Enhanced Recognition of Keystroke Dynamics using Gaussian Mixture Models

Hayreddin Çeker and Shambhu Upadhyaya
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY, 14260
Email: hayreddi@buffalo.edu, shambhu@buffalo.edu

***Abstract*—Keystroke dynamics is a form of behavioral biometrics that can be used for continuous authentication of computer users. Many classifiers have been proposed for the analysis of acquired user patterns and verification of users on computer terminals. The underlying machine learning methods that use Gaussian density estimator for outlier detection typically assume that the digraph patterns in keystroke data are generated from a single Gaussian distribution. In this paper, we relax this assumption by allowing digraphs to fit more than one distribution via the Gaussian Mixture Model (GMM). We have conducted an experiment with a public data set collected in a controlled environment. Out of 30 users with dynamic text, we obtain 0.08% Equal Error Rate (EER) with 2 components by using GMM; while pure Gaussian yields 1.3% EER for the same data set (an improvement of EER by 93.8%). Our results show that GMM can recognize keystroke dynamics more precisely and authenticate users with higher confidence level.**

***Index Terms*—authentication, biometrics, Gaussian mixture model, human computer interaction, keystroke dynamics.**

I. Introduction

Keystroke dynamics is one of the efficient and inexpensive techniques that can authenticate computer users in the background while the user is actively working at the terminal. Typing characteristics have been shown to be distinctive enough to distinguish a computer user from another because of the unique timing of keystrokes that each individual performs during typing. There have been many proposed techniques for authenticating users using statistical approaches and machine learning. Of the statistical methods, the most common one is the Gaussian density estimator. It assumes that timing information of two consecutive keystrokes, better known as digraph, exhibits a pure Gaussian. The user can be identified if most of the typed digraphs are within a specific time interval determined by the mean and standard deviation of the distribution. However, the assumption of a single Gaussian can be too restrictive in practice. Depending on the word that the digraph is embedded in, the mean time can change for some cases significantly [20], which causes a degradation in accuracy of the system and leads to false rejections. Also, there are always some under-represented digraphs that do not frequently occur in language as pointed out in [11], which might require more advanced techniques for separability and user identification.

Gaussian Mixture Model (GMM) can simply be described as the weighted sum of Gaussian components. It can potentially represent seemingly complex and hard-to-map data to an understandable and distinguishable format. The use of GMM for representing dipraphs can also be supported by the common notion in pattern recognition that the individual Gaussians can model some underlying set of hidden features or attributes. That is to say, a user can be authenticated by some under-represented features that are consolidated using current methods. For example, a letter might be typed with different speeds depending upon the position within the word, as pointed out by Salthouse [16]. This distortion can only be acquired by a more complex estimator, for instance, the GMM.

In this paper, we focus on the fact that digraphs do not necessarily exhibit pure Gaussian but a mixture of Gaussians. Users might have different tendencies in typing some of the digraphs based on the word they are embedded in, the position, etc., which causes the pattern to be distorted or shifted to some extent. Therefore, since high accuracy in biometrics is required by many standards (The European Standard for Access Control (EN-50133) states that FAR should be less than 0.001% and FRR should be less than 1% for any commercially available authentication systems [13]), our paper investigates the application of GMM to keystroke dynamics for higher security and identification. The contributions of this paper include the demonstration of the strength/advantage of the mixture model over pure Gaussian in recognition of keystroke dynamics. Also, we want to show that existing studies which involve pure Gaussian can improve the accuracy by involving GMM.

The paper is organized as follows. The paper continues with the background information and a brief review of Gaussian Mixture Model in Section II followed by related work about the use of Gaussian distribution and mixture model in keystroke dynamics in Section III. Then, the details on data collection and feature extraction are given and the experiments conducted are described in Section IV. Section V presents the results of using GMM and its effectiveness in recognizing users over existing pure Gaussian models. Section VI discusses how current works can make use of GMM, and finally Section VII summarizes our findings and gives an insight into how our technique could be improved further.

II. Background

To better understand how digraphs spread over the distribution and how they vary, we rather go deeper into the historic literature in social sciences and provide a brief background about how human brain processes action of typing at low levels.

Salthouse [16] proposes a model for the steps taking place during typing. In the first stage, the text is perceived and divided into easily remembered chunks [4]. In the parsing stage, the perceived text is stored in memory for a short time and the chunks are separated into discrete characters. Having divided the text into chunks and then into characters can imply that not all digraphs are processed in the same way because the conversion can slightly change based on the chunks the digraphs are embedded in and the internal clock mentioned in [17]. For example, the pattern of the digraph *or* in the word *orange* may be somewhat different than that of *color*. Related to this example, it has been found that for the first keystroke in a word, the typing speed is generally slower than that of subsequent keystrokes in the word. This *word-initiation effect* has been documented clearly by Salthouse [15], where the latency of the first keystroke in a word is found to be approximately 20% longer than the latency of the following keystrokes.

In the translation stage, the characters are converted into movement commands. These commands specify which hand and finger to be used, and to which direction to extend. The fourth stage is the actual execution of the text followed by a feedback mechanism. After the keys are typed, a feedback is sent to ensure the accuracy.

A. Gaussian Mixture Model

Gaussian Mixture Model (GMM) is a parametric density function shaped by the weighted sum of Gaussian components. GMM generates a vector of mean values corresponding to each component and a matrix of covariance including components' variances and the co-variances between each other. GMM has been used for representing features in biometric systems [14] especially in speaker recognition as it has the potential to encompass a large set of sample distributions and fit arbitrarily shaped densities with smooth approximations.

Pure Gaussian fits the data by a single peak (mean) and an elliptic shape (variance); whereas GMM can represent it in higher dimensions by using a discrete set of Gaussian functions, each with its own mean and covariance matrix, to allow a better modeling capability.

GMM is expressed by the parameter set λ comprising of component weights w_i, mean vector $\vec{\mu_i}$ and covariance matrix Σ_i:

$$\lambda = \{w_i, \vec{\mu_i}, \Sigma_i\},\ i = 1, \ldots, M \tag{1}$$

The parameters are estimated using the iterative expectation–maximization (EM) algorithm [6]. Parameter lambda (λ) is updated in every iteration that yields a higher likelihood to refine the parameters and fit the distribution of the training dataset.

For the $\vec{x}$ vector, the mixture density is defined as the weighted linear combination of M pure Gaussian distributions:

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} w_i p_i(\vec{x}) \tag{2}$$

where

$$p_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu})'(\Sigma_i)^{-1}(\vec{x}-\vec{\mu})\right\}$$

In our experiments, we vary the number of components, M, from 1 (pure Gaussian) to 5 to investigate the effect of components on the accuracy of results.

III. Related Work

Over the years, many different methods and classifiers have been proposed in the area of keystroke dynamics to distinguish an individual from another. In security, typing pattern of a user is a behavioral biometrics that can be used as a means of authentication. In this section, basically the research on keystroke dynamics involving only Gaussian (normal) distribution is discussed. Also, the way the Gaussian distribution is integrated in the experiments is explored. Some studies report promising results and low error rates in the recognition of the individuals; while some of them deal with verifying active users transparently. However, because of the lack of a standard comparison method, it is a difficult task to compare previous works accurately. Therefore, Table I summarizes the related works referred in this section by listing the reported results and important details about the various experiments.

Study	FAR %	FRR %	EER %	Text Length	Free Text	# of users
Leggett et al. Static / Dynamic [11]*	5.0 / 12.8	5.5 / 11.1	-	Long	χ	36
Bleha et al. [2]	3.1	0.5	-	Short	χ	10+22$^\top$
Monrose et al. [12]	85.63 / 87.18 $^\gamma$			Short	χ	63
Hosseinzadeh et al.[9]	4.3	4.8	4.4	Short	$\surd$	41
Teh et al. [18]	-	-	6.46	Short	-	50
Deng et al. (GMM / GMM-UBM) [7]	-	-	8.7 / 5.5	Short	χ	51
Vural et al. [22] ($Th = 0.9/0.85$)	0.25 / 3.45	17.65 / 8.82	-	Long	$\surd$	39

*FAR stands for False Alarm Rate in that paper. $^\top$ The number of valid and invalid (random) users, respectively. $^\gamma$Only detection rates were reported. Th = Threshold

TABLE I: Comparison of Error Rates

Generally in statistical models, a reference profile is created using a feature set, then a test profile is compared against the reference to measure the similarity score as to verify the user's identity. Leggett et al. [11] use the digraph latency (the time elapsed between the two keys of the digraphs) as a feature in the profile creation process. The generation of the reference profile consists of recording all possible digraphs latency values and the calculation of their mean and standard deviation. Since every digraph is assumed to exhibit a Gaussian distribution, a normal curve is plotted using the corresponding mean and standard deviation to validate the digraphs in the testing process. Accordingly, the test digraphs are expected to fall within at most $\delta{*}\sigma$ distance from the mean (μ) of the normal curve which is represented as the *zone of acceptance* by the shaded area of Fig. 1.

The similarity of the test profile against the reference profile is calculated as the ratio of the number of digraphs fallen into the zone of acceptance over all digraphs. It is necessary for a user to pass a certain percentage of test digraphs to be accepted as legitimate, where 60% was found enough [21] to confirm that the test profile was typed by the same user. However, by

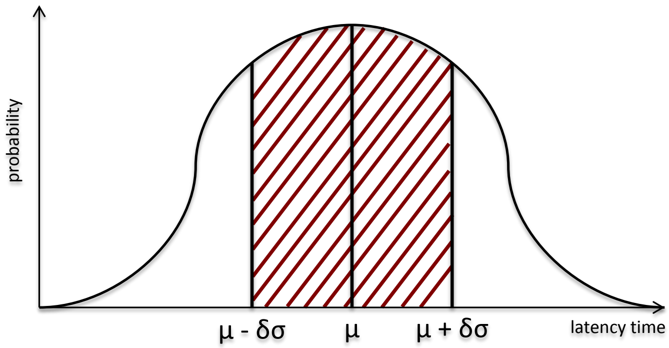

Fig. 1: Zone of Acceptance

utilizing the same technique, Vural et al. [22] report that with a δ = 1 distance and 0.85 threshold, they obtain a better result using their own dataset.

Similarly, Bleha et al. [2] extract features from keystroke timings and compare the test profile against the reference profile by setting a predetermined threshold value to accept or reject a user. They use a more complex classification method: multivariate Gaussian distribution, though, which is the generalization of Gaussian distribution in higher dimensions. Basically, the multivariate function is trained with each user's passphrase (or name) timing vector separately. Then, the user whose density is most likely to have generated the test vector is chosen as the legitimate one among the remaining users who could have passed the elimination process in previous steps. Hwang et al. [10] refer to the same technique as a novelty detector as to how the trained system can detect the outliers that are different from the normal ones. Also, note that multivariate Gaussian distribution is a different concept than the GMM that we use in this paper.

The works by Hosseinzadeh et al. [9] and Deng et al. [7] are the only ones that make use of GMM for keystroke dynamics. Hosseinzadeh et al. [9] measure the digraph latency, hold time and flight time and the combination of them to explore which fusion ends with the lowest error rate. They find out that hold time and flight time result in the lowest EER using decision-level (unanimity rule) fusion. Deng et al. [7], in addition, uses a Universal Background Model to train another GMM from a large pool of impostor subjects to generate an impostor profile to improve authentication. The experiments conducted in these studies are for verification purposes using only short texts (e.g., name, password) to be used as a secondary/supplementary authentication mechanism rather than continuous user monitoring and transparent authentication. The features are extracted when a user enters password several times in training session. The next time the user tries to login, the extracted features are compared and the user is authenticated if enough similarity is found. However, in our experiments each user has long text data to train the system. Each user has 20000 keystroke records on average. The separation of short vs long text is an important criterion since in long text data, the system can capture all digraph statistics and a user can be authenticated transparently by typing any arbitrary text; whereas short text data is only based on the length and sequence of the characters in the phrase, and users are required to type exact same phrase for every attempt.

Monrose and Rubin [12] generate a reference profile represented by N-dimensional feature vectors for a user. They assume that each feature is distributed by a Gaussian function, and the features are assigned a score using the corresponding Gaussian by applying similar procedures with the previous works. Now that a score is obtained for every feature, the final score is calculated by summing up the scores with two different methods: *weighted* and *non-weighted* probability measure. In non-weighted measure, all the weights are the same and the scores are directly added; whereas in weighted measure, the scores are multiplied with a weight. Teh et al. [18] also involve weighted sum rule by employing Gaussian probability distribution and another method to enhance the final result.

IV. Methodology

A. Data Collection

In the data collection process, a desktop environment is set up to record the keystroke data in a lab at Clarkson University. Thirty nine subjects are enrolled in two different sessions within a period of 11 months. Each session takes approximately 1 hour on two separate days. The users who didn't take the second session or who had a very high variability between the sessions are removed. At the end, 30 users are left for cross-validation.

The first session includes a set of survey questions that the subjects are asked to answer. The survey is carefully designed so that the subjects can respond to questions without long pauses and hesitations. To involve more natural typing effect in the experiment, some questions require subjects to choose their own writing topics. Also, the participants describe a picture of a crowded scene with various human activities. The second session consists of a static-text typing process in which Steve Jobs' famous commencement speech at Stanford University is required to be transcribed by the subjects.

The key-logger is a browser based Java Script program that collects the character and key's press and release time in millisecond. It records the timing data in real time and transfer them to a PHP web server. The program enforces subjects to type at least 500 characters to answer the questions. For more details on the data set, the reader may refer to Vural et al. [22] whose results can be found in Table I. This data set is publicly available for research by contacting the authors at Clarkson University.

B. Measure of Similarity

Digraph is the major feature used in keystroke dynamics [8, 19]. We are also utilizing the digraph latency between the press times of two consecutive keys in this paper. We expand the procedure for a single Gaussian described in [11] by applying it on Gaussian mixtures to which the digraphs are compared with each component separately. In this method, we created a 26x26 matrix in which rows and columns are assigned to letters and the values inside the cells correspond to the mean and standard deviation of the intersecting letters, i.e., digraph.

In our dataset, the digraphs whose latency is above 200 ms are ignored. Also, if the occurrence frequency of the digraph in the text is less than 50, we exclude it.

Algorithm 1 Digraph similarity algorithm

```
Input: Digraph latency information $D = \{l_1, l_2, ..., l_n\}$
Input: Number of components $M$
Input: Distance $\delta$
Output: Measure of similarity from every component $S = \{s_1, s_2, ..., s_m\}$
 1: Partition data $D$ into training $D_{train}$ and test $D_{test}$ dataset
 2: Fit data $D_{train}$ to a GMM with $M$ number of components
 3: Set $\vec{\mu}, \vec{\sigma}, \vec{w}$ from the previous step
 4: for i:=1 to M do
 5:     $pass := 0$
 6:     for j:=1 to $N_{test}$ do
 7:         if $l_j > \mu_i - \delta * \sigma_i$ and $l_j < \mu_i + \delta * \sigma_i$ then
 8:             $pass := pass + 1$
 9:         end if
10:     end for
11:     $s_i := pass * w_i$
12: end for
```

In the training session, the digraphs are parsed from raw data, and the latency is calculated as the difference between timestamps. We use the digraph latency between successive key presses as the measure of similarity as one of the inputs in Alg. 1. Data is partitioned into train (80%) and test (20%) data set. The GMM is trained with the digraph train dataset and the statistics are saved to corresponding vectors. Then, each data point in the test dataset is checked if it falls into the zone of acceptance referred in Fig. 1 within $\delta * \sigma$ tolerance (line 7). The numbers of passed data points are recorded for each component of the mixture and scaled with the weight vector to reflect the relative importance of the components on the output score vector S.

This process is repeated for every digraph of a particular user. Once we iterate over all users, we scale the scores resulted by Alg. 1 and report Equal Error Rate (EER) with respect to various threshold values.

V. Results

The test results are based on two separate long-text sessions from 30 users. Each user is compared against the other, and the similarity score is recorded. We use digraph latency with various options mentioned in [11] to reach the optimum configurations. We end up with the conclusion that using all alphabet letters yields the lowest error rate in distinguishing users. In contrast to Leggett et al. [11], including the space character does not improve the detection rate in our study.

False Accept Rate (FAR) and False Rejection Rate (FRR) are two common measurements to report the accuracy of a system. FAR is calculated by leave-one-out cross-validation where each user's similarity is compared with all other users. FRR is the ratio of the legitimate users who couldn't pass the threshold.

We occasionally use pure Gaussian to mean unimodal / single Gaussian (normal) distribution. Sometimes the word distribution (or density) is dropped from Gaussian distribution and only Gaussian (or Gaussians for plural) is used in the rest of the paper. Also, 1G (single Gaussian), 2G (GMM with 2 components), 3G (GMM with 3 components) abbreviations are used for ease of describing the results in this section.

The current practice in error reporting is generally by iterating over distance values with a predefined threshold. However, by varying both the distance (δ) and threshold (Th) values, we surprisingly found out that each component (M) has its own optimal configuration (Γ). 1G (pure Gaussain) exhibits its best result with $\{\delta = 1, Th = 0.95\}$; while 2G (GMM with 2 components) peaks at the configuration, $\{M = 2, \delta = 1, Th = 1\}$. Table II summarizes all the results and corresponding configurations.

Leggett et al. [11] set $\{\delta = 0.5, Th = 0.6\}$ to accept the digraph as valid, while Vural et al. [22] use $\{\delta = 1, Th = [0.85, 0.9]\}$ as the configuration. The corresponding error rates can be found in Table I of Section III. Note that we use the same dataset with Vural et al. [22] in our experiments.

Component	FAR %	FRR %	Distance (δ)	Threshold (Th)
1G	0.61	2.94	1	0.95
2G	0.09	2.94	1	1
3G	4.58	5.88	0.9	0.9
4G	4.93	2.94	0.9	0.9
5G	5.44	5.88	1.1	0.9

TABLE II: FAR and FRR

Nonetheless, comparing the accuracy of a system only with respect to FAR and FRR can sometimes be questionable. These two error measures tend to be mutually exclusive because if one of the rates improves to a considerable degree, the other one turns out to be disruptive [5]. Accordingly, in contrast to the enhancements in the false accept rate, having a better detection can cause an increase in the false rejection rate and users might be declined even if they're genuine. Therefore, Equal Error Rate (EER), the intersection of FAR and FRR, is suggested as a good candidate for comparative analysis [3]. In this way, the definition of *configuration* can be expanded to $\Gamma = \{M, \delta, Th, EER\}$ by adding the EER. Fig. 2 and Fig. 3 show a more detailed error reporting analysis.

Fig. 2 displays the EER with respect to distance. Although 1G begins with lower error for small thresholds (tight acceptance zone), 2G outperforms as it reaches to its best distance and threshold values. However, the recognition in higher dimensions degrades and doesn't follow a regular pattern because of the *singularities* that occur in maximum likelihood approaches (see Section VI).

The best distance values obtained from this experiment are used in Fig. 3 to plot the charts with respect to threshold. The charts in Fig. 3 display how FAR and FRR differ by the threshold values, along with the intersection points (EER) marked

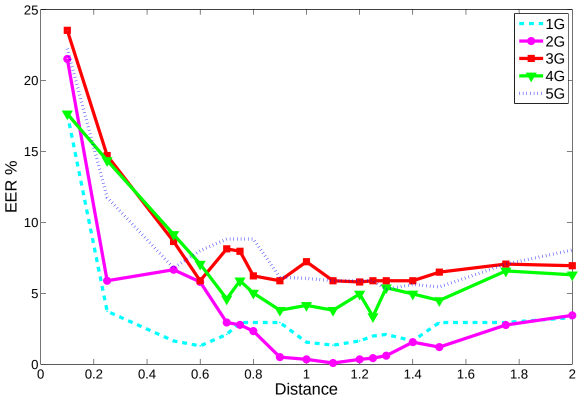

Fig. 2: EER % by distance values

with black dot to show the performance of each component. The corresponding EER values are listed in Table III. The EER analysis shows that, 2G can enhance the recognition by up to 93.8 % if applied properly on long text data.

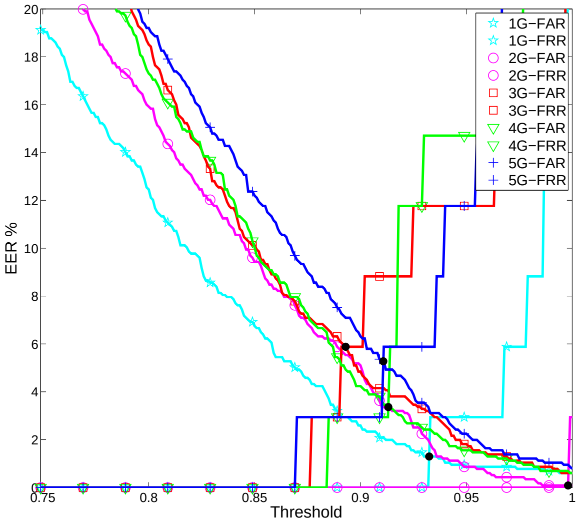

Fig. 3: Performance by threshold

1G	2G	3G	4G	5G
1.3 %	0.08 %	5.88 %	3.36 %	5.28 %

TABLE III: Equal Error Rate

VI. Discussion

In keystroke dynamics, false acceptance usually stems from the similarity in user's rhythm during typing the most common digraphs. In Gaussian-based identification, this statement corresponds to having similar mean and variance values. Gaussian Mixture Model (GMM) is capable of overcoming this issue by incrementing the number of components, if enough distinction is not provided. For example, Fig. 4 and Fig. 5 show the 'th' digraph distribution for two different users. Although the histogram bars are quite different, statistically they both share similar mean and standard deviation with 1 and 2 Gaussians (components). However, when we run the experiment with 3 components, the users are separated to a considerable degree. In this way, the system can be adjusted to involve more components and separate the graph into 2, 3, 4 and more Gaussians until it eventually makes the separation clear with additional performance overhead. Consequently, existing studies which employ Gaussian distribution might enhance the recognition, by using GMM where applicable.

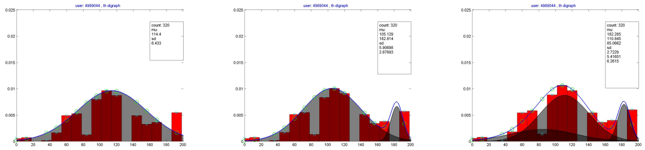

Fig. 4: GMM with 1, 2 and 3 components

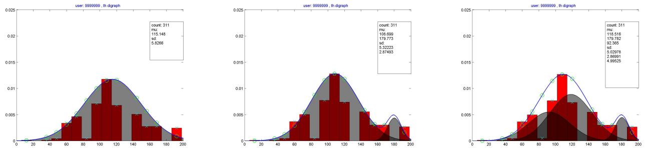

Fig. 5: GMM with 1, 2 and 3 components

In this paper, we show that 2G outperforms 1G in almost all distance values. However, training in higher dimensions (3G, 4G, etc.) can sometimes adversely affect the recognition as shown in Fig 2. This phenomenon can be explained by the *curse of dimensionality* notion and presence of singularities [1]. Basically, when one of the Gaussian components 'collapses' onto an outlier data point, it may cause severe overfitting that can occur in a maximum likelihood (ML) approach in expectation–maximization algorithm. ML estimates may lead to mixture models with high variance which yields overconfident predictions when it underestimates the noise level [23]. Adopting a Bayesian approach can alleviate this problem for which we will leave as future work. Readers may refer to Bishop's book [1] for more detail.

VII. Conclusion

Psychological experiments show that keystroke dynamics are performed by a set of actions that in each stage, it implements uniqueness to user's typing behavior [16]. Keystroke dynamics can be used in security to authenticate users as they provide enough distinction if the stages involved during typing are addressed properly. Since the digraph pattern can be distorted based on the position or the word it is embedded in, Gaussian Mixture Model (GMM) is a more appropriate tool to model the digraphs rather than fitting it into a pure Gaussian with a single mean and variance. In our experiments, we show that 2G (GMM with 2 components) can enhance the recognition by 93.8% over the commonly used 1G (pure Gaussian). 2G reduces the error rate from 1.3% EER down to 0.08 % EER. This improvement not only demonstrates the high accuracy in detection but also signifies the fact that existing studies can enhance their recognition by applying the mixture model elaborated in this paper.

In addition, our work points out how the optimal distance and threshold values can change by varying the number of components. While 2G outperforms 1G, the improvement does not follow a regular pattern.

The variability of optimal configuration value based on the component number will be investigated as a future work along with user specific parameters emphasized in [9]. Furthermore, the implementation of a generic algorithm to adjust the parameters is in our future plans. The algorithm will make the sensitivity analysis and decide the required number of components that is enough for the system to differentiate all users, taking into account the required accuracy and performance overhead.

References

[1] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[2] S. Bleha, C. Slivinsky, and B. Hussien. Computer-access security systems using keystroke dynamics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(12):1217–1222, 1990.

[3] N. L. Clarke and S. Furnell. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*, 6(1):1–14, 2007.

[4] W. E. Cooper. *Cognitive aspects of skilled typewriting*. Springer, 1983.

[5] B. Cope. Biometric systems of access control. *Electrotechnology*, 18:71–4, 1990.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[7] Y. Deng and Y. Zhong. Keystroke dynamics user authentication based on gaussian mixture model and deep belief nets. *International Scholarly Research Notices*, 2013, 2013.

[8] S. Hocquet, J.-Y. Ramel, and H. Cardot. User classification for keystroke dynamics authentication. In *Advances in biometrics*, pages 531–539. Springer, 2007.

[9] D. Hosseinzadeh and S. Krishnan. Gaussian mixture modeling of keystroke patterns for biometric applications. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(6):816–826, 2008.

[10] S. Hwang, H. Lee, and S. Cho. Improving authentication accuracy using artificial rhythms and cues for keystroke dynamics-based authentication. *Expert Systems with Applications*, 36(7):10649–10656, 2009.

[11] J. Leggett, G. Williams, M. Usnick, and M. Longnecker. Dynamic identity verification via keystroke characteristics. *International Journal of Man-Machine Studies*, 35(6):859–870, 1991.

[12] F. Monrose and A. D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Generation computer systems*, 16(4):351–359, 2000.

[13] D. Polemi. Biometric techniques: review and evaluation of biometric techniques for identification and authentication, including an appraisal of the areas where they are most applicable. *Reported prepared for the European Commision DG XIIIC*, 4, 1997.

[14] D. Reynolds. Gaussian mixture models. *Encyclopedia of Biometrics*, pages 659–663, 2009.

[15] T. A. Salthouse. Effects of age and skill in typing. *Journal of Experimental Psychology: General*, 113(3):345, 1984.

[16] T. A. Salthouse. Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological bulletin*, 99(3):303, 1986.

[17] L. Shaffer. Attention and performance. *Latency Mechanisms in Transcription*, IV, 1973.

[18] 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*, pages 918–923. IEEE, 2007.

[19] P. S. Teh, A. B. J. Teoh, and S. Yue. A survey of keystroke dynamics biometrics. *The Scientific World Journal*, 2013, 2013.

[20] C. Terzuolo and P. Viviani. Determinants and characteristics of motor patterns used for typing. *Neuroscience*, 5(6):1085–1103, 1980.

[21] D. Umphress and G. Williams. Identity verification through keyboard characteristics. *International journal of man-machine studies*, 23(3):263–273, 1985.

[22] E. Vural, J. Huang, D. Hou, and S. Schuckers. Shared research dataset to support development of keystroke authentication. In *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, pages 1–8, Sept 2014.

[23] S. Waterhouse, D. MacKay, and T. Robinson. Bayesian methods for mixtures of experts. *Advances in neural information processing systems*, pages 351–357, 1996.