CHAPTER 2

Keystroke Dynamics User Authentication Using Advanced Machine Learning Methods

Yunbin Deng and Yu Zhong

User authentication based on typing patterns offers many advantages in the domain of cyber security, including data acquisition without extra hardware requirement, continuous monitoring as the keys are typed, and non-intrusive operation with no interruptions to a user's daily work. In this chapter, we adopt three popular voice biometrics algorithms to perform keystroke dynamics based user authentication, namely, 1) Gaussian Mixture Model with Universal Background Model (GMM-UBM), 2) identity vector (i-vector) approach to user modelling, and 3) deep machine learning approach. Unlike most existing keystroke biometrics approaches, which only use genuine users' data at training time, the proposed methods leverage data from a large pool of background users to enhance the model's discriminative capability. These algorithms make no assumption about the underlying probability distribution of the data and are amenable to real-time implementation. Although these techniques were originally developed for speech analysis, our experiments on the publicly available CMU keystroke dynamics dataset using these algorithms have shown significant reduction in the equal error

Yunbin Deng and Yu Zhong
BAE Systems
6 New England Executive Park, Burlington MA 01803, USA
e-mail: {Yunbin.deng, Yu.Zhong}@baesystems.com

Editors: Y. Zhong and Y. Deng, *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics*

DOI: 10.15579/gcsr.vol2.ch2, GCSR Vol. 2, pp. 23-40, 2015

rate over other published approaches. Finally, we discuss challenges and concerns for practical deployment of keystroke authentication technology.

2.1 Introduction

With the ever increasing demand for secure access control in many of today's security applications, traditional password methods fail to keep up with the challenges. As there are too many passwords to remember, users end up with simple passwords and/or shared passwords among many applications. To make things worse, studies have shown that even carefully crafted user names and passwords can be hacked easily. For example, during the Defcon 2010 context, hacker was able to hack 30,000 passwords out of 53,000 given passwords within 18 hours. Fortunately, biometrics [26, 27, 28, 41, 53] based on "who" a person is or "how" a person acts, as compared to what a person has (key) or knows (password) has made significant advancement to meet these new challenges.

Traditional biometrics, such as fingerprint [35], iris, face, and voice recognition, typically performs a one-time intrusive authentication with a user's cooperation. To achieve high security without any impact on a user's work efficiency, biometrics modalities capable of continuous authentication without a user's awareness are highly desirable. Among them, keystroke dynamics [37] provides a natural choice for continuous secure monitoring merely as a consequence of a user's hand interacting with the device. Keystroke dynamics refers to the habitual patterns or rhythms an individual exhibits while typing on a keyboard input device, including smartphones and tablets where a soft keyboard may be used. These rhythms and patterns of tapping are idiosyncratic, the same way as handwritings or signatures are, due to their similar governing neurophysiological mechanisms [12]. Back in the 19th century, telegraph operators could recognize each other based on one's specific tapping style [33]. Recently, it is shown that typing text can be deciphered simply based on the sound of key typing [60]. As such, it is believed that the keystroke dynamics contains enough information to ascertain a user at the keyboard.

Keystroke dynamics can be characterized using timing (such as the key down time for each key, latency between consecutive keys, and typing speed, etc.), finger pressure (the strength of typing, the sound it makes, etc.), touching style (touching size, resulting acceleration on the device, etc.), and typing habit (such as the use of special characters, typing errors and corrections, etc.). In addition, keystroke dynamics can be combined with traditional authorship biometrics to form a new "keyboard dynamic authorship" biometric that captures the neurophysiological process of both typing and writing. Unlike many other biometrics, the keystrokes information can be collected using software only without additional hardware. In summary, keystroke dynamics biometrics enables an emerging cost effective, user friendly, and continuous user authentication modality.

Although keystroke dynamics is governed by a person's neurophysiological pathway to be highly individualistic, it can also be influenced by his or her physical and psychological state. As a "behavioral" biometrics [55], keystroke dynamics exhibits instabilities due to transient factors such as fatigue, emotions, stress, drowsiness, etc.

[13]. It also depends on some external factors, such as the specific keyboard device used, possibly due to different layout of the keys. For example, the soft key layout on the smartphone is quite different form the PC keyboard layout. In addition, user typing patterns can be highly context dependent. Extreme cases can be programming v.s. online chatting. As such, keystroke data need to be collected at multiple sessions to model large variation and assess the robustness of various approaches.

Keystroke biometrics can use "static text", where a pre-defined text string, such as a password, is analyzed at a certain time, e.g., during the log on process. For more secure applications, "free text" with arbitrary input text and language should be used to continuously authenticate a user. Free text authentications are typically achieved by comparing some common word or sub-word strings that appear in both the enrollment and the verification phases. It has been shown that frequently used words tend to have better typing timing consistency [37]. To make timely decision for continuous keystroke authentication, the verification accuracy achievable using a single word is highly important and should be optimized first. Authentication accuracy can further be improved by observing more words.

This book chapter is focused on algorithmic work reducing error rate of single word "static text" task, but the discussed algorithms can be easily extended to the "free text" application domain. Experimental studies have shown that accurate keystroke biometrics can be achieved with a single word (with equal error rate $< 5\%$) and thus keystroke has the potential to be a highly accurate biometric modality with sufficient well-chosen testing data.

The rest of this chapter is organized as follows. Section 2.2 gives a brief overview of the current state of the art of keystroke biometric algorithms. Section 2.3 details our new approaches to the problem of accurate and discriminative keystroke dynamics model. Section 2.4 describes user authentication experiments and performance of the proposed algorithms on the CMU keystroke dynamics dataset. Section 2.5 discusses challenges for practical keystroke technology deployment and future research directions.

2.2 Keystroke Authentication Algorithm Literature Survey

The use of keystroke dynamics for verification and identification of a user has a long history and can be dated back to the 1970's [16, 46]. Since then, more than 200 papers, patents, and thesis have been published to tackle this problem [38, 40, 43, 47, 48, 49, 50], leveraging advances in signal processing, patter recognition, and machine learning. This area has been receiving growing research interest due to the increasing concerns of cyber security and access control.

However, most work used privately collected dataset to assess their system performance. This also makes the comparison of algorithm performance difficult, as each data set contains different number of subjects, varying number of data sessions, custom data collection protocols, among many other confounding factors. As such, some researchers have made their datasets available to the public, including the work by Allen, Bello, Giot, Jugurta and Maxion [2, 5, 19, 15, 31]. As data collection is a non-

trivial effort, these public datasets greatly reduce the barrier in this research field. Most of the existing datasets are based on "static text", with the exception of the BioChaves and the recent CMU dataset, which contains both static and free text [15, 32]. Compared to more mature biometrics such as fingerprints and iris, keystroke dynamics is still at its very early stage and various existing public keystroke datasets contain very limited number of subjects, ranging from over a dozen to over one hundred.

The most common keystroke dynamics features are based on the timing information of the key down/hold/up events, although some custom commercial keyboard can collect pressure information. This situation has changed dramatically with the popularity of the mobile smart devices, which are typically embedded with a rich suite of advanced sensors and will be detailed in the next section. The hold time or dwell time of individual keys, and the latency between two keys, i.e., the time interval between the release of a key and the pressing of the next key, also called flight time, are typically exploited. "Digraphs", which are the time latencies between two successive key down press, are commonly used. "Trigraphs", which are the time latencies between every three consecutive key down press, and similarly, n-graphs, have been investigated as well. In a keystroke study using free text, Sim and Janakiraman [45] investigated the effectiveness of digraphs and more generally n-graphs for free text keystroke biometrics, and concluded that n-graphs are discriminative only when they are word-specific.

In addition, the total duration of certain strings can be used as features. The relative order of duration times for different n-graphs was found to be more robust to the variations than absolute timing [6, 20]. The relative feature, when combined with absolute timing features, improved the authentication performance under free text scenario. Furthermore, derived features, including first and second order statistics and entropy of the basic lower level features, are investigated [36]. Other keystroke features studied including typing speed, percentage of special character, editing patterns, error rate, key-pair duration, n-graph duration [39, 52, 22].

From a pattern recognition and machine learning perspective, these approaches can be broadly classified into four categories: statistical methods based on distance metrics [51, 57], neural networks, statistical machine learning methods, and many other algorithms [4]. A brief review is given here and more recent advances are given in Section 2.4.

The first category uses the first and second order statistics of the basic features and applies various distance metrics and hypothesis testing. For example, Gaines et al. [17] performed a preliminary study on keystroke dynamics based authentication using T-test on digraph features. Monrose and Rubin [37] later extracted keystroke features using the mean and variances of digraphs and trigraphs. Using Euclidean distance metric and Bayesian-like classifiers, they achieved correct identification rate of 92% on their dataset. In addition to the Euclidean distance [7, 37], different distance metrics, such as the Mahalanobis distance [7, 9], and the Manhattan distance [3, 29], were also explored. Recently, we proposed a new distance metrics to combine the merits of both the Mahalanobis distance and the Mahattan distance metrics [59].

Various artificial neural networks (ANN) have been applied to the keystroke classification problem, including perceptrons, backpropagation neural networks, and Art 2 neural networks [21, 34, 4]. Neural networks are well known to be capable of learning

non-linear relationship among data dimensions. However, they often suffer from much slower training, manual selection of model architecture, tuning of many parameters with complex relationship, and poor generalization to new data.

Statistical machine learning algorithms, ranging from simple K-Nearest Neighbors (KNN) classifiers [9], to Bayesian classifiers [37], and support vector machines (SVMs) [56] have been applied to the keystroke classification problem. The SVMs have shown to work well for both identification and verification tasks. Compared with ANN approaches, SVMs have fewer parameters to tune and can be highly efficient in terms of both training and testing time.

The long history of keystroke research has also resulted in various other approaches, including K-means methods [30], Fuzzy logic [21], Fuzzy-ARTMAP, Histogram equalization of time intervals, Gaussian Mixture Model (GMM) [25], Hidden Markov Model (HMM), and genetic algorithms.

Various studies have reported a wide range of performance for similar algorithms, because most studies used their own dataset. To address this issue, Killourhy and Maxion collected and published a keystroke dynamics benchmark dataset [31]. Furthermore, they evaluated fourteen existing keystroke dynamics algorithms on this dataset, including Neural Networks [9], K-means [30], Fuzzy Logic [21], KNN, Outlier Elimination [21], SVMs [56], etc. Distance metrics including Euclidean distance [7], Mahattan distance [3, 29] and Mahalanobis distance [7] were also included. This keystroke dataset with the evaluation methodology and the performances of the state-of-the-art algorithms, provide a good benchmark to objectively assess progresses of new keystroke biometric algorithms. We report performance of recent advanced algorithms on this data set in Section 2.4.

2.3 Adopting Advanced Voice Biometrics Approaches to Keystroke Dynamics User Authentication

Most existing studies in keystroke authentication only use genuine users' data at training time to build a model for each genuine user and apply a user specific threshold at testing time for decision making on unforeseen test data. The key idea of recent advanced algorithms is to take advantage of the large amount of existing data from many subjects to build a background model of typical users, which enhances generalization and discriminative capability of the models. Note that the testing imposter subject does not need to be seen at the training time. We introduce three new algorithms into the keystroke authentication domain: Gaussian mixture model with universal background model (GMM-UBM), identity vector (i-vector), and deep neural network (DNN). A key common feature of these three algorithms is that an unsupervised training is conducted on a large pool of subjects at the first stage to allow the classifier take advantage of the overall data distribution in the feature space for improved performance. This enables a much more informed model than traditional methods which only use the data from the genuine data during the training and/or testing even though additional data are available to model the variations in the feature space.

For example, the traditional GMMs are trained on data collected from the genuine

user alone. Although the imposters' data are never seen, and not used in UBM training, the GMM-UBM approach has been a great success in state-of-the-art speaker verification system [42]. Here we apply this method to the domain of keystroke authentication.

Although many purely discriminative model approaches exist, such as ANNs and SVMs, models trained on a large amount of background users, without access to the real imposter's data at training time, do not guarantee good generalization performance to unforeseen imposters. Recently, DNN was proposed in the machine learning community as a generative-discriminative hybrid approach [24]. The unsupervised generative training step grants the model with good generalization capabilities to unforeseen test data, while the discriminate fine tune step endows the model with super classification accuracy. It has achieved better performance than ANNs and SVMs approaches in a variety of applications, including hand-writing digits recognition, speech and language modelling, face recognition, and object recognition. In this chapter we apply the DNN modeling approach to keystroke dynamics biometrics as well.

The following subsections detail the basic theory of these new approaches.

2.3.1 Gaussian Mixture Model with Universal Background Model (GMM-UBM)

2.3.1.1 Gaussian Mixture Model (GMM)

The Gaussian mixture model has been widely used in many statistical modeling tasks. It is a parametric model in the sense that it is parameterized by the mean vectors and covariance matrixes of data clusters with Gaussian distributions and the weights of all Gaussian components. It is a non-parametric model in the sense the real distribution of the data is unknown. A nice property of Gaussian mixture model is that with sufficient number of mixtures, the GMM can approximate an arbitrary probability distribution. However, a higher number of mixtures required more training data to achieve a well-trained model. In practice, the number of mixtures is determined by the amount of training data, the complexity of the real underlying data distribution, and the tradeoff between the accuracy and computational complexity.

A GMM is a weighted sum of M multivariate Gaussian functions [26]. The probability of a feature vector under the GMM is given by

$$p\left(x\,|\lambda\right) = \sum_{i=1}^{M} p_i b_i\left(x\right)$$

where x is a D-dimensional feature vector, $\lambda = \{p_i, \mu_i, \Sigma_i\}$is the model parameter, p_i is the mixture weights for the multi-variants Gaussian component densities $b_i\left(x\right)$, and μ_i, Σ_i are the mean vector and co-variance matrix for the multi-variant Gaussian distribution. The co-variance matrixes are sometimes assumed to be diagonal to dramatically reduce the number of parameter needed to be estimated, thus reduce the required amount of training data. These parameters are trained using a maximum-likelihood estimation principle, implemented using the Expectation Maximization (EM) algorithm. We perform an incremental GMM model training procedure, i.e., it starts

with a single mixture Gaussian model and estimates its parameter from the data. The single mixture Gaussian is then split into two mixtures of Gaussians and the parameters are re-estimated from the training data using the EM algorithm. This process repeats until the final desired number of mixtures is achieved. An important parameter in GMM is the variance floor. When a certain mixture of Gaussian has little sample in the training data, the estimated variance becomes very small to be a good estimate. In this case, we use a floor number to replace the estimated variance and this grants the model with better generalization capability.

2.3.1.2 GMM-UBM

Existing works in applying GMM to keystroke authentication is to train a GMM for each genuine user. At testing time, a keystroke feature is evaluated against the genuine user's GMM and a threshold is applied to the likelihood of the feature vector to make the decision [25].

The idea of GMM-UBM is to train another GMM from a large pool of so-called background subjects (except the genuine user and the actual testing subjects), in addition to the GMM for the genuine subject. When the pool of background subjects is large enough, the UBM will have a good chance to reasonably represent any imposter's data. Thus, the imposter can have a relatively high likelihood score under UBM, as compared to the genuine user's GMM. On the other hand, as the UBM is trained from a large pool of subjects, it is a relatively poor model for the genuine user, as compared to the genuine user's GMM, which is only trained from the genuine user. Thus, the genuine user's data has a better score on his/her own model as compared to the UBM. A likelihood ratio test can then be performed based on scores from these two models to make the authentication decision.

2.3.2 Identity Vector Approach

Identity Vector (i-vector) method was developed in the domain of speaker recognition research [10]. It can be considered as a way to learn a new compact low dimensional feature representation given an arbitrary sequence of feature vectors. This learning is typically conducted in an unsupervised fashion using data from a large pool of subjects. The learned new feature vector can then be either used to perform simple vector distance based similarity matching or as input to any further feature transform or machine learning modelling. This method has gained increasing popularity and became the state-of-the-art technique in the field of speaker recognition [8, 14, 18]. Recently, it was also reported to achieve the best mobile gait authentication accuracy on the largest mobile phone gait dataset [58].

The key advantages of i-vector method are:

1. It can be applied to any type of raw input feature.

2. It converts an input sequence of any length to a fixed low dimension feature vector, thus enables compact modelling and very fast matching.

3. Factor analysis is a build in step of i-vector training, which helps to remove many confounding factors in biometric analysis and extract a unique identity feature vector.

4. The i-vectors can be further processed with any existing discriminative feature transform and machine learning method.

However, this method has been mostly used in the speech community and less known to other fields of biometrics or object recognition. Effort to introduce it to the machine learning community has just started [1]. In this chapter, we adopt the i-vector model that is commonly used for speaker verification to keystroke biometrics. Despite their different application domains, voice biometrics and keystroke biometrics are similar in nature as both need to extract subject specific signatures from sensory data corrupted with variations from various irrelevant sources. The i-vector extraction method using total variability factor analysis provides an appealing solution to the keystroke identity extraction problem.

In the following we outline the i-vector extraction procedure. Interested readers should refer to [10] for more details. The i-vector modeling for user authentication consists of three major steps:

1. Build a universal background model (UBM) using a Gaussian mixture model (GMM) by pooling all or a subset of the feature vectors from the training data set.

2. Given the trained UBM (Ω), we compute a supervector for each enrollment or authentication keystroke feature vector of dimension F.

   a) the posterior probability (N_c) and Baum-Welch statistics ($\tilde{F}_c$) for each Gaussian component are computed as:

   $$N_c = \sum_{t=1}^{L} P\left(c \,|y_t, \Omega\right), \; and \, \tilde{F}_c = \sum_{t=1}^{L} P\left(c \,|y_t, \Omega\right)\left(y_t - m_c\right),$$

   where $m_c$is mean vector for Gaussian component c.

   b) The supervector M is obtained by concatenating ($\tilde{F}_c$) for all Gaussian components to form a vector of fixed dimension $C \times F$ for an input sequence of arbitrary length L.

3. Conduct factor analysis in the supervector space using a simplified linear model:

   $$M = m + Tw$$

   where m is a subject independent component, T is a low rank rectangular matrix, and w is the i-vector. The training process learns the total variability matrix T and a residue variability covariance matrix Σ. The i-vector is then computed as:

   $$w = \left(I + T^t \Sigma^{-1} N T\right)^{-1} T^t \Sigma^{-1} M,$$

   where N is a diagonal matrix consisting of diagonal blocks of $N_c I$.

Once an i-vector is extracted for each keystroke, the similarity between two keystroke is then computed as the cosine distance between their corresponding i-vectors:

$$d(w_1, w_2) = \frac{\langle w_1, w_2 \rangle}{\|w_1\| \|w_2\|}.$$

A simple threshold can be applied to the cosine distance score to make the final decision of accepting or rejecting an authentication attempt. When many enrollment sessions are available, multiply i-vectors can be computed for each subject, and more advanced algorithms can be applied to these i-vectors, such as linear discriminative analysis and SVM modeling for i-vectors. This study only uses simple cosine distance based matching on i-vectors to perform authentication.

2.3.3 Deep Neural Networks (DNN)

Deep neural networks are probabilistic generative models that are composed of multiple layers of hidden variables. The hidden variables typically have binary values and are called the feature detectors. These hidden layers can be trained one layer at a time, with the output of lower level layer serve as input to the higher level layer. The idea is to build a hierarchical generative model, so that each higher level layer captures more complex non-linear features in the data. These pre-trained two-layer generative models are then collapsed into a single multi-layer model and serve as an initialized ANN for further discriminative parameter fine tuning. The pre-training of a generative model is important to the generalization capability of the final model. It also facilitates the fine tuning of the DNN. It is well known that ANNs are sensitive to model parameter initialization and can be easily trapped in local optimal. The DNN pre-train not only avoids the random initialization of ANN parameters, but also helps to converge to global optimal, and significantly speeds up the ANN training process.

2.3.3.1 Pre-Train of DNN

Specifically, the first step of DNN training performs a layer-wise unsupervised training of restricted Boltzmann machines (RBMs). A RBM is one type of Markov random field that has two layers: the visible layer and hidden layer. The units in the visible layer ($\boldsymbol{v}$), are connected to all units in the hidden layer ($\boldsymbol{h}$) with associated weights $\boldsymbol{W}$. Note there is no connection within each layer. A simple RBM with four input units and three hidden units is shown in Fig.(2.1).

The units in the visual layer can be real value, integer, or binary, depending on the type of input data. The hidden units are typically binary stochastic variables, i.e, $h \in \{0, 1\}$. The Gaussian RBM is chosen for the first layer of RMB to model the real values of the keystroke features. The value of input and hidden variable defines the state of the machine and the energy of the state $\{v, h\}$ is defined as [44]

$$E(v, h; \theta) = \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i=1}^{D} \sum_{j=1}^{F} W_{ij} \frac{v_i h_j}{\sigma_i} - \sum_{j=1}^{F} a_j h_j$$

where $\theta = \{W, a, b, \sigma\}$ are parameters specifying the RBM. D is the number of input units, which is equal to the keystroke feature dimension. F is a user defined

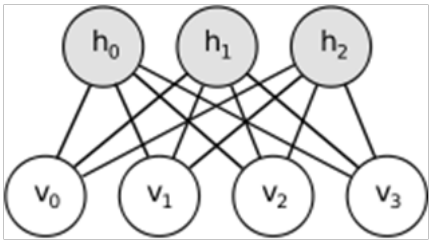

Figure 2.1: Restricted Boltzmann Machine.

parameter specifying the number of hidden units, a is a weight vector for the hidden units, while b and σ are bias and variance parameters for the input layer. The binary output of the first layer Gaussian RBM further serves as input for higher level RBMs to capture more complex non-linear structure embedded in the data. This process is also known as automatic feature engineering.

Higher level RBMs in the hierarchical generative are all defined as binary RBMs, i.e., both the visible and hidden layers contains only binary units. Their energy functions are defined as

$$\begin{aligned} E(v,h;\theta) &= -v^T W h - b^T v - a^T h \\ &= -\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} v_i h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j \end{aligned}$$

The RBMs are stochastic and the joint distribution of visible and hidden units is defined by

$$P(v,h;\theta) = \frac{exp(-E(v,h;\theta))}{z(\theta)},$$

where $z(\theta)$ is the normalization factor, known as partition function, and can be defined as

$$z(\theta) = \sum_{v}\sum_{h} exp(-E(v,h;\theta)).$$

The likelihood of the training data is then specified as

$$P(v;\theta) = \frac{\sum_h exp(-E(v,h;\theta))}{z(\theta)}.$$

However, the exact layer-wise maximum likelihood training of the RBM is intractable as the computation takes time that is exponential to the dimension of D and F. The approximate solution is provided by a technique known as "*Contrastive Divergence*" [23].

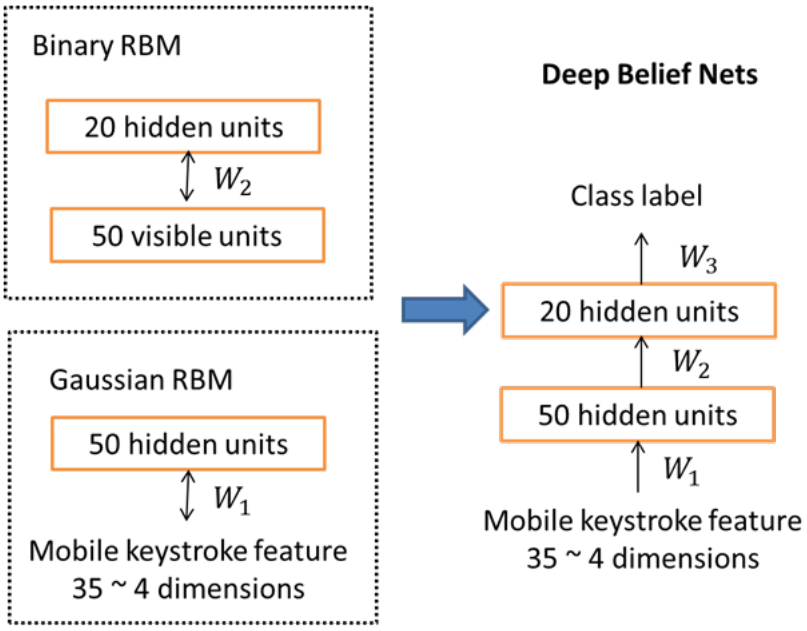

Figure 2.2: Train DNN for keystroke dynamics authentication in two steps. Left: unsupervised training of RBMs, Right: Convert RBMs into DNN.

2.3.3.2 Fine Tuning of DNN

The output of the unsupervised pre-train step is decks of RBMs, which can be stacked together and be added with a final classification layer to form an initialized ANN. This is conceptually illustrated in Fig.(2.2), where two RBMs are collapsed by sharing the middle units and a final layer (with weight W_3) is added to perform keystroke classification. The parameters of the final classification layer can be trained the same way as training a typical ANN with back-propagation.

2.4 Experiments

In this section, we evaluate the proposed keystroke biometric algorithms using the CMU keystroke dynamics benchmark dataset [31] because it came with the performance numbers of a range of existing keystroke dynamics algorithms for objective comparisons.

2.4.1 The CMU keystroke dynamics dataset

The CMU benchmark dataset contains keystroke dynamics consisting of the dwell time for each key and the latencies between two successive keys for static password string ".tie5Roanl". There were 51 subjects in the dataset. For each subject, there were eight

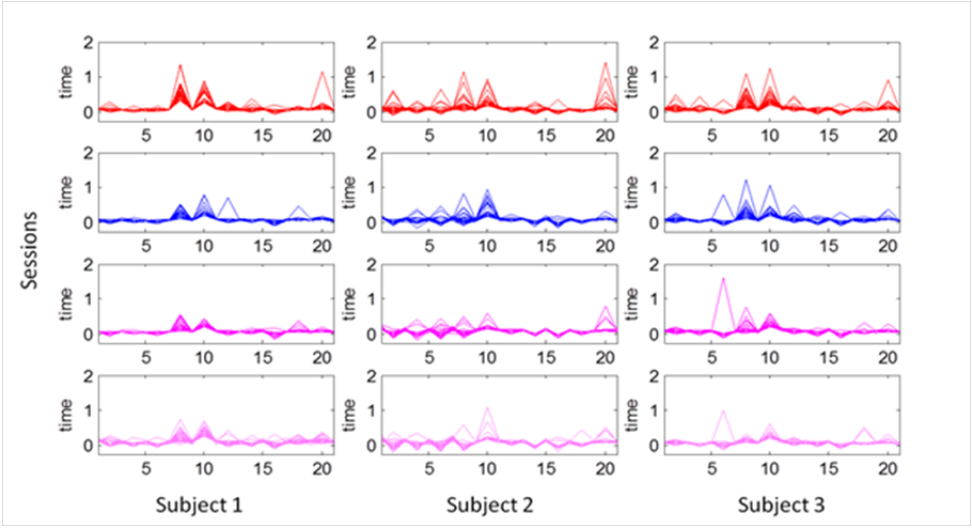

Figure 2.3: Keystroke dynamics features for static key string ".tie5Roanl" from the CMU keystroke dynamics benchmark dataset [31].

data collection sessions with at least one day apart between two sessions. 50 repeated keystroke strings were collected in each session, resulting in a total 400 sample for each subject.

For each typed 10-letter password and the final enter key, the dwell time and diagraphs give rise to a 21 dimensional feature vector. These feature vectors for three subjects from the first four data collection sessions are shown in Fig.(2.3). Although the keystroke features provide sufficiently distinguishing patterns for each subject, they are highly correlated, with large scale variations, and typical of noise and outliers. We have previously proposed a new distance metric to effectively handle these challenges that are intrinsic to keystroke dynamics data [59]. In this work, we show that GMM-UBM, i-vector and DNN based approaches perform even better to model large variations and correlations in the data.

We used the same evaluation methodology as in [31] to ensure objective performance comparisons. For each subject, we used the first 200 feature vectors as the training data. The remaining 200 feature vectors were used as positive test data and the first 5 samples from the remaining 50 subjects are used to form 250 negative feature vectors as imposters in the authentication phase for this user. To demonstrate the advantage of UBM, simple GMM (without UBM) experiment was also conducted for comparison. For the GMM-UBM and DNN experiments, the first four samples from background users were also included in the training set, resulting in additional 196 training samples from the negative class. Note that to test each subject as an imposter, all of his/her samples were excluded during the training time. It requires a total of 51*51 sets of experiments, for each genuine and imposter pair, such that each used different subjects' data for training and testing. For the simple GMM case (without UBM), only 51 sets of experiments are required.

The authentication accuracy is evaluated using the Equal Error Rate (EER) where the miss rate and false alarm rate are equal. The evaluations are performed for each subject. In the GMM-UBM and DNN experiments, for each genuine user, one single threshold is applied to all 51 tests. The mean and standard deviation of the equal error rates for the 51 subjects are reported.

2.4.2 The GMM, GMM-UBM, i-vector, and DNN Modeling Setup

For the GMM experiment, we build a Gaussian mixture model with 32 components for each genuine user. The variance floor for all feature dimensions and all Gaussian components are set to 0.01 to avoid poorly trained parameters. Each model for a genuine user applies its own threshold value to the likelihood scores of all test samples to compute the EER.

Under the GMM-UBM setting, for each genuine user, 51 sets of experiments were conducted, one for each test subject, to exclude the test subject's data from the UBM subjects set. The UBM is also modeled with 32 mixture of Gaussian. For each testing sample, the log likelihood ratio is computed for the genuine user model and the UBM model. A single likelihood ratio threshold is applied to the 51 sets of experiment to compute the EER.

We also performed 51 sets of experiments to evaluate the i-vector approach, one for each genuine subject. In each experiment, the first 200 tokens from the genuine subject and tokens 6 to 50 of other subjects are used to train the UBM and i-vector extractor matrix. The UBM contains 256 Gaussians and the i-vector dimension is set to 200.

To apply DNN to keystroke model, we first build a Gaussian RBM, with 31 visible units and 100 hidden units, and a Binary RBM, with 100 visible units and 100 hidden units. The ANN parameter fine tuning stops when the training error improvement is less than 1%.

2.4.3 Experimental Results

The performance, measured in mean and stand deviation of equal error rate (EER), of proposed GMM-UBM, i-vector and DNN approaches are listed in Table 2.1. For comparison, some of the best published results on the same dataset are also included in the table. The results have shown that the simple GMM based approach performs very close to our recently reported best results based on combined Mahalanobis and Mahattan distance, which outperform all 14 published algorithms on the same task [31].

When we include background users' data in the keystroke model, the GMM-UBM approach reduced the EER significantly compared to the simple GMM approach. The best performance is achieved using the DNN approach. Compared to the best reported EER of 8.4% on this dataset [10], the DNN approach reduces the EER to 3.5%, which is a 58% relative error rate reduction. This dramatic improvement is due to DNN's generative and discriminative modeling. The RBM generative modeling not only effectively captures the non-linear dynamics in keystrokes, but also ensures better

Table 2.1: Performance comparison of the proposed approach to the existing best reported algorithms on the same CMU dataset. Mean and (standard deviation) are shown for the equal error rate (EER). The proposed approaches significantly outperform the state of the art approaches.

Algorithm	**EER**
Neural Network (auto-assoc) [31]	0.161 (0.080)
SVM (one-class) [31]	0.102 (0.065)
Manhattan (scaled) [31]	0.096 (0.069)
Combined Mahalanobis and Mahattan distance [59]	0.084 (0.056)
GMM	0.087 (0.058)
I-Vector [10]	0.062 (0.053)
GMM-UBM	0.055 (0.052)
DNN [11]	0.035 (0.027)

generalization to samples from new users. The DNN discriminative parameter fine tuning step further boosts classification accuracy.

2.5 Discussion and Future Work

We have studied the characteristics of keystroke dynamics for traditional PC keyboards. We have introduced three popular machine learning approaches in the field of speaker verification to the domain of keystroke dynamics user authentication. These methods feature generative models trained using both positive samples from the genuine user and a large pool of background users, resulting in enhanced discriminative power. Our experimental studies on the CMU keystroke dynamics dataset have demonstrated the superior performances of the proposed GMM-UBM, i-vector and DNN approaches to a spectrum of top performing keystroke dynamics classifiers using traditional distance metrics statistics and advanced machine learning algorithms.

The achieved performance of at around 3% EER level is very encouraging, because this is achieved using only a single word. As a potential continuous user authentication modality, keystroke dynamics can conveniently acquire a large set of training and verification data from a specific user to achieve highly accurate keystroke authentication. Different from traditional one time authentication, continuous authentication could bring cyber technology security to a new level.

Although the proposed approaches are only evaluated on keystroke dynamics tasks using static text, they can be easily extended to free text use cases for continuous authentication. Given a large pool of free typing texts from a large set of subjects, the discriminative power each word and sub-word string can be discovered. Future research should work toward making large datasets of keystroke dynamics available to the research community, investigating the more challenging problem of keystroke biometrics using free text, developing richer key stroke features for mobile devices, studying context dependent sub-word and across word models, and seamlessly integrating language model score, i.e., the authorship, into the keystroke dynamic authentication system.

In addition, to deploy the technology to the market, the following issues need to be addressed: 1) Mitigate the effect of different hardware and network delay for remote access applications. 2) Enhance security without impact work efficiency by using subject-dependent threshold to balance two types of error. 3) Address the privacy issue because the continuous keyboard input can contain sensitive information, such as password and banking account, etc. 4) Like all existing biometics modalities, keystroke dynamics does not work very well for everyone [54]. Even with the most advanced algorithms, we noted poor performance for a few subjects. A multi-modal approach to non-intrusive continuous authentication is required for practical system depolyment.

References

[1] The 2013-2014 speaker recognition i-vector machine learning challenge. https://ivectorchallenge.nist.gov/.

[2] J.D. Allen. An analysis of pressure-based keystroke dynamics algorithms. Master's thesis, Southern Methodist University, Dallas, TX, U.S.A., 2010.

[3] L.C.F. Araujo, L.H.R. Sucupira Jr., M.G. Lizarraga, L.L. Ling, and J.B.T. Yabu-Uti. User authentication through typing biometrics features. In *1st International Conference on Biometric Authentication (ICBA)*, volume 3072 of *LNCS*, pages 694–700, 2004.

[4] S.P. Banerjee and D.L. Woodard. Biometric authentication and identification using keystroke dynamics: a survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.

[5] L. Bello, M. Bertacchini, C. Benitez, J. Pizzoni, and M. Cipriano. Collection and publication of a fixed text keystroke dynamics dataset. In *XVI Congreso Argentino de Ciencias de la Computación (CACIC)*, pages 822–831, 2010.

[6] F. Bergadano, D. Gunetti, and C. Picardi. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security*, 5(4):367–397, 2002.

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

[8] P.M. Bousquet, D. Matrouf, and J.F. Bonastre. Intersession compensation and scoring methods in the i-vectors space for speaker recognition. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 485–488, 2011.

[9] S. Cho, C. Han, D.H. Han, and H.I. Kim. Web-based keystroke dynamics identity verification using neural network. *Journal of Organizational Computing and Electronic Commerce*, 10(4):295–307, 2000.

[10] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[11] Y. Deng and Y. Zhong. Keystroke dynamics user authentication based on Gaussian mixture model and deep belief nets. *ISRN Signal Processing*, 2013, Article ID 565183, 7 pages, 2013.

[12] A. Dvorak, N. Merrick, W. Dealey, and G. Ford. *Typewriting Behavior*. American Book Company, New York, USA, 1936.

[13] C. Epp, M. Lippold, and R.L. Mandryk. Identifying emotional states using keystroke dynamics. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 715–724, 2011.

[14] X. Fang, N. Dehak, and J. Glass. Bayesian distance metric learning on i-vector for speaker verification. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2514–2518, 2013.

[15] J.R. Montalvão Filho and E.O. Freire. On the equalization of keystroke timing histograms. *Pattern Recognition Letters*, 27(13):1440–1446, 2006.

[16] G. Forsen, M. Nelson, and R. Staron Jr. Personal attributes authentication techniques. Technical Report RADC-TR-77-333, Rome Air Development Center, 1977.

[17] R. Gaines, W. Lisowski, S. Press, and N. Shapiro. Authentication by keystroke timing: some preliminary results. Technical Report Rand Rep. R-2560-NSF, RAND Corporation, 1980.

[18] D. Garcia-Romero and C.Y. Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 249–252, 2011.

[19] R. Giot, M. El-Abed, and C. Rosenberger. GREYC keystroke: a benchmark for keystroke dynamics biometric systems. In *IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, pages 1–6, 2009.

[20] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Transactions on Information and System Security*, 8(3):312–347, 2005.

[21] S. Haider, A. Abbas, and A.K. Zaidi. A multi-technique approach for user identification through keystroke dynamics. In *IEEE International Conference on Systems, Man, and Cybernetics (ICSMC)*, volume 2, pages 1336–1341, 2000.

[22] K. Hempstalk, E. Frank, and I.H. Witten. One-class classification by combining density and class probability estimation. In *European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD)*, volume 5211 of *LNCS*, pages 505–519, 2008.

[23] G.E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[24] G.E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep neural networks. *Neural Computation*, 18(7):1527–1554, 2006.

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

[26] A. K. Jain, R. Bolle, and S. Pankanti. *Biometrics: Personal Identification in Networked Society*. Kluwer Academic Publishers, 1999.

[27] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.

[28] A.K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross. Biometrics: a grand challenge. In *17th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 935–942, 2004.

[29] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2):168–176, 1990.

[30] P. Kang, S. Hwang, and S. Cho. Continual retraining of keystroke dynamics based authenticator. In *International Conference on Advances in Biometrics (ICB)*, volume 4642 of *LNCS*, pages 1203–1211, 2007.

[31] K.S. Killourhy and R.A. Maxion. Comparing anomaly detectors for keystroke dynamics. In *39th Annual International Conference on Dependable Systems and Networks*, 2009.

[32] K.S. Killourhy and R.A. Maxion. Free vs. transcribed text for keystroke-dynamics evaluations. In *Workshop on Learning from Authoritative Security Experiment Results (LASER)*, pages 1–8, 2012.

[33] J. Leggett and G. Williams. Verifying identity via keystroke characterstics. *International Journal of Man-Machine Studies*, 28(1):67–76, 1988.

[34] C.C. Loy, W.K. Lai, and C.P. Lim. Keystroke patterns classification using the ARTMAP-FD neural network. In *3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, volume 1, pages 61–64, 2007.

[35] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, NY, 2003.

[36] F. Monrose, M.K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

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

[38] B. Ngugi, B.K. Kahn, and M. Tremaine. Typing biometrics: impact of human learning on performance quality. *Journal of Data and Information Quality*, 2(2), 2011. Article No. 11.

[39] S. Park, J. Park, and S. Cho. User authentication based on keystroke analysis of long free texts with a reduced number of features. In *2nd International Conference on Communication Systems, Networks and Applications (ICCSNA)*, volume 1, pages 433–435, 2010.

[40] A. Peacock, X. Ke, and M. Wilkerson. Typing patterns: a key to user identification. *IEEE Security and Privacy*, 2(5):40–47, 2004.

[41] S. Prabhakar, S. Pankanti, and A.K. Jain. Biometric recognition: security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.

[42] D.A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *5th European Conference on Speech Communication and Technology (EUROSPEECH)*, volume 2, pages 963–966, 1997.

[43] J.A. Robinson, V.W. Liang, J.A.M. Chambers, and C.L. MacKenzie. Computer user verification using login string keystroke dynamics. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 28(2):236–241, 1998.

[44] R. Salakhutdinov. *Learning Deep Generative Models*. PhD thesis, University of Toronto, 2009.

[45] T. Sim and R. Janakiraman. Are digraphs good for free-text keystroke dynamics? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2007.

[46] E. Al Solami, C. Boyd, A. Clark, and A.K. Islam. Continuous biometric authentication: can it be more practical? In *12th IEEE International Conference on High Performance Computing and Communications (HPCC)*, pages 647–652, 2010.

[47] R. Spillane. Keyboard apparatus for personal identification. *IBM Technical Disclosure Bulletin*, 17(11), 1975.

[48] P.S. Teh, A.B.J. Teoh, and S. Yue. A survey of keystroke dynamics biometrics. *The Scientific World Journal*, Article ID 408280, 2013.

[49] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain. Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004.

[50] D. Umphress and G. Williams. Identity verification through keyboard characteristics. *International Journal of Man-Machine Studies*, 23(3):263–273, 1985.

[51] A. Vadivel, A.K. Majumdar, and S. Sural. Performance comparison of distance metrics in content-based image retrieval applications. In *International Conference on Information Technology*, pages 159–164, 2003.

[52] M. Villani, C. Tappert, G. Ngo, J. Simone, H.St Fort, and S.H. Cha. Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, page 39, 2006.

[53] J.D. Woodward, N.M. Orlans, and P.T. Higgins. *Biometrics: Identity Assurance in the Information Age*. McGraw-Hill, New York, USA, 2002.

[54] N. Yager and T. Dunstone. The biometric menagerie. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):220–230, 2010.

[55] R.V. Yampolskiy and V. Govindaraju. Behavioral biometrics: a survey and classification. *International Journal of Biometrics*, 1(1):81–113, 2008.

[56] E. Yu and S. Cho. GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification. In *International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 2253–2257, 2003.

[57] R.S. Zack, C.C. Tappert, and S.H. Cha. Performance of a long-text-input keystroke biometric authentication system using an improved k-nearest-neighbor classification method. In *4th IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS)*, pages 1–6, 2010.

[58] Y. Zhong and Y. Deng. Sensor orientation invariant mobile gait biometrics. In *International Joint Conference on Biometrics (IJCB)*, 2014.

[59] Y. Zhong, Y. Deng, and A.K. Jain. Keystroke dynamics for user authentication. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 117–123, 2012.

[60] L. Zhuang, F. Zhou, and J.D. Tygar. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security*, 13(1):Article No. 3, 2009.