Keystroke Biometric studies with Hidden Markov Model and Its Extension on Short Fixed-Text Input

Md Liakat Ali
School of Business and Computer Science
Caldwell University
New Jersey, USA
mali@caldwell.edu

John V. Monaco
Computational and information Sciences Directorate
U.S. Army Research Laboratory
Aberdeen Proving Ground, MD, USA
john.v.monaco2.civ@mail.mil

Charles C. Tappert
Seidenberg School of Computer Science and Information Systems
Pace University
New York, USA
ctappert@pace.edu

***Abstract*—The hidden Markov model (HMM) and its extensions have been applied in numerous scientific and engineering areas. In speech recognition, HMMs still outperform many other models. HMMs have also demonstrated significant performance in signature and gesture recognition. Nonetheless, the performance of HMM in Keystroke Biometric (KB) systems is typically lower when compared to other biometric systems. Moreover, there seems to be limited research conducted in Keystroke Biometrics using HMMs. This study discusses the hidden Markov model and its various extensions used in biometric areas. It also evaluates the hidden Markov model and the recently proposed partially observable hidden Markov model (POHMM) on a benchmark KB dataset. The POHMM, an extension of HMM that conditions the hidden state on an independent Markov chain, achieves a 0.045 equal-error rate (ERR), a significant performance improvement over the HMM and other leading methods in user verification.**

***Keywords*—partially observable hidden Markov mode; anomaly detector; verification; keystroke biometrics**

I. Introduction

In modern life, we heavily depend on computers to store and process our sensitive information. However, intruders are everywhere, and they attack an individual's information as well as companies' databases. As a result, securing the information from intruders has become a burning question as well as creating the necessity for developing a foolproof measure against unauthorized access of information. User authentication is the most important and challenging issue to control unauthorized access during the access control step. Authentication is a process to verify the right person by matching some pre-arranged identity, which guarantees that somebody shares data with the right person, and only that authorized person can access that data.

Keystroke Dynamics are one type of behavioral biometrics used to recognize a user's unique typing pattern. It measures a person's typing pattern on some digital devices and creates a unique profile to identify the legitimate user. A unique keystroke signature, or profile, for an individual can be constructed from a person's typing speed, the duration between successive keys presses, pressure applied on the keys, and fingers position on the keys. User recognition based on this unique typing rhythm is non-intrusive, offers continuous monitoring as the user types, and can be both a transparent and noninvasive authentication method for users. Moreover, it is very easy to capture data, since it requires only a standard desktop or laptop keyboard [1].

Although HMMs have been widely used in many biometric systems, especially in speech recognition, researchers have paid little attention of using HMM in keystroke biometric systems. Chen and Chang [2] have used HMM for the first time in keystroke dynamics in the year of 2004 and Chang has extended their work using histogram of the similarity measured as part of HMM approach to find suitable threshold [3]. Rodrigues et al. [4] introduced a new approach in keystroke dynamics through numerical keyboard. The experiment was conducted by using a statistical classifier and HMM, where the best results were achieved by the HMM.

Vuyyuru *et al.* [5] have proposed a method using modified HMM which can dynamically add or remove users and can adapt to the changing typing pattern of the users. For each user, a distinct HMM was developed by using a modified Rabiner's re-estimation formulae of multiple observation sequences on the reference keystroke patterns. Their experiment found a best false accept rate of 0.74%, false reject rate of 8.06%, and 3.04% *equal error rate* (EER). Jiang, Shieh, and Liu [6] worked on a statistical model for web authentication using HMM and Gaussian modeling. The author claimed that their proposed system is able to enhance the security in web authentication mechanisms and got best error rate of 2.54%. Another study conducted by Zhang, Chang, and Jia [7] achieved an EER of less than 2% by using the HMM with a modified forward algorithm to score users' typing behavior in authentication phase. Monaco and Tappert [8] have introduced an extension of the HMM. The experiment was conducted using several datasets such as Password, Keypad, Mobile, Free-text, and Fixed-text. The best performance of 0.6% EER was achieved on mobile keyboard input with sensors.

The paper organization is as follows: section II describes the hidden Markov model and its extensions, section III describes experiment methodology, section IV discusses experimental results and finally section V is the conclusion and suggestion for future work.

II. Hidden Markov model and its extensions

A Markov model is a stochastic process in which the future state of the system depends on only the present state. Different types of Markov models have been used depending on the

978-1-5386-1104-3/17/$31.00 ©2017 IEEE

system's behavior and environment. If the system is autonomous and is fully observable, then the Markov chain is used. If the system is autonomous but only partially observable, then the hidden Markov model is used. The Markov decision process is used when the system is controlled and fully observable. In the Markov decision process, state transitions depend on the current state and an action vector applied to the system. A partially observable Markov decision process is used when the system is controlled but partially observable [9].

A. Hidden Markov Model

The *hidden Markov model* (HMM) [10] is a generative model which has been widely used in biometric areas such as speech, gesture, handwriting, and recently in keystroke dynamics. The HMM is a finite state model that consists of a fixed number of discrete hidden states and the observed values. The emission at time t depends on the underlying latent state. Let $O = o_1, \ldots, o_T$ be the complete sequence of observation vectors (emission vectors) from times 0 to T, the total number of observation is N, and the corresponding hidden state sequence, $\theta = \theta_1, \ldots, \theta_T$ while M is the total number of hidden states. A feature vector is observed at the nth time step, t_n. The model starts in the initial state j at time $t = 0$ with probability π_j. At each consequent time instance t, the model transitions into a new state θ_t with the transition probability a_{ij}. The system usually proceeds with a first-order dependency between the hidden states in discrete steps. Fig. 1 shows the graphical representation of the HMM. The model parameters, $\lambda = (\pi, \boldsymbol{A}, \boldsymbol{B})$, associated with a HMM is given by:

1. *Initial state distribution*, $\pi = [\pi_j]$ is the starting probability vector.

2. *State transition probability matrix*, $A = [a_{ij}]$.

3. *State emission probability distributions*, $\mathbf{B} = [b_j(.)]$. At time t_n and in state j, the system emits an observation vector $\boldsymbol{o_n}$ according to the density function $b_j(.)$ parameterized by vector $\boldsymbol{b_j}$.

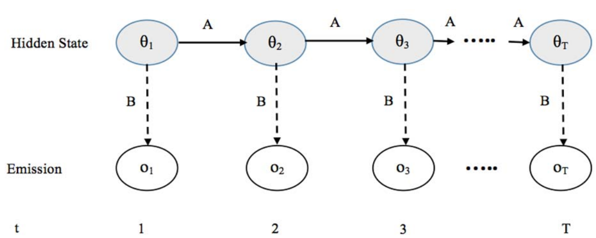

Fig. 1. Graphical representation of HMM, redrawn from [10]

Given an HMM model, there are three basic problems that need to be solved for the model to be useful in practical applications (Rabiner 1989). *Likelihood Calculation*- determine $P(O|\lambda)$, the likelihood of an observation sequence O and HMM model parameters λ. *Inferring the hidden state sequence*- given the model parameters λ and observation sequence O; determine the maximum likelihood sequence of hidden states θ. *Parameter estimation*- determine arg $\max_{\lambda\epsilon\Theta} P(O|\lambda)$, the maximum likelihood parameters λ, given the observation sequence O. Likelihood calculation and parameters estimation are necessary for identification and verification purpose in biometric applications while the determination of maximum likelihood sequence of hidden states provides insight to user behavior.

1) Solution to problem 1: Model likelihood

The goal is to compute the likelihood $P(O|\lambda)$ of the observation sequence O given the model λ. The most straightforward way to find the likelihood is Naïve way by enumerating every possible state sequence of length N. This approach is computationally unfeasible because it requires summing over all possible state paths. The order of computations required for this procedure is $O(2N.\ M^N)$ where N is the number of observations and M is the number of states [10]. However, there exists a computationally simpler algorithm called forward-backward procedure to compute $P(O|\lambda)$. We only need the forward part of the procedure to find the model likelihood. Consider the forward variable $\alpha_j(n)$ is the probability of the partial observation sequence O from time 1 to n with observation o_n at time t_n being in state j, and given the model parameters λ. The forward variable $\alpha_j(n)$ can be efficiently computed inductively by follows:

a) Initialization: $\alpha_j(1) = \pi_j b_j(o_1)$
b) Induction: $\alpha_j(n+1) = (\sum_{i=1}^{M} \alpha_i(n) a_{ij}]\, b_j(o_{n+1})$
c) Termination: $P(O|\lambda) = \sum_{j=1}^{M} \alpha_j(N)$

Similar to forward variable consider the backward variable $\beta_j(n)$ is the probability of the partial observation sequence O from time $n+1$ to T, and state j at time t_n and given the model parameters λ. The backward variable $\beta_j(n)$ can be efficiently computed inductively by follows:

a) Initialization**:** $\beta_j(N) = 1$
b) Induction: $\beta_j(n) = (\sum_{j=1}^{M} a_{ij}\, b_j(o_{n+1})\beta_j(n+1)$
c) Termination: $P(O|\lambda) = \sum_{j=1}^{M} \beta_j(1)\pi_j$

Total computation involved in the calculation of the forward and the backward algorithm is $O(M^2N)$.

2) Solution to problem 2: Infering hidden states

The goal is to find most likely sequence of hidden state $\theta = (\theta_1, \theta_2, \ldots \theta_T)$ that is optimal, given the observation sequence O and parameters λ. The most widely used criterion is to find the single best state sequence path, i.e., maximize $P(\theta|O,\lambda)$ which is equivalent to maximizing the posterior $P(\theta, O|\lambda)$. The well known Viterbi algorithm is a formal technique for finding this single best state sequence which is based on dynamic programming methods.

Given observation sequence and the model parameters, the posterior probability of being in state j at time t_n can be calculated using the forward variable $\alpha_j(n)$ and backward variable $\beta_j(n)$. To implement the solution we defined the forward-backward variable $\gamma_j(n)$,

$$\gamma_j(n) = \frac{\alpha_j(n)\beta_j(n)}{P(O|\lambda)} = \frac{\alpha_j(n)\beta_j(n)}{\sum_{i=1}^{M} \alpha_i(n)\beta_i(n)} \qquad (1)$$

$$where\ 1 \le n \le N$$

Using the $\gamma_j(n)$, the most likely hidden state at time t_n can be calculated by

$$\theta_n = \arg max_{1\le j\le M}\, \gamma_j(n) \qquad (2)$$

3) Solution to problem 3: Parameter estimation

Problem 3, parameter estimation, is the most difficult and important problem of HMM's which is to determine $arg\,max_{\lambda\in\Theta}P(O|\lambda)$ the maximum likelihood parameters, given observed value *O*. To solve this problem requires an estimate of the starting probabilities, transition probabilities, and observation distribution parameters. In order to re-estimate the HMM parameters, let $\xi_{ij}(n)$ be the transitioning probability from state *i* at time t_n to state *j* at time t_{n+1} given model $\lambda = (A, B, \pi)$. From the definition of forward-backward variable

$$\xi_{ij}(n) = \frac{\alpha_i(n)\alpha_{ij}b_j(o_{n+1})\beta_j(n+1)}{P(O|\lambda)} = \frac{\alpha_i(n)\alpha_{ij}b_j(o_{n+1})\beta_j(n+1)}{\sum_{k=1}^{M}\alpha_k(n)\beta_k(n)} \quad (3)$$

The forward-backward variable $\gamma_i(n)$ is the probability of being in state *j* at time t_n, given the observation sequence and the model parameters λ. $\xi_{ij}(n)$ is the probability of transitioning from state *i* at time t_n to state *j* at time t_{n+1}. So, $\gamma_i(n)$ can relate to $\xi_{ij}(n)$ by summing over t_n, giving

$$\gamma_i(n) = \sum_{j=1}^{M}\xi_{ij}(n) \quad (4)$$

If we sum $\xi_{ij}(n)$ over the time index t_n , we get the expected number of transitions from *i* to *j*, equivalently, If we sum $\gamma_j(n)$ over the time t_n , we get the expected number of transitions from state *j*.

$$\sum_{n=1}^{N-1}\xi_{ij}(n) = expected\ number\ of\ transition\ from\ i\ to\ j \quad (5)$$

$$\sum_{n=1}^{N-1}\gamma_j(n) = expected\ number\ of\ transition\ from\ \ j \quad (6)$$

Using the variables $\gamma_j(n)$ and $\xi_{ij}(n)$ from the above formula, we can update the parameters. The re-estimation formula for π, ***A*** and ***B*** are

$$New\ starting\ probabilities\ determined\ by, \quad \bar{\pi}_j = \gamma_j(1) \quad (7)$$

$$Transition\ matrix\ updated\ by, \quad \bar{a}_{ij} = \frac{\sum_{n=1}^{N-1}\xi_{ij}(n)}{\sum_{n=1}^{N-1}\gamma_i(n)} \quad (8)$$

$$updated\ state\ probabilites, \quad \bar{\Pi}_j = \frac{\sum_{n=1}^{N}\gamma_j(n)}{\sum_{i=1}^{M}\sum_{n=1}^{N}\gamma_i(n)} \quad (9)$$

The re-estimates for parameter vectors $\boldsymbol{b}_j$ can be determined by

$$\bar{\boldsymbol{b}_j} = \arg max_{\boldsymbol{b}\epsilon b(.)}\sum_{n=1}^{N}\gamma_j(n)\ln b(o_n) \quad (10)$$

Using above formula the complete parameter estimation which is also known as Baum-Welch re-estimation algorithm is describes as follows [7]:

a) Initialization: find initial parameters $\acute{\lambda}$ and let $\bar{\lambda} \leftarrow \acute{\lambda}$
b) Expectation: Compute $\alpha_j(n)$, $\beta_j(n)$, $\gamma_j(n)$ and $\xi_{ij}(n)$. Let $\bar{P} \leftarrow P(O|\lambda)$
c) Maximization: Using re-estimation formula, update the model parameter π, ***A*** and ***B***, and let $\bar{\lambda} \leftarrow (\bar{\pi}, \bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$
d) Termination: If $P(O|\bar{\lambda}) - \bar{P} < \varepsilon$ then terminate and let $\bar{\bar{\lambda}} \leftarrow \bar{\lambda}$, otherwise go to step (b). ε is the convergence criterion threshold.

B. Partly-HMM

The *Partly-hidden Markov model* (PHMM) proposed by Kobayashi and Haruyama [11] in the application of gesture recognition. Later Iobayasi, Furuyama, and Masumitsu [12] have used the model in speech recognition. PHMM is a modified second order Markov model in which the first state is hidden, and the second one is observable. The difference between HMM and PHMM is that, in PHMM, both hidden state and observation state are dependent on the previous observation state. Therefore, it can be used in the applications that have a transient underlying process such as gesture and voice recognition. *Expectation maximization* (EM) algorithm or segmental k-means algorithm are used to train PHMM for parameters estimation. The forward algorithm and Viterbi algorithm are used to get likelihood for PHMM similar to HMM. As PHMM has more model parameters, it is expected that PHMM needs more training data to achieved better performance.

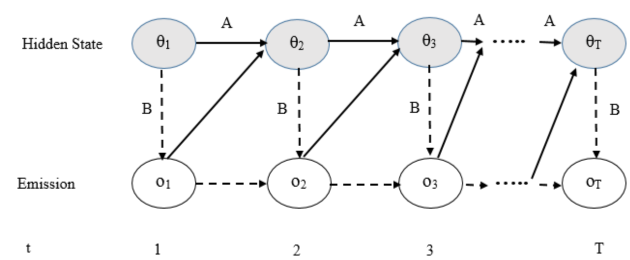

Fig. 2. Structure of partly-HMM, redrawn from [12]

Fig. 2 shows the structure of partly-HMM. In HMM, the observation state is dependent only on the current hidden state, and state transition or hidden state dependent only on the previous state. In PHMM, observation state is dependent on the current hidden state and previous observation state, and the hidden state is defined by the previous hidden state and previous observation. Partly hidden Markov model has been applied in gesture recognition and speech recognition, and performed better than the hidden Markov model in both cases. The authors have suggested that as partly hidden Markov model has more parameters than HMM, so it is expected that PHMM require more training data for better performance.

C. Hidden Semi-Markov Model (HSMM)

A *hidden semi-Markov model* (HSMM) is an extension of HMM, originally proposed by Ferguson [13] which is partially included in the survey paper by Rabiner [10]. HSMM has also been called by some other names such as "*HMM with explicit duration*" [14], "*segmental HMM*"[15], "*explicit duration HMM*" [13], "*variable-duration HMM*"[16], "*generalized HMM*" [17], and "*segment model*" [18]. A semi-Markov model is similar to an HMM except that each hidden state can emits a sequence of observations with explicit duration. The state duration probability distribution are denoted by $D = \{p_i(d)\}, p_i(d) = P(\tau_t = d|\theta_t = \theta_i)$, where τ_t is the residual time of the current state θ_t before time *t*. Fig. 3 shows the graphical presentation of general HSMM.

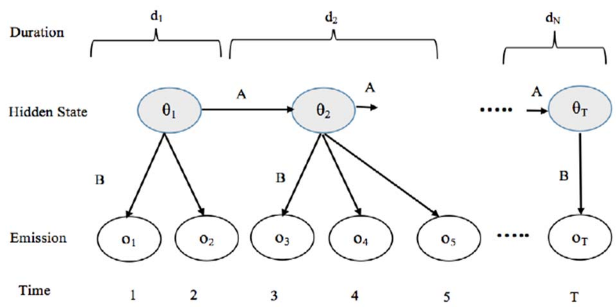

Fig. 3. Structure of hidden semi-Markov model, redrawn from [19]

The state duration distributions and the observation distributions can be either parametric or non-parametric depending on the specific application. The observations in HSMM are usually assumed to be dependent on the hidden states that produce them and conditionally independent to each other for the given state. An exception is the segmental model, which assumes the observations depend not only on the observation state but also on the state duration. Some variants of HSMM are switching HSMM, multi-channel HSMM, and adaptive HSMM. Hidden semi-Markov models and its variants have been applied in many areas and the major applications include speech recognition, speech synthesis, printed text recognition, electrocardiograph, recognition of human genes in DNA, language identification, semantic learning, remote sensing, financial time series modeling, human activity recognition, handwriting recognition, network traffic modeling and anomaly detection [19].

D. Partially-HMM

Partially-HMM is an extension of HMM proposed by Ozkan, Akman, and Kozat [20]. Beside observation states, the model also provides partial and noisy access to the hidden state sequence as side information. As the hidden state sequence is partially observed thorough side information, the authors categorized their model as *partially hidden Markov model* (PHMM). Fig. 4 shows the structural representation of partially-HMM. At every time instant *t*, the model observes the hidden state θ_t as x_t with probability $\flat$ and hence, the probability that the state stays hidden is 1- $\flat$. Like HMM, the parameters of partially-HMM can be estimated by the modified EM algorithms for considering partial and noisy access to the hidden state sequence as side information.

In HMM, the state observations are not essentially limited to a time interval; it may also be randomly distributed along the complete time span of the application. For example, if $\flat$ is equal to zero, then HMM will have no state observation. In this case, partially-HMM model provides a generalized framework by allowing partial access to the state sequence. In case of state observation corrupted with noise, partially-HMM also provides a new set of iterative EM equations that incorporates the side information and estimates the model parameters accordingly. Partially-HMM uses both labeled and unlabeled data to train the model similar to semi-supervised learning. Biometric applications such as in speech processing, there may have limited number of labeled data with large number of unlabeled data. Partially-HMM is appropriate model in this scenario to handle large number of unlabeled data. The partial information may also help in parameters estimation in the scenario of missing or incorrect labeling. The research [20] has claimed that using the side information, state recognition performance was improved by 70%, and the model has shown to be robust to the training condition.

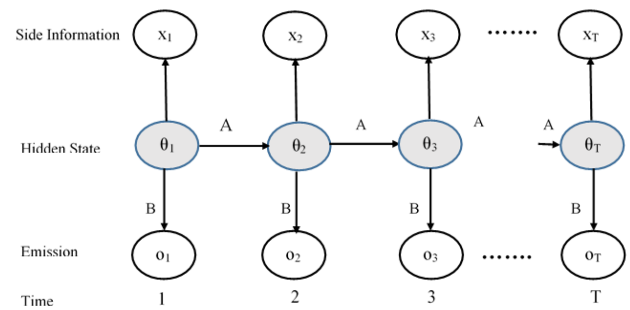

Fig. 4. Structure of partially-HMM, redrawn from [20]

E. Context-based Partially Hidden Markov Model

Context-based partially hidden Markov model has been proposed by Forchhammer and Rissanen [21] and its extension has been applied in image coding [22]. Context-based PHMM combines hidden states and the contexts specified by the past states. The difference between HMM and context-based PHMM is that both hidden state and observation state are conditioned on the past observations. Past observations provide context (Hidden state context, observed context) for the transition and observation state in this model. A generalized version of Baum-Welch re-estimation procedure has been used for parameter re-estimation in context-based PHMM. A "Context" is useful particularly in image data modeling. The conditioning of the probabilities in context PHMM's may also be useful in speech recognition and machine reading of handwritten text.

Fig. 5 shows the general structure of context PHMM. At every time step, the hidden state and emission are conditioned on the context from the previous step. The first version of context-based PHMM has introduced second dimension in the HMM model by involving a one dimensional sequence of hidden states and context conditioned on model probabilities possibly two dimensional sequence [21]. This model is adaptive because the model parameters are trained or re-estimated on the whole dataset. The second version of context-based PHMM [22] has extended the adaptability. Context-based PHMM model adapt data by pruning the context tree, and by introducing local adaptability. This adaptive coding scheme helps to avoid the need for explicitly coding the parameters as in the hidden Markov model.

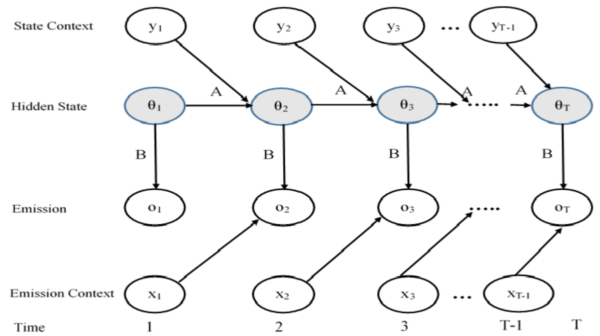

Fig. 5. General structure of context PHMM, redrawn from [22]

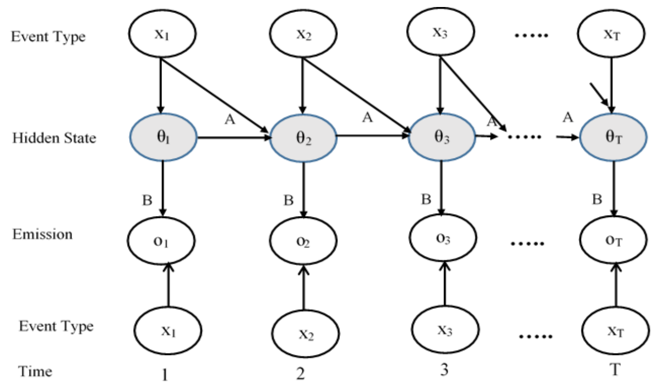

Fig. 6. Graphical presentation of POHMM, redrawn from [7]

F. *Partially Observable Hidden Markov Model(POHMM)*

Recently, Monaco and Tappert [7] have proposed a generalization of the hidden Markov model named *partially observable hidden Markov model* (POHMM), in which the emission state depends on hidden state and partially observed state, state transition is dependent on previous hidden state and on partially observed state. An important characteristic of POHMM is the event types that revel partial information about the hidden state. For example, the event types are the keyboard key names in the keystroke biometrics that revel partial information about the hidden state. In a two-state model of typing behavior, users can be either in active state or in passive state. The posterior probability of occurrence in either of these states greatly depends on which key was pressed. Such as, there is a greater possibility that the system is in passive state while user type "*Space*" key. Because, when user type "*Space*" key, they usually pauses between words and sentences. Fig. 6 shows the graphical presentation of POHMM.

POHMM follows the same algorithms and techniques used in HMM to determine model likelihood, predict hidden state and parameter estimation. Beside these, marginal distributions of the model act as fallback mechanism when new event types are faced during likelihood estimation. The model also describes a parameter smoothing technique that handles missing data during parameter estimation and at the same time reduces the degrees of freedom of the model to avoid overfitting. In short, marginal distributions handle missing or novel data during likelihood calculation and parameter smoothing handles missing or infrequent data during parameter estimation.

The most important features of POHMM in parameter estimation are initialization, marginal distributions, and parameter smoothing. In POHMM, fixed parameter initialization approach has been used which reproduces the equal probabilities in the absence of any event x in X. Fallback mechanism [23] have been used in keystroke biometric to handle missing during training time and handle new data during the testing time. The marginal distribution have been used in POHMM to handle missing, new or unusual events during model likelihood calculation which generates two-level fallback order in which event type is marginalized out. Parameter smoothing techniques in POHMM handle missing or infrequent data during parameter estimation. Parameter smoothing helps to reduce overfitting and provides excellent estimates in missing or infrequent data.

III. Methodology

In this study we have evaluated the hidden Markov model and the partially observable hidden Markov model using CMU keystroke benchmark dataset [24]. The CMU dataset was chosen for evaluation since there exist a number of keystroke dynamics algorithms for objective comparisons. The authors have evaluated the datasets with fourteen existing keystroke dynamics classifiers. The dataset contains keystroke dynamics consisting of the dwell time for each key as well as the flight time between two successive keys. There were 51 subjects (typists) in the dataset, each typing a static password string "*.tie5Roanl*". The dataset also considers the "*Enter*" key to be a part of the password constructing the 10-character password to become 11 keystrokes long. There were eight data-collection sessions for each subject with at least one day apart between each two-session period. A quantity of 50 repetitions for the password string was collected in each session, resulting in 400 samples for each subject.

In order to compare the performance of the HMM and POHMM with existing techniques, we have used the same evaluation methodology followed by Killourhy and Maxion [24]. Each of the 51 users are considered genuine users and the remaining 50 users are considered as imposters. For each of the 51 subjects, we used the first 200 feature vectors (Sessions 1-4, samples 1-200) as the training data. We ran the training phase of the HMM and POHMM for the first 200 password repetitions typed by the genuine user. Sessions 5-8 were used for testing for both genuine and imposter users. The testing module was conducted in two steps. First, we ran the test phase of HMM and POHMM on the samples 201-400 repetitions typed by the genuine user. Anomaly scores were assigned to the each timing vector as user scores. Secondly, we ran the test phase of HMM and POHMM on the timing vectors from the first five repetitions typed by each of the 50 imposters and anomaly scores were assigned to each timing vector, labeled as the imposter score.

The POHMM conditions the hidden state on an independent Markov chain which is assumed to be the sequence of keys pressed. However, the independent Markov chain can also be a function of the keystroke sequence, such as the keyboard row that contains the key, or whether the key is a consonant or vowel. We evaluated four different ways of specifying the event type:

- *Left/right*: the side of the keyboard that the key is in (2 unique event types).

- *Keyboard row*: the row on the keyboard that the key is in (3 unique event types).
- *Cons/vowel*: whether the key is a consonant, vowel, or something else (3 unique event types).
- *Key name*: the name of the key, ".tie5Roanl"+"Enter" (11 unique event types).

The most common and widely accepted techniques for determining threshold in a detector-independent way and summarizing performance are equal error rate and zero-miss false-alarm rate. The *receiver operating characteristic* (ROC) curve is also a common technique to visualize a detector's accuracy. Various measures of error such as false acceptance rate and false rejection rate can be derived through ROC curve. To calculate the EER and ROC, the threshold is chosen at a point where the probabilities of false acceptance and false rejection rate are equal.

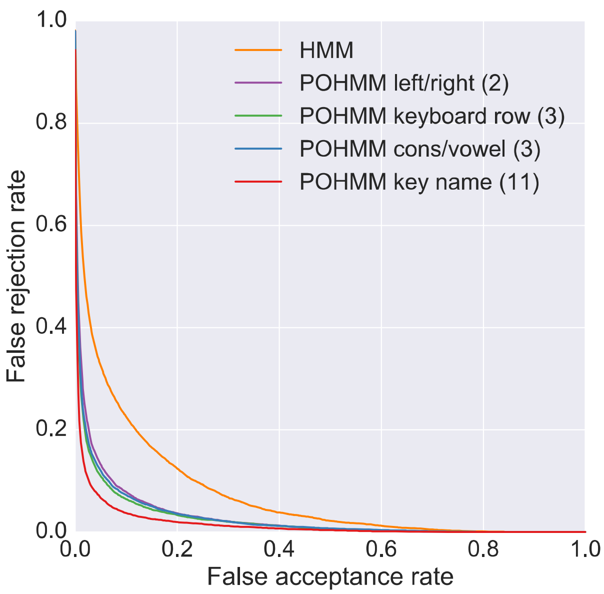

Fig. 7. ROC curve depicts the performance of the HMM on CMU dataset

IV. Experimental Results

Biometric authentication systems mainly perform two functions: verification and identification. User identification, also known as user recognition, requires the system to classify the input pattern into one of the *N* known classes. In the verification stage, the KB system either accepts a claim of ownership to a particular sample or rejects it as an imposter. When verification is used as an access control mechanism, it is typically referred to as authentication. In this experiment, we have used two models: the HMM and POHMM for verification purposes. We have trained the HMM and POHMM using the CMU benchmark datasets to create user profiles. Key press-press latency and hold time were used as features, modeled by a log-normal emission distribution. In the verification phase, when a user's typing pattern is provided to the HMM and POHMM, the likelihood of the claimant user's profile is created and normalized by the minimum and maximum likelihoods of other profiles in the system. Finally, the normalized score is compared with a threshold to accept or reject the claimant.

TABLE I. Comparison of HMM-based Researches in KB Systems

Detector	EER (std. dev.)
POHMM key name (11 event types) [7]	**0.045 (0.053)**
POHMM keyboard row (3 event types) [7]	**0.063 (0.064)**
POHMM cons/vowel (3 event types) [7]	**0.066 (0.066)**
POHMM left/right (2 event types) [7]	**0.069 (0.065)**
Gaussian mixture model [25]	0.087 (0.058)
Manhattan (scaled) [24]	0.096 (0.069)
Nearest Neighbor (Mahalanobis) [24]	0.100 (0.064)
Outlier Count (z-score) [24]	0.102 (0.077)
Support Vector Machine (one-class) [24]	0.103 (0.065)
Mahalanobis [24]	0.110 (0.065)
Mahalanobis (normed) [24]	0.110 (0.065)
Hidden Markov model (HMM)	**0.131 (0.097)**
Manhattan (filter) [24]	0.136 (0.083)
Manhattan [24]	0.153 (0.093)
Neural Network (auto-assoc) [24]	0.161 (0.080)
Euclidean [24]	0.171 (0.095)
Fuzzy Logic [24]	0.215 (0.119)
k Mean [24]	0.221 (0.105)
Neural Network (standard) [24]	0.372 (0.139)

To compare performance, we have measured the mean and standard deviation of equal error rate of the HMM and POHMM. The training and testing procedure for HMM and POHMM are exactly the same except the POHMM uses the key sequence (or a function of the key sequence) as event types that condition the hidden state. Using the same evaluation methodology described in [24], HMM has achieved average EER of 0.131 with standard deviation 0.097 where POHMM has achieved average EER of 0.045 with standard deviation 0.053 using key names as event types. Fig. 8 shows the ROC curve obtained for CMU dataset using the HMM and each of the POHMMs respectively.

Table I shows the performance comparison of HMM and POHMM along with other anomaly detectors. From the table, it is obvious that the performance of hidden Markov model is poorer compared to several other anomaly detectors in the list while POHMM outperforms all of the anomaly detectors considered. Using the key name as event type provides the best performance, although this method also results in the most model parameters since there are 11 unique event types. Even using just 2 unique event types (left/right), the POHMM achieves an EER about half of the HMM.

V. Conclusion

The generative model and discriminative model are the two categories of statistical models used in keystroke biometric areas. Generative approach models the underlying distribution of the classes while discriminative approaches focus only on learning the class boundary. Both of these approaches have been widely used in the area of biometrics, including keystroke biometrics. In our experiment, we have used two generative models HMM and POHMM on the CMU benchmark datasets for user authentication. The experimental result shows that the POHMM outperformed several generative and discriminative models on CMU short fixed-text dataset.

Generative models have the quality to handle missing or irregular data, and also perform better than the discriminative models for small amounts of training data especially for static authentication. Discriminative models are fast in making predictions for new data, resulting in a faster classification of new data compared to the generative model. In addition, discriminative models have lower asymptotic error and higher performance than the generative model as the number of training samples become larger in continuous authentication. The experimental part of this work demonstrates that the partially observable hidden Markov model outperforms other anomaly detectors on short fixed-text keystroke benchmark datasets. The future work will be implementing a hybrid model using generative and discriminative model that will be suitable for both static and continuous authentication.

REFERENCES

[1] M. L. Ali, J. V. Monaco, C. C. Tappert, and M. Qiu. "Keystroke biometric systems for user authentication." Journal of Signal Processing Systems 86, no. 2-3 (2017): 175-190.

[2] W. Chen, and W. Chang. "Applying hidden Markov models to keystroke pattern analysis for password verification." In Information Reuse and Integration, 2004. IRI 2004. Proceedings of the 2004 IEEE International Conference on, pp. 467-474. IEEE, 2004.

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

[4] 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 ICB, pp. 640-646. 2006.

[5] S. K. Vuyyuru, V. V. Phoha, S. S. Joshi, S. Phoha, and A. Ray. "Computer user authentication using hidden markov model through keystroke dynamics." Manuscript submitted to ACM Transactions on Information and System Security (2006).

[6] 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, pp. 359-361. ACM, 2007.

[7] J. V. Monaco, and C. C. Tappert. "The Partially Observable Hidden Markov Model with Application to Keystroke Biometrics." arXiv preprint arXiv:1607.03854 (2016).

[8] Y. Zhang, G. Chang, L. Liu, and J. Jia. "Authenticating user's keystroke based on statistical models." In Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on, pp. 578-581. IEEE, 2010.

[9] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. "Planning and acting in partially observable stochastic domains." Artificial intelligence 101.1 (1998): 99-134.

[10] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77, no. 2 (1989): 257-286.

[11] T. Kobayashi, and S. Haruyama. "Partly-hidden Markov model and its application to gesture recognition." In Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, vol. 4, pp. 3081-3084. IEEE, 1997.

[12] Iobayashi, T., Junko Furuyama, and Ken Masumitsu. "Partly hidden Markov model and its application to speech recognition." In Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, vol. 1, pp. 121-124. IEEE, 1999.

[13] J. Ferguson, "Variable duration models for speech." In Proc. Symposium on the Application of Hidden Markov Models to Text and Speech, 1980, pp. 143-179. 1980.

[14] C. Mitchell, M. Harper, and L. Jamieson. "On the complexity of explicit duration HMM's." IEEE transactions on speech and audio processing 3, no. 3 (1995): 213-217.

[15] M. Russell, "A segmental HMM for speech pattern modelling." In Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on, vol. 2, pp. 499-502. IEEE, 1993.

[16] M. Russell, and R. Moore. "Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition." In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85., vol. 10, pp. 5-8. IEEE, 1985.

[17] D. K. D. Haussler, and M. G. R. F. H. Eeckman. "A generalized hidden Markov model for the recognition of human genes in DNA." In Proc. int. conf. on intelligent systems for molecular biology, st. louis, pp. 134-142. 1996.

[18] M. Ostendorf, V. V. Digalakis, and O. A. Kimball. "From HMM's to segment models: A unified view of stochastic modeling for speech recognition." IEEE Transactions on speech and audio processing 4, no. 5 (1996): 360-378.

[19] S-Z. Yu, "Hidden semi-Markov models." Artificial intelligence 174, no. 2 (2010): 215-243.

[20] H. Ozkan, A. Akman, and S. S. Kozat. "A novel and robust parameter training approach for hmms under noisy and partial access to states." Signal Processing 94 (2014): 490-497.

[21] S. Forchhammer, and J. Rissanen. "Partially hidden Markov models." IEEE Transactions on Information Theory 42, no. 4 (1996): 1253-1256.

[22] S. Forchhammer, and T. S. Rasmussen. "Adaptive partially hidden Markov models with application to bilevel image coding." IEEE transactions on image processing 8, no. 11 (1999): 1516-1526.

[23] J. V. Monaco, N. Bakelman, S-H Cha, and C. C. Tappert. "Recent advances in the development of a long-text-input keystroke biometric authentication system for arbitrary text input." In Intelligence and Security Informatics Conference (EISIC), 2013 European, pp. 60-66. IEEE, 2013.

[24] K. S. Killourhy, and R. A. Maxion. "Comparing anomaly-detection algorithms for keystroke dynamics." In Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on, pp. 125-134. IEEE, 2009.

[25] Y. Deng, and Y. Zhong. "Keystroke dynamics user authentication using advanced machine learning methods." Recent Advances in User Authentication Using Keystroke Dynamics Biometrics, GCSR 2 (2015): 23-40.