sum rule and product rule (both with approximately 4.3% EER) due to the way the weights are set.

On the other hand, Max rule was only able to attain an average EER of 11.37% among all keystroke feature combinations. This may due to the nature of this rule which favor towards a higher matching score regardless the score from either GPD or DSM. So, even though Max rule may be able to reduce the EER if compared to using DSM alone (EER=22.744%), but it degrades against the performance of GPD (EER=7.7199%).

Based on our observation, additional keystroke features enhance the overall performance as compared to using them individually. However, the combination of three and four keystroke features does not seem to have much advantage as compared to the combination of two keystroke features. The result from the combination of any two keystroke features involving D_1 generally out perform all the other combinations of more than two features regardless which fusion rule in used as shown in Table 5. Apart from the superiority of the performance from the combination of two keystroke features, the complexity and time consumption are logically lower as compared to utilizing three or four keystroke features. Hence we could say that in our experiment the optimal case is to use two keystroke features as combination, any more than that will not only yield minimal performance improvement but increase complexity and time consumption.

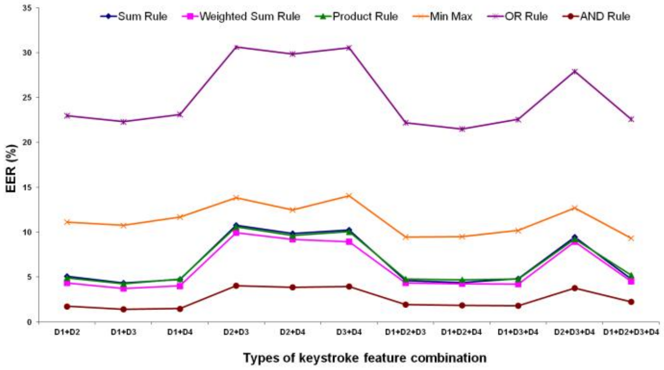

Figure 10. Comparison between the performances of different possible combinations of all four keystroke features in MLME fusion approach.

Table 5. Breakdown result of MLME fusion approach.

Feature	**Fusion Rule**		**EER**
Combination	**Layer 1**	**Layer 2**	**(%)**
$D_1 + D_2$	Sum	Sum	5.098
$D_1 + D_3$			4.34
$D_1 + D_4$			4.746
$D_2 + D_3$			10.763
$D_2 + D_4$			9.869
$D_3 + D_4$			10.234
$D_1 + D_2 + D_3$			4.589
$D_1 + D_2 + D_4$			4.394
$D_1 + D_3 + D_4$			4.836
$D_2 + D_3 + D_4$			9.45
$D_1 + D_2 + D_3 + D_4$			4.777
$D_1 + D_2$	Weighted Sum	Sum	4.351
$D_1 + D_3$			3.733
$D_1 + D_4$			4.008
$D_2 + D_3$			9.928
$D_2 + D_4$			9.199
$D_3 + D_4$			8.925
$D_1 + D_2 + D_3$			4.33
$D_1 + D_2 + D_4$			4.269
$D_1 + D_3 + D_4$			4.215
$D_2 + D_3 + D_4$			8.944
$D_1 + D_2 + D_3 + D_4$			4.506
$D_1 + D_2$	Product	Sum	4.908
$D_1 + D_3$			4.265
$D_1 + D_4$			4.768
$D_2 + D_3$			10.595
$D_2 + D_4$			9.614
$D_3 + D_4$			10.048
$D_1 + D_2 + D_3$			4.79
$D_1 + D_2 + D_4$			4.682
$D_1 + D_3 + D_4$			4.782
$D_2 + D_3 + D_4$			9.257
$D_1 + D_2 + D_3 + D_4$			5.23
$D_1 + D_2$	Max	Sum	11.126
$D_1 + D_3$			10.75
$D_1 + D_4$			11.705
$D_2 + D_3$			13.846
$D_2 + D_4$			12.471
$D_3 + D_4$			14.066
$D_1 + D_2 + D_3$			9.444
$D_1 + D_2 + D_4$			9.494
$D_1 + D_3 + D_4$			10.184
$D_2 + D_3 + D_4$			12.685
$D_1 + D_2 + D_3 + D_4$			9.336
$D_1 + D_2$	Sum	OR Voting	22.986
$D_1 + D_3$			22.298
$D_1 + D_4$			23.098
$D_2 + D_3$			30.624
$D_2 + D_4$			29.863
$D_3 + D_4$			30.533
$D_1 + D_2 + D_3$			22.181
$D_1 + D_2 + D_4$			21.477
$D_1 + D_3 + D_4$			22.564
$D_2 + D_3 + D_4$			27.906
$D_1 + D_2 + D_3 + D_4$			22.569
$D_1 + D_2$	Sum	AND Voting	1.736
$D_1 + D_3$			**1.401**
$D_1 + D_4$			1.475
$D_2 + D_3$			4.03
$D_2 + D_4$			3.875
$D_3 + D_4$			3.939
$D_1 + D_2 + D_3$			1.944
$D_1 + D_2 + D_4$			1.855
$D_1 + D_3 + D_4$			1.816
$D_2 + D_3 + D_4$			3.783
$D_1 + D_2 + D_3 + D_4$			2.239

4.5 Comparison with Other Techniques

In this section we make a result comparison among other existing methods. The best performance from our fusion approach (MLME) was selected for comparison. The performance comparison is based on each author’s own experiment settings and database respectively. Based on Table 6 it is noticeable that our proposed methods generally achieve better result as compared to most of the other existing methods. On the other hand, it is also worth noting that although our performance may be inferior to certain research works but the data size in our experiment is competitively larger. We can also observe that in most study, authors chosen D_1 or D_3 as part of the keystroke features for their experiments. This further supports our experimental result outcome where by combination of D_1 and D_3 provides us with the best result as we discussed in the section above.

Table 6. Comparison of existing research works with our proposed methods

Study	**Data Size**	**Feature**	**EER (%)**
Obaidat et al. [36]	15	D_1,D_3	0
Chang et al. [31]	20	D_1,D_3	1.2
Hocquet et al. [37]	13	D_1,D_3	1.8
Bartlow et al. [38]	41	D_1,D_3	2
Filho et al. [23]	47	D_2	12.7
Tran et al. [39]	40	D_1,D_2	8.6
Hosseinzadeh et al. [40]	41	D_1,D_2,D_3	4.4
Giot et al. [41]	100	D_1,D_2,D_3,D_4	6.96
Killourhy et al. [42]	51	D_1,D_2,D_3	9.6
Hwang et al. [43]	25	D_1,D_3	1
Ngugi et al. [44]	24	D_1,D_3	2
Balagani et al. [45]	33	D_1,D_2,D_3	1.72
This Paper	**100**	D_1,D_3	**1.401**

5 CONCLUSIONS AND FUTURE WORKS

As a summary, we have analyzed the influence of four keystroke features and by using fusion approaches to enhance the efficiency of a keystroke dynamic recognition system. We also show the consistency of performance with regards to the increase number of data size, which illustrate the sign of scalability.

Contrasting to normal fusion approach which usually only combine scores from different methods, MLME fuses more information by considering the scores from two different methods as well as the information of different combinations of keystroke features. The experimental results show that our proposed fusion method and keystroke feature combination are able to obtain reliable result at near to one percent of EER. D_1 offers the best performance among all four keystroke features if used independently, while the combination of D_1 and D_3 produces the optimal result in fusion mode. Hence, it is now clear why these two types of keystroke features are preferred by most of research works in keystroke dynamics domain. Lastly, based on our experimental result we recommend that D_1 and D_3 should both be utilized in any future keystroke dynamics research.

In view of the increasing popularity of mobile devices, we plan to conduct our experiment on mobile platform in the near future. It will be interesting to see what is the result and outcome when keystroke dynamics is implemented in high-end portable devices. We are in the process of collecting a dataset on the said platform and will be hoping to make it openly available in future, so that fellow researchers who are interested can use it as a benchmark dataset.

ACKNOWLEDGMENT

This work was supported in part by The University of Lincoln School of Computer Science Postgraduate Research Fund and in part by the EU FP7-PEOPLE-2010-IRSES Project (269118) Building a Visual Brain for Fast Human Machine Interaction (EYE2E).

REFERENCES

[1] L. C. F. Araujo, L. H. R. Sucupira, M. G. Lizarraga, L. L. Ling, and J. B. T. Yabu-uti, "User authentication through typing biometrics features," in *Biometric Authentication, Proceedings*, vol. 3072, 2004, pp. 694–700.

[2] R. S. Gaines, W. Lisowski, S. J. Press, and N. Shapiro, "Authentication by keystroke timing: Some preliminary results," Rand Corporation, Santa Monica, CA, R-2526-NSF, 1980.

[3] R. Joyce and G. Gupta, “Identity authentication based on keystroke latencies,” *Commun. ACM*, vol. 33, no. 2, pp. 168–176, 1990.

[4] W. G. De Ru and J. H. P. Eloff, “Enhanced password authentication through fuzzy logic,” *IEEE Expert*, vol. 12, no. 6, pp. 38–45, 1997.

[5] D. C. D’Souza, “Typing Dynamics Biometric Authentication,” University of Queensland, Queensland, 2002.

[6] R. N. Rodrigues, G. F. G. Yared, C. R. D. Costa, J. B. T. Yabu-Uti, F. Violaro, and L. L. Ling, “Biometric access control through numerical keyboards based on keystroke dynamics,” in *Advances in Biometrics, Proceedings*, vol. 3832, 2006, pp. 640–646.

[7] D. Hosseinzadeh, S. Krishnan, A. Khademi, and Ieee, “Keystroke identification based on Gaussian mixture models,” in *2006 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol III, Proceedings - Signal Processing Theory And Methods, Design And Implementation Of Signal Processing Systems, Industry Technology Tracks*, 2006, pp. 1144–1147.

[8] F. Monrose and A. D. Rubin, “Keystroke dynamics as a biometric for authentication,” *Future Gener. Comput. Syst.*, vol. 16, no. 4, pp. 351–359, 2000.

[9] C. C. Loy, W. K. Lai, and C. P. Lim, “Keystroke Patterns Classification Using the ARTMAP-FD Neural Network,” in *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*, 2007, vol. 1, pp. 61 –64.

[10] H. R. Lv and W. Y. Wang, “Biologic verification based on pressure sensor keyboards and classifier fusion techniques,” *Ieee Transactions on Consumer Electronics*, vol. 52, no. 3, pp. 1057–1063, 2006.

[11] S. Hocquet, J. Y. Ramel, and H. Cardot, “User classiflcation for keystroke dynamics authentication,” in *Advances in Biometrics, Proceedings*, vol. 4642, 2007, pp. 531–539.

[12] P. S. Teh, A. Teoh, T. S. Ong, and H. F. Neo, “Statistical Fusion Approach on Keystroke Dynamics,” in *Signal-Image Technologies and Internet-Based System, 2007. SITIS ’07. Third International IEEE Conference on*, 2007, pp. 918 –923.

[13] K. Xi, Y. Tang, and J. Hu, “Correlation Keystroke Verification Scheme for User Access Control in Cloud Computing Environment,” *Comput.J.*, vol. 54, no. 10, pp. 1632–1644, 2011.

[14] A. Kolakowska, “Generating training data for SART-2 keystroke analysis module,” in *Information Technology (ICIT), 2010 2nd International Conference on*, 2010, pp. 57–60.

[15] Wen-Pinn Fang, Hong-Ru Lee, and Fang-Pan Line, “An novel two layer user identification method,” in *Intelligence and Security Informatics, 2008. ISI 2008. IEEE International Conference on*, 2008, pp. 292–293.

[16] D. Gunetti and C. Picardi, “Keystroke analysis of free text,” *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 3, pp. 312–347, 2005.

[17] K. Revett, S. T. de Magalhaes, and H. Santos, “Data Mining a Keystroke Dynamics Based Biometrics Database Using Rough Sets,” in *Artificial intelligence, 2005. epia 2005. portuguese conference on*, 2005, pp. 188–191.

[18] A. Meszaros, Z. Banko, L. Czuni, and Ieee, *Strengthening passwords by keystroke dynamics*. 2007.

[19] K. A. Rahman, K. S. Balagani, and V. V. Phoha, “Making impostor pass rates meaningless: A case of snoop-forge-replay attack on continuous cyber-behavioral verification with keystrokes,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, 2011, pp. 31–38.

[20] S. Singh and K. V. Arya, “Key Classification: A New Approach in Free Text Keystroke Authentication System,” in *Circuits, Communications and System (PACCS), 2011 Third Pacific-Asia Conference on*, 2011, pp. 1–5.

[21] Y. Kaneko, Y. Kinpara, and Y. Shiomi, “A hamming distance-like filtering in keystroke dynamics,” in *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*, 2011, pp. 93–95.

[22] T. H. Cho and Ieee, *Pattern classification methods for keystroke analysis*. 2006.

[23] J. R. Montalvao Filho and E. O. Freire, “On the equalization of keystroke timing histograms,” *Pattern Recogn.Lett.*, vol. 27, no. 13, pp. 1440–1446, 2006.

[24] P. Teh, A. Teoh, C. Tee, and T. Ong, “A multiple layer fusion approach on keystroke dynamics,” *Pattern Analysis & Applications*, vol. 14, no. 1, pp. 23–36, 2011.

[25] A. Sulong, S. Wahyudi, and M. U. Siddqi, *Intelligent Keystroke Pressure-Based Typing Biometrics Authentication System Using Radial Basis Function Network*. 2009.

[26] H. Saevanee, P. Bhattarakosol, and Ieee, *Authenticating user using keystroke dynamics and finger pressure*. 2009.

[27] N. J. Grabham and N. M. White, “Use of a Novel Keypad Biometric for Enhanced User Identity Verification,” in *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, 2008, pp. 12–16.

[28] Hai-Rong Lv and Wen-Yuan Wang, “Biologic verification based on pressure sensor keyboards and classifier fusion techniques,” *Consumer Electronics, IEEE Transactions on*, vol. 52, no. 3, pp. 1057–1063, 2006.

[29] H. Nonaka and M. Kurihara, “Sensing Pressure for Authentication System Using Keystroke Dynamics,” in *International Conference on Computational Intelligence*, 2004, pp. 19–22.

[30] M. Villani, C. Tappert, N. Giang, J. Simone, H. S. Fort, and C. Sung-Hyuk, “Keystroke Biometric Recognition Studies on Long-Text Input under Ideal and Application-Oriented Conditions,” in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW ’06. Conference on*, 2006, p. 39.

[31] W. Chang, “Improving hidden Markov models with a similarity histogram for typing pattern biometrics,” in *IEEE International Conference on Information Reuse and Integration, Conf, 2005. IRI -2005*, 2005, pp. 487 – 493.

[32] K. Revett, S. T. de Magalhaes, and H. M. D. Santos, “Enhancing login security through the use of keystroke input dynamics,” in *Advances in Biometrics, Proceedings*, vol. 3832, 2006, pp. 661–667.

[33] J. C. Stewart, J. V. Monaco, S.-H. Cha, and C. C. Tappert, “An investigation of keystroke and stylometry traits for authenticating online test takers,” in *Biometrics (IJCB), 2011 International Joint Conference on*, 2011, pp. 1–7.

[34] H. Dozono, S. Ito, and M. Nakakuni, “The authentication system for multi-modal behavior biometrics using concurrent Pareto learning SOM,” in *Proceedings of the 21st international conference on Artificial neural networks - Volume Part II*, Berlin, Heidelberg, 2011, pp. 197–204.

[35] P. S. Teh, A. B. J. Teoh, C. Tee, and T. S. Ong, “Keystroke dynamics in password authentication enhancement,” *Expert Systems with Applications*, vol. 37, no. 12, pp. 8618–8627, Dec. 2010.

[36] M. S. Obaidat, “A verification methodology for computer systems users,” in *Proceedings of the 1995 ACM symposium on Applied computing*, 1995, vol. Nashville, Tennessee, United States, pp. 258–262.

[37] S. Hocquet, J. Y. Ramel, and H. Cardot, *Fusion of methods for keystroke dynamic authentication*. 2005.

[38] N. Bartlow and B. Cukic, “Evaluating the Reliability of Credential Hardening through Keystroke Dynamics,” in *Software Reliability Engineering, 2006. ISSRE ’06. 17th International Symposium on*, 2006, pp. 117 –126.

[39] D. Tran, W. Ma, G. Chetty, and D. Sharma, “Fuzzy and Markov models for keystroke biometrics authentication,” in *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*, WSEAS; Stevens Point, Wisconsin, USA, 2007, vol. Beijing, China, pp. 89–94.

[40] D. Hosseinzadeh and S. Krishnan, “Gaussian Mixture Modeling of Keystroke Patterns for Biometric Applications,” *Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, vol. 38, no. 6, pp. 816–826, 2008.

[41] R. Giot, M. El-Abed, C. Rosenberger, and Ieee, *Keystroke Dynamics With Low Constraints SVM Based Passphrase Enrollment*. 2009.

[42] K. S. Killourhy, R. A. Maxion, and Ieee, *Comparing Anomaly-Detection Algorithms for Keystroke Dynamics*. 2009.

[43] S. S. Hwang, H. J. Lee, and S. Cho, “Improving authentication accuracy using artificial rhythms and cues for keystroke dynamics-based authentication,” *Expert Systems with Applications*, vol. 36, no. 7, pp. 10649–10656, 2009.

[44] B. Ngugi, B. K. Kahn, and M. Tremaine, “Typing Biometrics: Impact of Human Learning on Performance Quality,” *J.Data and Information Quality*, vol. 2, no. 2, pp. 11:1–11:21, 2011.

[45] K. S. Balagani, V. V. Phoha, A. Ray, and S. Phoha, “On the discriminability of keystroke feature vectors used in fixed text keystroke authentication,” *Pattern Recogn.Lett.*, vol. 32, no. 7, pp. 1070–1080, 2011.