Predicted Templates: Learning-curve Based Template Projection for Keystroke Dynamics

1st Ali Khodabakhsh
Department of Information Security and Communication Technology, Norwegian University of Science and Technology
Gjovik, Norway
ali.khodabakhsh@ntnu.no

2nd Erwin Haasnoot
Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente
Drienerlolaan, Netherlands
e.haasnoot@utwente.nl

3rd Patrick Bours
Department of Information Security and Communication Technology, Norwegian University of Science and Technology
Gjovik, Norway
patrick.bours@ntnu.no

***Abstract*—Keystroke Dynamics (KD) as a biometric modality can provide authentication tools in many real-life applications, virtually at zero-cost on the client side, due to the reliance of these techniques on existing hardware, and their low computational expense. One promising application is the use of KD as a second factor in password-based authentication. A downside of the existing modeling methods is the assumption of stationary behavior from the clients. However, it is expected that humans show improvements in performing a specific task following practice. In this study, we propose methods for utilization of learning models in predicting the future behavior of the clients, even with little enrollment data, and generate predicted behavioral models that can be used in different classifiers. In our experiments, the predicted templates show a reduction in the average equal-error-rate (EER) consistently across different classifiers a benchmark dataset. A reduction of 20% is achieved on the best classifier. Given fewer enrollment data, the performance gain was shown to reach above 30%. Furthermore, we show that blind detection of attacks is possible, solely relying on the global learning curve, with an EER of 16%.**

***Index Terms*—Keystroke Dynamics, Learning Curve, Predicted Template, Keystroke Biometrics**

I. Introduction

Keystroke Dynamics (KD) provides simple and effective tools for biometric authentication of users. These methods are easily deployable in a wide range of access control applications, as they do not require additional hardware or any adaptation from the clients. KD has a long history of application [1] , and the effectiveness of these systems has been the focus of many studies [2]. A major use-case for KD is as a second factor for password authentication. The need for more security in password authentication is evident as for the existing common issues of the password-only authentication. Examples are password sharing, same password for multiple accounts, insecure passwords, attack techniques such as phishing, and password leaks. Many datasets have been proposed for the task, and good detection accuracies have been achieved [2].

A major disadvantage of KD as a biometric is its relative low permanency compared to other biometrics. To address this problem, template updating methods have been studied recently [3], [4] . The goal of template updating mechanisms is to recover performance drops due to changes in typing patterns by either periodically or constantly updating the stored templates. These changes are caused by many factors ranging from environmental factors (e.g. new input devices and interaction position), to behavioral ones (e.g. mood, adapting new typing behavior, and improvements in typing proficiency). As the matter of fact, template updating introduces new attack vectors to biometric systems, and methods used to report template update performance do not always map well to applications in practice [3]. Among the factors that influence typing patterns, gradual improvement by practice is well-studied in the field of psychology [5], and mathematical models have been proposed for modeling these learning trends [6]. Since the effects of practice and learning on typing patterns are not well-studied in KD [7], it is worth while to investigate the use of learning curves to predict behavior changes pro-actively.

Template prediction differs from template updating as it updates templates in advance, taking into account the predictable changes in the behavior of the subject, in contrast to template updating, which does so re-actively. Template prediction can improve the permanency of KD templates by removing predictable variability factors, but cannot be a replacement of template updating as it does not provide any mechanisms for unpredictable sources of variability. In this study, we take initial steps for taking advantage of the learning curve trend and show significant gains in the performance of the resulting systems. The proposed methods are flexible and can be adapted for many existing classifiers by detrending the training data.

The rest of this document is organized as follows: the methods proposed in this study are explained in section II. Experiment setup is presented in section III, followed by the discussion on the results in section IV. Finally, the paper is concluded in section V.

II. Methodology

Many features are recorded in different KD applications (e.g. pressure, mouse movements, etc) but the most commonly used KD features are derived from timing information, in particular, the time when a key is pressed and released [8].

Later, machine learning based or statistical models are trained on the enrollment data and given a probe, the similarity or dissimilarity of the sample to the model is calculated. Modeling is usually done with the assumption of stationary behavior from the client which is not always true. In this section, we explain how the learning curve can be used for generating predicted templates (PT).

The features in the focus of this study are derived from the key-press and -release times as listed below:

- **Duration (aka hold or dwell time):** The time between pressing a specific key and releasing it.
- **Press-Press latency (PP-latency):** The time between pressing one key and pressing the next one. This feature follows a learning curve, as shown in Fig. 1.
- **Release-Press latency (RP-latency) (aka flight time):** The time between releasing one key and pressing of the next key. This RP-latency can be negative as the next key can be pressed before the previous is released. Furthermore, it is linearly related to the previous two features, and can be calculated by subtracting the duration from PP-latency.

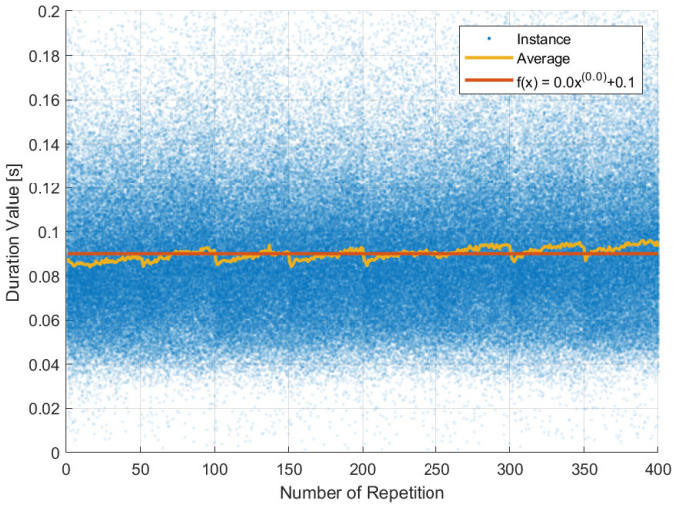
A. Learning Curve

The 3-parameter learning curve can be described using the power-law (PL) formula [6], $g[r] = ar^b + c$, where r is the repetition number, a is the slope, b is the power, and c is the asymptote. Given the time series on training data $g[r_i], r_i \in R$ sampled at repetition number set R, the PL parameters a, b, and c can be estimated using non-linear least square method. The resulting parameters can then be used to predict the value of $g[r_t]$, where r_t is the probe repetition number. As a result, ideally, the model can be represented as the set of PL parameters $P = \{a, b, c\}$.

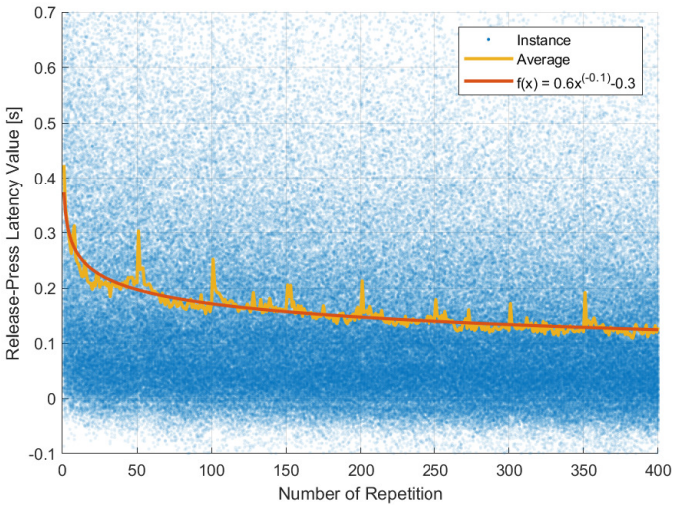
B. Bayesian Inference

In many applications, it is not realistic to assume always having enough samples of $g[r]$ series to have an accurate estimate of the PL parameters P. To alleviate this problem, one can generate an accurate estimate of the global model parameters P_0 over the general durations and latencies over a large number of repetitions in a subject independent manner, to be used as a prior model. Having such an accurate estimate as a prior, given observations from subject model parameters P_{s_0} of a trained model on limited noisy samples from his/her training data, one can arrive at the updated parameters using Bayesian inference.

The updating mechanism is explained as follows on each individual model parameter. For the sake of simplicity, appealing to the central limit theorem, each fitted parameter in the PL model follows a normal distribution, with a mean of μ and a standard deviation of σ derived from the estimated standard error. For a specific parameter, the global fit provides initial values μ_0 and σ_0, while the fit on a specific subject from a number of samples n, leads to values μ_{s_0} and σ_{s_0}. Now, the



(a) Duration values



(b) Release-Press latency values

Fig. 1: All instances of (a) duration and (b) release-press features, across all subjects, plotted over the number of repetition. The average values are shown in yellow and the power-law function fitted to average values in each case is reported in its legend.

posterior distribution $\mathcal{N}(\mu_s, \sigma_s^2)$ can be calculated using the following equations:

$$\sigma_s^2 = \left(\frac{1}{\sigma_0^2} + \frac{1}{\sigma_{s_0}^2/n}\right)^{-1}, \mu_s = \sigma_s^2\left(\frac{\mu_0}{\sigma_0^2} + \frac{\mu_{s_0}}{\sigma_{s_0}^2/n}\right) \quad (1)$$

This process can be done for each parameter in the model separately to acquire the posterior distribution means μ_s, which will be used as inferred model parameters P_s.

The global parameters can be learned per feature, however, such a feature specific model would be of limited use. In this study, a more fruitful approach is taken by learning parameters over global features (i.e. duration, PP-, and RP-latencies), in a key-inspecific way. An example of the process is explained and

depicted in section IV-A. This allows a more accurate average model and a wider applicability of the global parameters.

C. Blind Detector

In real-life scenarios, the client is usually in the later stages of the learning curve, while the unpracticed attacker is in the initial stages. Even though the assumption of having an unpracticed attack may not always be true, this gap may be utilized for detection of attacks in special cases, where the assumption can be made, in the following manner. By bypassing the subject based inference step, the global parameters can on their own be used as a subject-independent model. This model, instead of presenting how the individual behaves, represents how well a well-practiced person will behave.

Such a model does not rely on enrollment data, hence the name blind, and as a result, it does not have a differentiation power between different subjects, as it will generate the same model for all subjects. Thus, it can only function as to contrast between existing bona fide (BF) instances and presentation attacks (PA).

D. Classifiers

In this study, four statistical methods (distance measures) are used as classifiers. The motivation behind this selection was the similarity between the modeling across all these classifiers, as well as the comparison possibility they provide in reference to the benchmark dataset [9]. These are Euclidean, Normed Euclidean, Manhattan, and Scaled Manhattan distances, implemented in accordance with the reference study [9]. These distance measures rely on a template consisting of a mean vector and an average absolute deviation (AAD) vector.

III. Experiment Setup

A. Dataset

Killourhy dataset [9] is selected for this study due to being one of the largest freely available datasets with a high number of repetition per subject. This dataset consists of data collected from 51 subjects. A total of 400 repetitions per subject were recorded in 8 sessions with at least one-day of spacing between two subsequent sessions. The dataset provides duration (called hold in [9]), PP-latency, and RP-latency values for each key in the password ".tie5Roanl" plus the final Enter key-press, while joining the Shift+R key combination as a single key-press. This results in 11 duration values, 10 PP-latency values, and 10 RP-latency values, in a timing vector of 31 numbers per repetition. The setup used in this article matches that of [9].

B. Parameters

a) Baseline:: For the baseline system, models are generated by calculating the average and AAD of the training observations per feature, resulting in a model vector of 31 means and 31 AAD values.

b) Predicted Template:: To generate the PTs, the training observations are used to train one PL function per feature, resulting in 31 PL parameter sets. A conservative model is then generated by averaging the predictions of each PL function over the repetition number range 201-400, corresponding to the last four sessions. The AAD model is calculated by the same method used for the baseline system.

c) Predicted Template with Prior:: To estimate the global PL function parameters, the leave-one-out approach is used. After excluding the data from the target subject, all the key entries over all the features of the same type (duration, PP-latency, and RP-latency values) are pooled, and the average is calculated for each repetition number (Fig. 1). Then, a PL function is fitted on the resulting values. The outcome of this process is 3 sets of PL parameters corresponding to each of the 3 types of features. These parameters are then used as prior for the parameters generated by the PT method, based on their corresponding feature category, and the posterior model is inferred using Eq. 1. The rest of the modeling is done in the same manner as for PT.

d) Blind Detector:: The blind detector relies solely on the global PL function parameters for generating the static predicted model. To avoid the effect of using the training data of target subjects in generating the global PL functions, again, the leave-one-out approach is used. Then, similar to PT, for each feature, samples from the corresponding category in the range of 201-400 are generated and averaged. To generate the AAD model, the AADs of repetitions of each subject for each feature is calculated and averaged per feature type.

e) Accuracy Measures:: To measure the performance of the proposed methods in a way comparable to the original study, the average equal-error-rate (EER) measure [9] is selected.

IV. Results and Discussion

A. Learning Curve

Fig. 1 presents all instances of features for each category, along with their average, and a PL function fitted to it. The duration values remain almost static as shown in Fig. 1a. There is a slight upward trend which was not captured by the PL function due to the lack of a linear term. However, the fitted PL function represents the mean of the data accurately. The RP-latency values show a clear exponential trend as depicted in 1b. This trend was well captured by the fitted PL function. The significance of the effects is especially evident in comparison with the initial repetitions with the last 100, showing a ratio of approximately 2 to 1. The same pattern was observed for PP-latency values, as PP-latencies are essentially the RP-latencies plus the duration values. It is interesting to observe the piece-wise recurrence of the overall patterns inside each session, mostly visible in repetitions 50, 100, 150, and 200, as a result of spacing between sessions [5].

B. Predicted Templates

As displayed in Tab. I the PTs outperform the baseline consistently across all classifiers. The percentile reduction in

TABLE I: Performance of the proposed systems in contrast to the baseline system in terms of mean equal error rates and their corresponding standard deviations, on the benchmark task. (The last column corresponds to 200 enrollment repetition in Fig. 2)

	Euclidean		Euclidean (normed)		Manhattan		Manhattan (Scaled)	
	Avg EER	(STD)	Avg EER	(STD)	Avg EER	(STD)	Avg EER	(STD)
Baseline	0.171	(0.095)	0.215	(0.119)	0.153	(0.093)	0.096	(0.069)
Predicted	0.142	(0.073)	0.148	(0.079)	0.110	(0.072)	0.077	(0.060)
Predicted w/ Prior	0.205	(0.169)	0.188	(0.145)	0.120	(0.084)	0.079	(0.057)
Blind	0.164	(0.162)	0.173	(0.124)	0.160	(0.153)	0.182	(0.168)

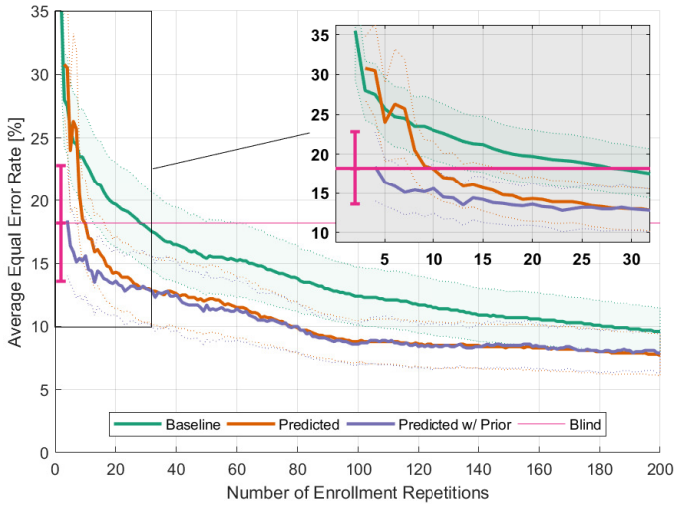


Fig. 2: Average equal error rate (with 95% confidence intervals) vs the number of enrollment repetitions for scaled Manhattan classifier on proposed methods and the baseline system. The graph starts at 3 for predicted templates and 4 for predicted templates with prior, as at least 3 observations are required to fit a power-law function, and another observation for estimating its parameters' confidence intervals.

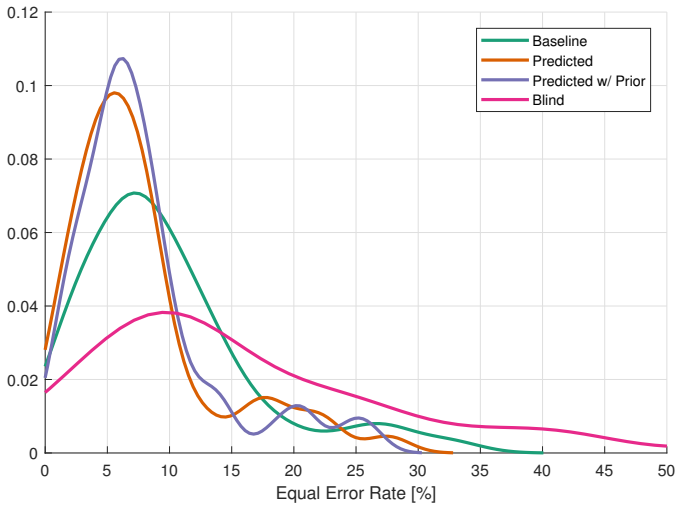


Fig. 3: Normalized equal error rate histogram (using kernel density estimation) of proposed systems in comparison to the baseline system and the blind method for the scaled Manhattan classifier.

the mean EER is 17%, 31%, 28%, and 20% for Euclidean, Normed Euclidean, Manhattan, and Scaled Manhattan classifiers respectively. The effect of the number of enrollment samples on the performance of the PTs is depicted in Fig. 2 for the best classifier (Scaled Manhattan). The PTs systematically perform better than the baseline for large numbers of enrollment repetitions consistently, however, they show no improvement if the number of enrollment repetitions drops below 8.

C. Predicted Templates with Prior

Using the prior knowledge has a negative impact of the performance of the system, as shown in Tab. I, however, compared to the PT method this negative impact is not significant (except for the Euclidean classifier). Following the standard experiment setup, 200 enrollment repetitions are used, and it was possible to generate the PTs with high accuracy without a need for prior information. However, the benefits of using prior knowledge are observable in Fig. 2. PT with prior outperform the PT method consistently and significantly when the number of enrollment repetitions falls below 15. As in most real-life applications, the number of enrollment repetitions falls in the lower range of 2, this method can be positively incorporated.

D. Blind Detector

The blind detector shows a very high performance, close to the baseline system, for Euclidean and Manhattan classifiers as shown in Tab. I. The high performance of this system can be explained by the major difference in the BF and PA trial selection in the dataset. The BFs are selected from after the first 200 trials, while the PAs are selected from the first 5 of every other subject. This major difference causes the BFs to have a high similarity with the blind detector model (which represents practiced behavior), while the PA trials are dissimilar (representing unpracticed behavior). This also shows the significant impact of the learning curve on the performance of the systems in this setup. It is important to note that the standard deviation of this system is very high compared to all other systems. To analyze this further, the histogram of the subject EERs are plotted in Fig. 3. The EERs of the blind detector have a more flat distribution along the x-axis, showing its performance to be variable for different subjects. Nevertheless, on average it performs well due to the high accumulation of EERs in the lower range. This system is the

only system where Manhattan classifier outperforms scaled Manhattan, showing the ineffectiveness of AAD estimation.

V. Conclusion and Future Work

In this paper, multiple methods have been proposed for utilization of the learning curve in template prediction. Consequently, the average EER of multiple classifiers has been reduced by 17% to 31% on a standard dataset. The proposed systems can be used as the modeling step of different classifiers and can provide outstanding performance even with a small number of enrollment data. A blind detector has also been proposed with an average EER of 16%, which can be incorporated without a need to individual subject modeling. This system, due to its simplicity, can have a wide range of applications, however, its variable performance across different subjects must be taken into account.

The future work includes: incorporating template prediction into more complex classifiers, evaluating the methods on more recent benchmark datasets, incorporating spacing information and the piecewise power laws [5] in the modeling, and replacing the 3-parameter PL function with a 4-parameter model that includes a delay factor. It is also recommended to study the transferability of the trained global model to other datasets.

VI. Acknowledgment

The authors would like to acknowledge the kind and invaluable comments of Mazaher Kianpour and Pawel Drozdowski that resulted in significant improvements in the paper.

References

[1] I. BioPassword, "Authentication solutions through keystroke dynamics," *BioPassword, Inc., Tech. Rep*, 2006.

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

[3] M. M. Seeger and P. Bours, "How to comprehensively describe a biometric update mechanisms for keystroke dynamics," in *2011 Third International Workshop on Security and Communication Networks (IWSCN)*, May 2011, pp. 59–65.

[4] R. Giot, B. Dorizzi, and C. Rosenberger, "Analysis of template update strategies for keystroke dynamics," in *2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM)*, April 2011, pp. 21–28.

[5] Y. Donner and J. L. Hardy, "Piecewise power laws in individual learning curves," *Psychonomic Bulletin & Review*, vol. 22, no. 5, pp. 1308–1319, Oct 2015.

[6] A. Newell and P. S. Rosenbloom, "The soar papers (vol. 1)," P. S. Rosenbloom, J. E. Laird, and A. Newell, Eds. MIT Press, 1993, ch. Mechanisms of Skill Acquisition and the Law of Practice, pp. 81–135.

[7] E. Haasnoot, S. J. Barnhoorn, J. L. Spreeuwers, J. N. R. Veldhuis, and B. W. Verwey, "Towards understanding the effect of practice on behavioural biometric recognition performance," in *European Signal Processing Conference (EUSIPCO)*, september 2018.

[8] S. P. Banerjee and D. L. Woodard, "Biometric authentication and identification using keystroke dynamics: A survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116–139, 2012.

[9] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *IEEE/IFIP International Conference on Dependable Systems Networks*, June 2009, pp. 125–134.