Applying Partially Observable Hidden Markov Model to keystroke dynamics

Alessio Cuccurullo
Università degli Studi di Salerno
Dipartimento di Informatica
Email: a.cuccurullo12@studenti.unisa.it

Paolo Panico
Università degli Studi di Salerno
Dipartimento di Informatica
Email: p.panico6@studenti.unisa.it

Vincenzo Veniero
Università degli Studi di Salerno
Dipartimento di Informatica
Email: v.veniero@studenti.unisa.it

***Abstract*—There have been research activities using Hidden Markov Models (HMM) undertaken in the past decades. However, in terms of Partially Observable Hidden Markov Model (POHMM) there has been little published work. Owing to the widespread use of the HMM, it is necessary for us to examine the effectiveness of the POHMM in the domain of keystroke dynamics biometrics. The aim of this paper is to provide some insights and comparative analysis of the difference between the two models referred to earlier. The POHMM is shown to outperform the standard HMM, in biometric identification, verification and continuous verification tasks.**

I. Introduction

Biometric authentication systems are being widely used in our daily life for identity management (e.g., biometric passport). Nowadays, many biometrics-based solutions exist and are divided into two main categories: physical (such as face [1]) or behavioral (such as keystroke dynamics [2]). In this paper, we are interested in the second category which is keystroke dynamics authentication systems, and we will make use of a variation of the Hidden Markov Model: the Partially Observable Hidden Markov Model.

A keystroke dynamic system measures a person's typing rhythm on a given digital device such as mechanical keyboard, mobile device, touchscreen-based devices, etc and creates a unique signature to identify the legitimate user.

A Hidden Markov Model (HMM) is a doubly stochastic process. It consists of an underlying stochastic process that is not observable (it is hidden), which can only be observed through a different set of stochastic processes. The latter set is also responsible for producing the sequence of observed symbols [3]. The Partially Observable Hidden Markov Model (POHMM) is an extension of the HMM in which the hidden state is conditioned on an independent Markov chain. This structure is motivated by the presence of discrete metadata (such as an event type) which may partially reveal the hidden state; however, the state itself is emanating information from a separate process. The is the exact scenario encountered by the model in keystroke dynamics [4].

In order to evaluate and compare HMM and POHMM, we apply both to address the problems of user identification, verification, and continuous verification, leveraging keystroke dynamics as a behavioral biometric. Identification is performed with the maximum a posteriori (MAP) approach, choosing the model with maximum a posterior probability; verification, a binary classification problem, is achieved by using the model likelihood as a biometric score; lastly, continuous verification is achieved by accumulating the scores within a sliding window over the sequence.

Evaluated on RHU Keystroke dataset [5], POHMM is shown to consistently outperform HMM.

II. Related Works

Over the last 15 years, a large number of research studies have been conducted in Keystroke Biometrics (KB) using different statistical and machine learning approaches. Although very few of them have used HMM as a classifier, having its accuracy and performance much lower than the other methods.

Ali et al. [6] surveyed on KB systems, summarized and compared research that used HMM. The study has found that all the researches have used fixed input, standard QWERTY mechanical keyboard for sample collection, and have used keystroke timing information as features. The authors have drawn conclusions that the accuracy is lower in those KB systems because very few training samples were used to train the models, whereas speech recognition systems achieved better accuracy because plenty of training samples were used to train the model.

In Chen and Chang [7] research, twenty training samples and two hundred testing samples were collected from twenty participants. The authors claimed that their experimental results were nearly perfect among twenty test subjects in a controlled environment. Chang [8] has later extended and improved 30% of their previous result by introducing similarity histograms to find a suitable threshold.

Rodrigues et al. [9] worked on KB system through numerical keyboard using a statistical classifier and HMM. HMM was trained by 30 samples using Baum-Welch algorithm and achieved Equal Error Rate (EER) of 3.6%, thus outperforming the statistical classifier.

Jiang et al. [10] have presented a new approach for web authentication in KB system combining Gaussian model and HMM as a classifier. For training, 870 samples were collected from 58 participants and 3528 imposter samples from 257 participants were collected for testing. The experiment have used forward algorithm to calculate the probability of keystroke sequence of each HMM and achieved EER of 2.54%.

Zhang et al. [11] have proposed a KB authentication system that uses HMM for keystroke sequence analysis and time series to compute the state output probability of HMM. A modified forward algorithm has been used during the authentication phase. The authors have achieved EER lower than 2%.

In regards to POHMM, there has been little published work. By analyzing five different public databases, Monaco and Tappert [4] claimed that the POHMM seem to perform better than other anomaly detectors (including the standard HMM) in biometric identification and verification tasks with EER ranging from 0.6% to 9%. Also, it is generally preferred over the HMM in a Monte Carlo goodness of fit test.

III. Methodology

In this study, we evaluated the HMM and the POHMM using a publicly available keystroke database, the RHU keystroke dataset [5]. The RHU keystroke dataset was chosen since the population composing the database consists of students from different majors, instructors, and merchants in an attempt to diversify the acquisitions and provide a source of results that are as close to real situations as possible. The dataset contains a set of keystroke dynamics consisting of four timings features, each evaluated in milliseconds:

- PP: stores the timing between two key pressures;
- PR: stores the timing between a key pressure and a key release;
- RP: stores the timing between a key release and a key pressure;
- RR: stores the timing between two key releases.

There were 53 subjects (typists) in the dataset, each typing a static 14-character long password string "rhu.university". There have been three data-collection sessions for each subject with at least three day apart between each session. On average, a quantity of five repetitions for the password string was collected in each session, resulting in 985 total acquisitions, averaging the number of acquisitions per user at 17.

The samples that contained errors or a different number of keystrokes were discarded, thus reducing the number of users from 53 to 51. This operation has been carried out by evaluating the number of values in each feature fields. As we mentioned earlier, "rhu.university" is a 14-character long password, so the PR fields was expected to contain 14 values, while the remaining fields (PP, RP, RR) were expected to contain 13.

In order to compare the performance of the HMM and POHMM with existing techniques, we have used the same evaluation methodology followed by Monaco and Tappert [4]. For this reason, we had to compensate for the lack of the 14th values in the PP, RP and RR fields. Thus, we evaluated such values by calculating, for each entry of the dataset, the standard deviation of the existing 13 values in the PP, RP and RR lists, and then added the result to the respective list (at end for RP and RR, and at the beginning for PP).

The following Table displays the growth of the Validation Accuracy for our model based on the training/testing ratio of our model. The ends of the intervals in Table I indicate the acquisition number for each user.

TABLE I
Model Validation Accuracy based on the size of the training and testing sets.

Training interval	Testing interval	ACCURACY
[1, 3)	[3, 16)	0.303681
[1, 4)	[4, 16)	0.381032
[1, 5)	[5, 16)	0.420000
[1, 6)	[6, 16)	0.420000
[1, 7)	[7, 16)	0.492239
[1, 8)	[8, 16)	0.547500
[1, 9)	[9, 16)	0.608571
[1, 10)	[10, 16)	0.613333
[1, 11)	[11, 16)	0.628713
[1, 12)	[12, 16)	0.632000
[1, 13)	[13,16)	0.668874

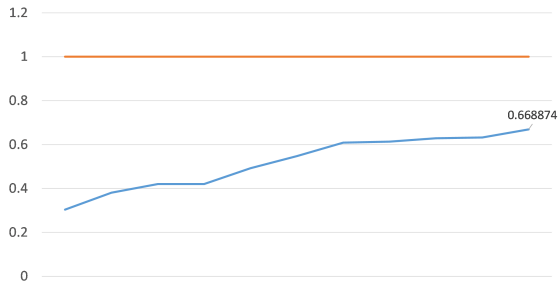

Fig. 1. Model Validation Accuracy growth.

We were satisfied with considering the last entry of Table I for our experimentation, as it splits the dataset exactly with an 80/20 ratio. Thus, for each of the 51 subjects, we used the trials in the interval [1, 13) as the training data. By using these trials, we ran two separate training phases, the first being for HMM and the second for POHMM. Afterwards, the testing was conducted by running the trained HMM and POHMM on the trials in the interval [13, 16) for both genuine and imposter users. The log probability of each event under the model was computed and labeled as the score, and once calculated, the score has been normalized.

A. Identification and Verification

We use HMM and POHMM to perform both user identification and verification.

The identification task, a multiclass classification problem, is performed by the MAP approach in which the model with maximum a posterior probability is chosen as the class label. This approach is typical when using a generative model to perform classification. On the other hand, the verification, which is a binary classification problem, is achieved by comparing the claimed user's model likelihood to a threshold.

The experimentation produced the following evaluation metrics. The identification accuracy (ACC) is measured by the proportion of correctly classified query samples. The verification performance is measured by the user-dependent equal error rate (EER), which is the point on the Receiver Operating

Characteristic (ROC) curve at which the False Rejection Rate (FRR) and False Acceptance Rate (FAR) are equal.

Each query sample is compared against every model in the population, but only one of which will be genuine. The resulting likelihood is normalized using the minimum and maximum likelihoods from every model in the population to obtain a normalized score between 0 and 1. Confidence intervals for both the ACC and EER are obtained over users in each dataset.

B. Continuous Verification

Continuous verification has been recognized as a problem in biometrics in which a resource is continuously monitored to detect the presence of a genuine user or impostor [12]. It is natural to consider the continuous verification of keystroke dynamics, and most behavioral biometrics, since events are continuously generated as the user interacts with the system. In this case, it is desirable to detect an impostor within as few keystrokes as possible.

This task is enforced through a penalty function in which each new keystroke incurs a non-negative penalty within a sliding window. The penalty at any given time can be thought of as a degree of suspicion. As the behavior becomes more consistent with the model, the cumulative penalty within the window can decrease, and as it becomes more dissimilar, the penalty increases. The user is rejected if the cumulative penalty within the sliding window exceeds a certain threshold. This is chosen for each sample such that the genuine user is never rejected, analogous to a 0% FRR in static verification.

The performance is reported as the number of events (up to the sample length) that can occur before an impostor is detected. This is determined by increasing the penalty threshold until the genuine user is never rejected by the system. Since the genuine user's penalty is always below the threshold, this is the maximum number of events that an impostor can execute before being rejected by the system, while also assuring that the genuine user is never rejected. The average maximum rejection time (AMRT) is considered as a good evaluation metric. The maximum rejection time (MRT) is the maximum number of keystrokes needed to detect an impostor without rejecting the genuine user. The MRT is determined for each combination of impostor query sample and user model in the dataset in order to compute the AMRT.

IV. Experimental Results

Over the course of the experimentation phase, we first took into consideration some of the possibilities offered by the nature of the RHU database. More precisely, having a modest number of entries and only four features per user, the RHU database allowed us to quickly evaluate the most efficient feature configuration for POHMM. Table II displays every possible subset of features and the corresponding identification ACC value.

Given these results, we decided to run our tests using all four features and see how POHMM performs in relation to identification, verification, and continuous verification. For each evaluation metric produced, we provide the mean value, standard deviation and quartiles.

TABLE II
Feature-dependant ACC using POHMM.

PP	PR	RP	RR	ACC
x	x	x	x	0.708609
x	x		x	0.701987
x	x			0.701987
	x	x		0.668874
x	x	x		0.662252
	x		x	0.615894
	x	x		0.615894
		x	x	0.609272
x		x	x	0.602649
x			x	0.602649
			x	0.602649
x		x		0.589404
		x		0.562914
x				0.496689
	x			0.119205

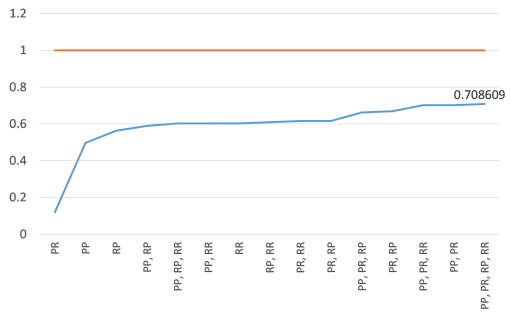

Fig. 2. Accuracy Growth based on features picked.

A. Identification and Verification

We evaluated the identification and verification performances of the model by using ACC, EER and ROC curve, in order to obtain:

- Session Accuracy (S-ACC);
- User Accuracy (U-ACC);
- Session Equal Error Rate (S-EER);
- User Equal Error Rate (U-EER);
- Area Under ROC Curve (AUC).

The results are shown in Table III.

TABLE III
Experimental results for the Identification and Verification tasks using POHMM.

	S-ACC	U-ACC	S-EER	U-EER	AUC
count	1.00000	51.00000	1.00000	51.00000	1.00000
mean	0.70861	0.71242	0.05338	0.03114	0.01521
std		0.33346		0.06547	
min	0.70861	0.00000	0.05338	0.00000	0.01521
25%	0.70861	0.66667	0.05338	0.00000	0.01521
50%	0.70861	0.66667	0.05338	0.00676	0.01521
75%	0.70861	1.00000	0.05338	0.02703	0.01521
max	0.70861	1.00000	0.05338	0.33333	0.01521

Figure 3 displays the Threshold, FAR and FRR curves, which are required to determine the EER.

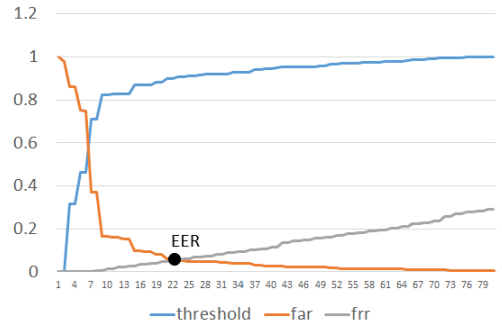

Fig. 3. Threshold, FAR and FRR curves.

Despite the limited number of entries per user, POHMM manages to perform surprisingly well with both tasks.

B. Continuous Verification

We evaluated the continuous verification performances of the model by using ACC and MRT, in order to obtain:

- Continuous Identification Accuracy (CIA);
- Average Maximum Rejection Time (AMRT).

The results are shown in Table IV.

TABLE IV
EXPERIMENTAL RESULTS FOR THE CONTINUOUS VERIFICATION TASK USING POHMM.

	CIA	AMRT
count	151.000000	151.000000
mean	0.216651	3.880000
std	0.204942	2.411958
min	0.000000	0.040000
25%	0.071429	2.100000
50%	0.142857	3.840000
75%	0.285714	5.090000
max	0.928571	12.520000

POHMM does not perform too well if trained on a limited number of features and is applied to address the problems of user continuous verification [4]. Our experiment shows very clearly how inconsistent the model can be on this specific setup, thus resulting in low average CIA and high MRT.

C. Comparing POHMM and HMM

Since POHMM is an extension of HMM, it is possible to create an instance, train, and test the latter by imposing restrictive rules to POHMM. We were interested in comparing the two models on the verification, identification and continuous verification tasks.

The Tables ranging from V to XI compare the results between POHMM and HMM for each standard biometric security system algorithm.

TABLE V
SESSION ACCURACY COMPARISON.

	POHMM	HMM
count	1.0000	1.0000
mean	0.7086	0.3179
min	0.7086	0.3179
25%	0.7086	0.3179
50%	0.7086	0.3179
75%	0.7086	0.3179
max	0.7086	0.3179

TABLE VI
USER ACCURACY COMPARISON.

	POHMM	HMM
count	51.0000	51.0000
mean	0.7124	0.3137
std	0.3335	0.4132
min	0.0000	0.0000
25%	0.6667	0.0000
50%	0.6667	0.0000
75%	1.0000	0.6667
max	1.0000	1.0000

TABLE VII
SESSION EQUAL ERROR RATE COMPARISON.

	POHMM	HMM
count	1.0000	1.0000
mean	0.0534	0.1609
min	0.0534	0.1609
25%	0.0534	0.1609
50%	0.0534	0.1609
75%	0.0534	0.1609
max	0.0534	0.1609

TABLE VIII
USER EQUAL ERROR RATE COMPARISON.

	POHMM	HMM
count	51.0000	51.0000
mean	0.0311	0.1306
std	0.0655	0.1373
min	0.0000	0.0000
25%	0.0000	0.0203
50%	0.0068	0.0743
75%	0.0270	0.2061
max	0.3333	0.6667

TABLE IX
AREA UNDER CURVE COMPARISON.

	POHMM	HMM
count	1.0000	1.0000
mean	0.0152	0.0907
min	0.0152	0.0907
25%	0.0152	0.0907
50%	0.0152	0.0907
75%	0.0152	0.0907
max	0.0152	0.0907

TABLE X
CONTINUOUS IDENTIFICATION ACCURACY COMPARISON.

	POHMM	HMM
count	151.0000	151.0000
mean	0.2167	0.1235
std	0.2049	0.1692
min	0.0000	0.0000
25%	0.0714	0.0000
50%	0.1429	0.0714
75%	0.2857	0.1786
max	0.9286	0.7143

TABLE XI
AVERAGE MAXIMUM REJECTION TIME COMPARISON.

	POHMM	HMM
count	151.0000	151.0000
mean	3.8800	6.3358
std	2.4120	3.0420
min	0.0400	0.1400
25%	2.1000	4.6400
50%	3.8400	6.4200
75%	5.0900	7.9200
max	12.5200	13.7400

Our results confirm those in [4], furtherly confirming that POHMM outshines its original iteration.

V. CONCLUSIONS

Overall, POHMM has been proven to be really effective for both identification and verification, despite the limited number of evaluated features, and also a slim training set. Furthermore, it managed to exceptionally outclass HMM, when applied to the same dataset. On the other hand, neither POHMM or HMM met the performance standards in regard to continuous verification.

REFERENCES

[1] K. Bonnen, B. F. Klare, and A. K. Jain, "Component-based representation in automated face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 239–253, 2013.

[2] S. Hocquet, J.-Y. Ramel, and H. Cardot, "User classification for keystroke dynamics authentication," in *Advances in Biometrics*, S.-W. Lee and S. Z. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 531–539.

[3] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[4] J. V. Monaco and C. C. Tappert, "The partially observable hidden markov model and its application to keystroke dynamics," 2017.

[5] M. El-Abed, M. Dafer, and R. E. Khayat, "Rhu keystroke: A mobile-based benchmark for keystroke dynamics systems," in *2014 International Carnahan Conference on Security Technology (ICCST)*, 2014, pp. 1–4.

[6] M. Ali, V. Monaco, and C. Tappert, "Hidden markov models in keystroke dynamics," 05 2015.

[7] W. Chen and W. Chang, "Applying hidden markov models to keystroke pattern analysis for password verification," in *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.*, 2004, pp. 467–474.

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

[9] R. N. Rodrigues, G. F. G. Yared, C. R. do N. 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*, D. Zhang and A. K. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 640–646.

[10] C.-H. Jiang, S. Shieh, and J.-C. Liu, "Keystroke statistical learning model for web authentication," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '07. New York, NY, USA: Association for Computing Machinery, 2007, pp. 359–361. [Online]. Available: https://doi.org/10.1145/1229285.1229327

[11] Y. Zhang, G. Chang, L. Liu, and J. Jia, "Authenticating user's keystroke based on statistical models," in *2010 Fourth International Conference on Genetic and Evolutionary Computing*, 2010, pp. 578–581.

[12] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar, "Continuous verification using multimodal biometrics," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, pp. 687–700, 05 2007.