Dynamic Signature Verification using Embedded Sensors

Abhijith Shastry, Ryan Burchfield, S. Venkatesan
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080, U.S.A
{abhijiths, ryanb, venky}@utdallas.edu

***Abstract*—This paper presents a new method for signature verification using a pen equipped with sensors. Traditional dynamic signature verification methods use digitizing tablets to record data. Here real time data is gathered using sensors embedded in the pen as the person signs. These sensors capture dynamic information of the signing process such as instantaneous acceleration, rotation, and other data. After processing raw data, classification is made using a combination of techniques such as dynamic time warping and hidden Markov models with Gaussian mixtures. Along with global feature comparison this method yields low false acceptance rate and false rejection rate. Details of a prototype system and performance on human subjects are also presented.**

Keywords-gaussian mixtures, hidden Markov model, sensors, signature verification

I. INTRODUCTION

Signatures are the primary mechanism for authentication and authorization in legal transactions [1]. Hence, the field of signature verification, which studies techniques to distinguish genuine signatures from forged ones, has been a major research area. However, signature verification has still remained a challenging problem because of large intra-class variations and, when considering forgeries, small inter-class variations [2].

The process of signature verification begins with a training phase in which multiple genuine samples of an individual are collected, personal features are extracted and stored in a knowledge base as shown in Fig. 1. Generally, automatic signature verification techniques involve three main phases: data acquisition and preprocessing, feature extraction, and classification as depicted in Fig. 2. Classification is made by comparing features from an input signature against the corresponding features of genuine signatures available in the knowledgebase.

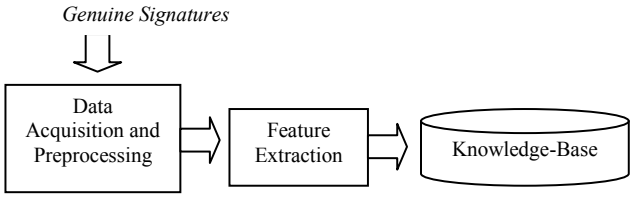

Figure 1. Training Phase of signature verification.

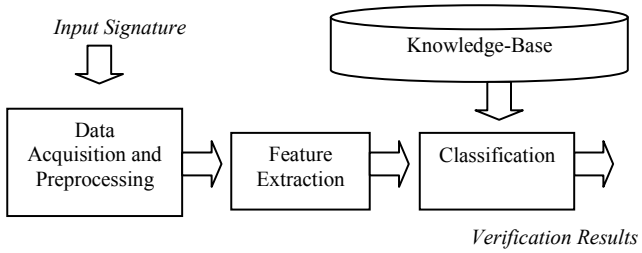

Figure 2. Process of signature verification.

On the basis of the data acquisition method, two categories of systems for handwritten signature verification can be identified: static (offline) systems and dynamic (online) systems [3]. Static methods analyze static visual record of the signature whereas dynamic methods classify the signature based on dynamic information of the pen trajectory (velocity, pressure, altitude, etc.). Significant research has been conducted in both static and dynamic signature verification techniques [3]. A variety of techniques have been proposed for signature verification which vary in the methods used in the three phases of signature verification. Techniques ranging from dynamic time warping (DTW) [4] to hidden Markov models (HMM) [2], [5] and neural networks [6], [7] have been used to aid signature verification. Different acquisition devices have also been used in order to capture data [3], [8]. Traditional dynamic signature verification techniques use devices such as touch screens or digitizing tablets [5], [9] as acquisition devices. Two error rates are commonly used to evaluate a signature verification mechanism: false acceptance rate (FAR) – fraction of forged signatures which are accepted as genuine and false rejection rate (FRR) – fraction of genuine signatures which are rejected as forgery attempts. The primary goal of any signature verification system is to obtain low FAR and FRR. Moreover, the automatic signature verification mechanism should be fast so that it can be used in real time transactions.

In this paper a signature verification technique is proposed which uses sensors embedded in the pen to acquire dynamic data. The technique has been tested on a custom fabricated prototype. The embedded sensors in the pen capture instantaneous acceleration and rotation over the duration of the signature. Thus, signatures are described by

978-0-7695-4431-1/11 $26.00 © 2011 IEEE
DOI 10.1109/BSN.2011.36

a set of six functions varying over time (acceleration along x, y and z axes and rotation along x, y and z axes). After processing the data and feature extraction, a combination of techniques based on DTW and HMM is used to classify the signature. DTW has been historically used as a measure of similarity between time series [3]. HMM has been extensively used in speech recognition [10] and in recent times has found application in the field of signature verification also [5]. There exist many customizations to the standard HMM which improve HMM as a modeling tool. Here, HMM with Gaussian Mixtures is used to model signatures. The advantage of modeling signatures with HMMs is that it is possible to accept variability in signing and, at the same time, capture the individual features of the signatures. Coupled with signature specific HMMs and thresholds we have designed a technique which has low error rates. It is possible that attempts may be made to crack this system by replay attacks. However such attempts may be thwarted by making small changes to the pens and requesting that subjects write different words instead of customary signing.

The data acquisition phase including hardware construction of the pen is explained in section II. Feature extraction is described in section III. A brief outline of DTW and HMM and the process of signature modeling and classification are presented in section IV. Section V describes experimental setup and results.

II. DATA ACQUISITION

A. Hardware Setup

Most of the existing work on signature verification uses commercially available electronic signature tablets to collect signatures for analysis. Unlike these previous works we investigate the use of accelerometers and gyroscopes to record the pen's movement during a signature. For this work, we have designed a hardware prototype. This custom designed inertial sensing unit is attached to an ordinary pen. The entire assembly is pictured in Fig. 3 and Fig. 4. The major components of the prototype are:

- Tri-axial accelerometer.
- Two gyroscopes.
- Microcontroller.
- Bluetooth module.
- USB interface.
- Power unit including battery charger.
- Temperature sensor.
- Magnetometer

Acceleration is measured by a STMicroelectronics LIS344ALH tri-axial accelerometer. The accelerometer is configured at the +/-2g sensitivity range as movements during a signature are not very extreme in nature. STMicroelectronics LPR510AL & LY510ALH gyroscopes are used to collect rotational information over all three motion axes. The gyroscopes collect data in the +/- 100 degree per second range. We use two gyroscopes to cover all three axes. The outputs of all sensors are filtered using single-pole RC filters.

The analog output of each sensor is digitized at a sample rate of 200Hz using built-in ADCs in the onboard Atmel Xmega192A3 microcontroller. Digitized samples are wirelessly transferred to a Bluetooth enabled device using the onboard STMicroelectronics SPBT2532C2 Bluetooth module. In addition to these currently used resources, the inertial sensing unit contains a temperature sensor, a USB interface, battery charger and a magnetometer. The temperature sensor is used to calibrate sensor readings in different thermal settings.

The prototype measures 1 inch by 3.5 inches and weighs 15.3 grams with battery. The pen used in the tests had a starting weight of 5.2 grams, for a total test weight of 20.5 grams. The sensing unit was placed 6.5 cms above the pen tip. A commonly available twist to extend tip pen weighs in at 20.3 grams so the additional weight of the inertial sensing unit is not expected to affect test subject's writing. A smaller prototype is possible and will be built subsequently.

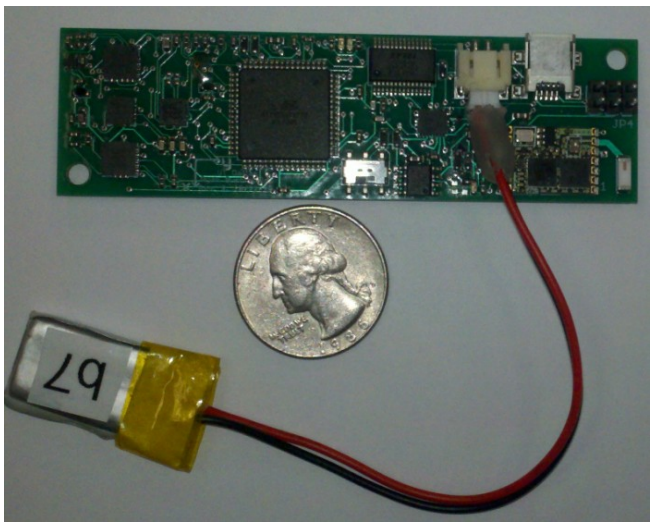

Figure 3. Custom designed hardware for data acquisition.

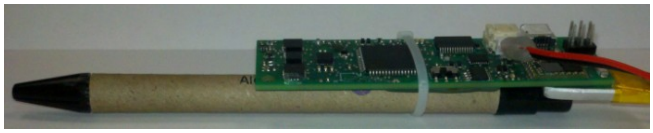

Figure 4. Data acquisition setup.

B. Pre Processing

Data obtained from the sensors is calibrated in order to compensate for hardware errors. After calibration, RMS (Root mean Square) of the accelerometer readings and RMS of the gyroscope readings are computed. This is done to normalize effects of different inclination during multiple uses by the same person. Thus the set of six time varying functions is reduced to two time varying functions. We plan

to include variation of pressure with time subsequently. This data is fed as input to the feature extraction phase.

III. FEATURE EXTRACTION

Two types of features can be used for signature verification: function or parameters [3]. When function features are used, the signature is usually characterized in terms of a time function whose values constitute the feature set. When parameter features are used, the signature is characterized as a vector of elements, each one representative of the value of a feature. We use a combination of function and parameter type features. The RMS values of gyroscope readings and accelerometer readings are used as the function features representing the signatures. Parameters are typically classified into two main categories: global and local. Global parameters deal with the whole signature; typical global features are total time duration of the signature, number of pen lifts, global orientation of the signature, coefficients obtained by mathematical transforms, etc. Local parameters are features extracted from specific parts of the signature [3]. In our work, some simple global parameters are derived from the input acceleration and rotation RMS values: number of maxima, number of minima, mean and standard deviation of the input acceleration and rotation time series.

IV. SIGNATURE MODELING AND CLASSIFICATION

A brief overview of DTW, HMM and Gaussian mixtures is presented after which their application in signature modeling and classification is discussed.

A. *Dynamic Time Warping*

DTW is popularly used as a time-series similarity measure which minimizes the effects of shifting and distortion in time by allowing elastic transformation of time series in order to detect similar shapes with different phases [11]. Given two time series X = (x1, x2, …xN) and Y = (y1, y2, …yM), DTW produces a real number d as output which represents the dissimilarity between the two sequences. Small values of d indicate that X are Y are similar. An example of how DTW aligns the signals in order to compute d is shown in Fig. 5.

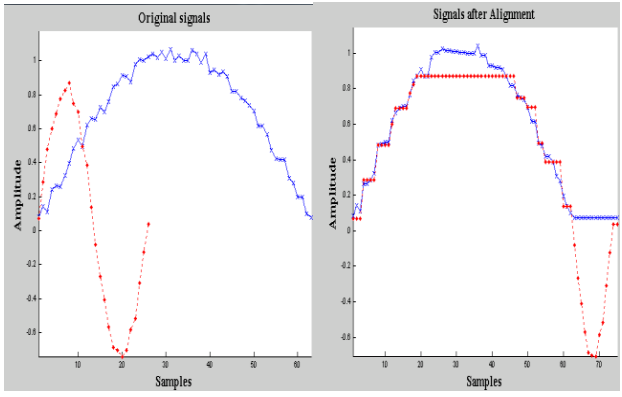

Figure. 5. Alignment of series as done by DTW.

B. *Hidden Markov Model*

HMM models a doubly stochastic process governed by an underlying Markov chain with a finite number of states and a set of random functions each of which is associated with one state [2]. The model is hidden in the sense that all that can be seen is a sequence of observations. The underlying state which generated each symbol is hidden. The observations may be discrete or continuous in nature. In the former case a discrete HMM is used, and in the latter case a continuous HMM is used.

The following items comprise a discrete HMM:

- A set of N states, $\{1, \ldots, N\}$
- An alphabet of M output symbols, $\Sigma = \{\sigma_1, \ldots, \sigma_M\}$
- A set of output probabilities, $B = \{b_{i,j} | 1 \le i \le N, 1 \le j \le M\}$
- A set of state transition probabilities, $A\{a_{i,j} | 1 \le i \le N, 1 \le j \le N\}$
- An initial state probability distribution, $\pi = \{p_1, \ldots, p_N\}$

A time series of values is generated by a discrete HMM as follows. At every time step the HMM is in one of its N states. The state in which the HMM begins at the first time step is chosen according to the initial state distribution π. An element of Σ is then selected according to the output distribution of the current state and that symbol is emitted. Next, the HMM transitions to a new state based on the transition distribution of the current state. Finally, this process of selecting an output symbol and transitioning to a new state repeats [11]. The description of a continuous HMM is similar to that of the discrete HMM except for the presence of output probability distribution function instead of discrete output probabilities.

It is possible to constrain a HMM such that only certain desired state transitions are allowed. A popular constrained HMM which is used in speech recognition [10] and signature verification [2] is left-to-right HMM. In this model, states are not permitted to transition to previous states. Fig. 6 illustrates a left-to-right HMM with four states.

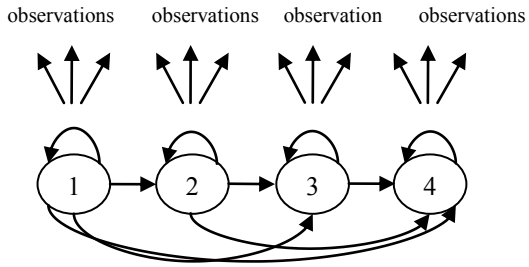

Figure. 6. Left-to-right HMM with four states.

C. *Gaussian Mixtures*

Gaussian Mixtures are combinations of Gaussian, or

'normal' distributions. A mixture of Gaussians can be written as a weighted sum of Gaussian densities. The d-dimensional Gaussian probability density function is generated using the parameters:

- Mean vector, μ
- Covariance matrix Σ

A weighted mixture of K Gaussians can be written as:

$$gm(x) = \sum_{k=1}^{K} w_k \cdot g_{(\mu k, \Sigma k)}(x) \qquad (1)$$

where the weights are all positive and sum to one.

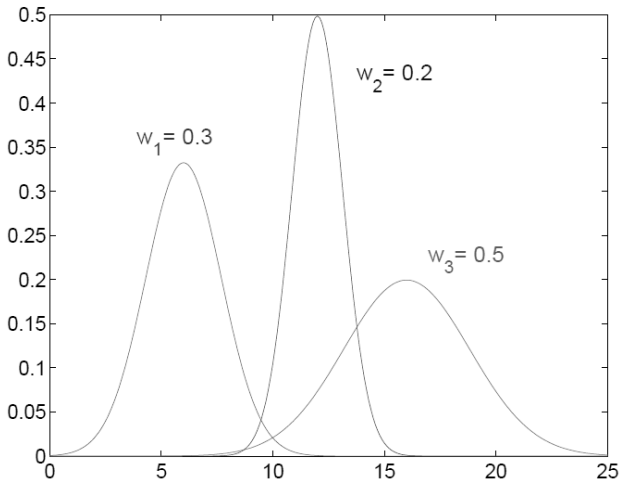

Figure 7. One dimensional Gaussian mixture pdf, showing the three components which make up the mixture.

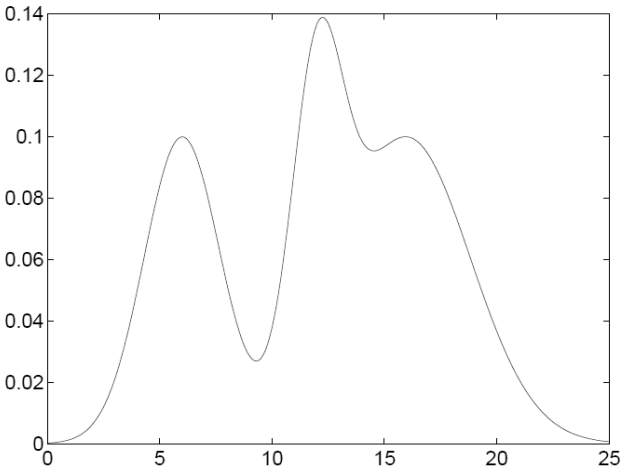

Figure 8. One dimensional Gaussian mixture pdf consisting of three single dimensional Gaussians.

The probability density function generated using a mixture of three one dimensional Gaussians is illustrated in Fig. 7 and Fig. 8. Any complex probability distribution function can be modeled using mixture of Gaussians with proper choice of parameters.

D. Signature Modeling

Signatures are modeled using continuous left-to-right HMMs with two dimensional Gaussian mixtures to model observation probability densities. The two dimensions are the RMS acceleration values and RMS rotation values. During the training phase a HMM is constructed using the genuine samples provided for training. The knowledge base stores this person-specific HMM which is used in the classification phase. Along with the HMM, all the training samples are stored in the knowledge base.

Although signatures can be modeled using discrete HMMs after quantizing the continuous input, a continuous HMM is preferred because errors can be induced due to the quantization process. For instance in a particular HMM state, the discrete probability of generating neighboring symbols of a particular observation symbol may vary by a large amount. However, in a continuous output space, the probability of neighboring points do not vary by large amounts. Thus, the observation probability distribution of each HMM state is estimated using a mixture of two dimensional Gaussians. To construct the HMM many parameters need to be computed. Some parameters are predefined for all models using cross validation:

- The number of states of the HMM, N.
- The number of Gaussian mixtures in each state, M.

Other parameters are computed specifically for the corresponding model during the training phase. These include global HMM parameters:

- Initial HMM state probability distribution, $\pi = \{p_1, \ldots, p_N\}$
- A set of state transition probabilities, $A\{a_{i,j} | 1 \le i \le N, 1 \le j \le N\}$

Gaussian mixture parameters need to be computed for each state and dimension. Thus 2N sets of the following parameters need to be computed:

- A set of weights of the Gaussians used in the mixture $W = \{w_1, w_2, \ldots, w_M\}$.
- A set of mean vectors $\mu = \{\mu_1, \mu_2, \ldots, \mu_M\}$.
- Set of covariance matrices $\Sigma = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_M\}$.

All the parameters are computed using EM (Expectation-Maximization) algorithm in order to maximize the likelihood of training data generation using the model as described in [13]. EM algorithm produces a set of parameters which using which a maxima is achieved. However, it does not guarantee a global maximum. EM algorithm starts with initial assignment of parameter values and improves the parameters iteratively. This initialization is very critical in deciding the maxima achieved. Since we use a constrained version of the HMM (left-to-right), initialization becomes easy using these constraints. Three other parameters are computed and stored in the knowledgebase to aid classification:

- The average DTW distance $d_{TRAINACCEL}(i)$ between

the RMS acceleration of i[th] training instance and the remaining training instances is computed. The maximum of these $d_{TRAINACCELMAX}$ is computed and stored.

- The average DTW distance $d_{TRAINGYRO(i)}$ between the RMS rotation of i[th] training instance and the remaining training instances is computed. The maximum of these $d_{TRAINGYROMAX}$ is computed and stored.
- The probability of generating the i[th] training instance by the HMM $p_{TRAIN(i)}$ is computed. This can be easily done using the Forward-Backward procedure as described in [5]. The maximum probability among these $p_{TRAINMAX}$ is computed and stored.

Thus, at the end of the training phase, the following are available in the knowledge base for each person:

- The processed training data. (RMS accelerations and gyroscope readings.)
- Global parameters computed in the feature extraction phase for all training instances.
- The parameter $d_{TRAINACCELMAX}$.
- The parameter $d_{TRAINGYROMAX}$.
- The parameter $p_{TRAINMAX}$.

E. Signature Classification

When a test signature is presented, it is classified as a genuine signature or a forgery attempt after the feature extraction phase. The following are used for classification:

- Global features which are extracted during feature extraction phase. These global features can be used to eliminate samples which differ a lot from those in the training set by comparing with those in the knowledgebase.
- The average DTW distance $d_{DTWACCEL}$ between the RMS acceleration of the given test sample and all the training instances is computed.
- The average DTW distance $d_{DTWGYRO}$ between the RMS rotation of the given test sample and all the training instances is computed.
- The probability of generating the test sample by the HMM for this particular signature p_{HMM} is computed.

Next, $d_{DTWACCEL}$, $d_{DTWGYRO}$, and p_{HMM} are compared with $d_{TRAINACCELMAX}$, $d_{TRAINGYROMAX}$,, and $p_{TRAINMAX}$ respectively to classify the signature. To classify the test sample as genuine the following should hold:

$$d_{DTWACCEL} \le (1+\varepsilon_1) \cdot d_{TRAINMAXACCEL} \quad (2)$$

$$d_{DTWGYRO} \le (1+\varepsilon_2) \cdot d_{TRAINMAXGYRO} \quad (3)$$

$$p_{HMM} \ge (1-\varepsilon_3) \cdot p_{TRAINMAX} \quad (4)$$

where ε_1, ε_2, and ε_3 represent the constant thresholds which should be close to zero. The threshold values are derived from experimental results.

V. Experiments

A. Experimental Setup

Experiments were carried out on four test subjects with a total of 60 genuine and 180 forgery signatures in two languages. Test subjects from different cultural and linguistic background were chosen in order to try out the proposed method on signatures resulting from different cultures. Signatures from languages such as Persian which are written from right-to-left were also collected and analyzed.

B. Experimental Results using HMM in isolation

This section describes experimental results obtained when error rates are measured using just the HMM for classifying the signature. This is done to evaluate the modeling capability of the HMM. Fig. 9 illustrates the variation of FRR and FAR with the number of states in the HMM used to model the signature. The number of Gaussian mixture components used in each state was one. The number of training instances used was five. As depicted by the figure, even with one component in the Gaussian mixture per state, low FRR and FAR values of 1.72% and 12.22% are obtained with 30 HMM states. The performance increases with the number of HMM states till 35 after which saturation is reached.

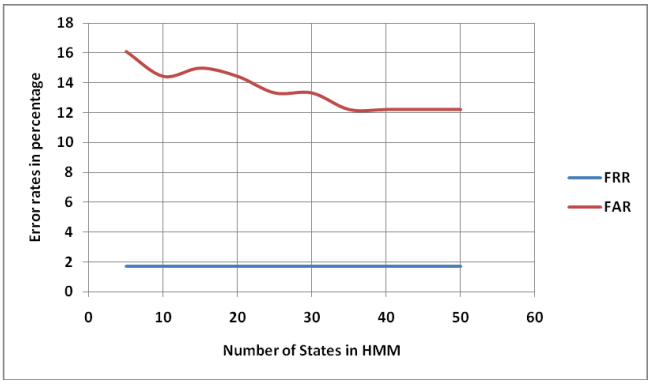

Figure 9. Variation of FAR and FRR expressed as percentage with the number of states in HMM.

Fig. 10 illustrates the variation of FRR and FAR with the number of Gaussian mixture components with 30 HMM states. The number of training instances used was five. Performance deteriorates with the increase of Gaussian mixture components. Thus, a HMM with just one Gaussian component used to model observation probability gives superior performance. Moreover, this reduces the model complexity and thus speeds up the process of signature verification.

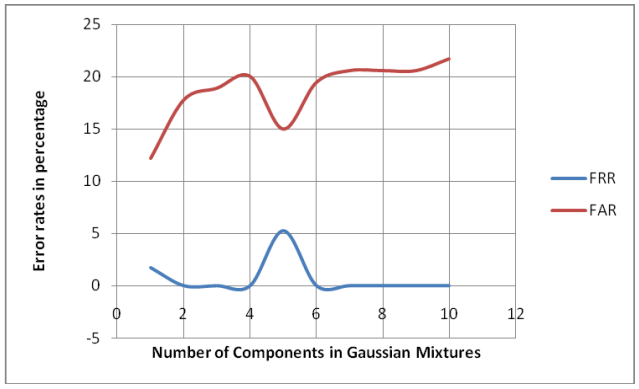

Figure 10. Variation of FAR and FRR expressed as percentage with the number of components constituting the Gaussian mixtures.

The variation of FRR and FAR with the number of training instances is illustrated in Fig. 11. This experiment was carried out with 30 HMM states and one Gaussian mixture component. As expected, performance increases with the number of training instances initially. However, threshold computation is affected when the number of training instances becomes very large due to which error rates increase. The error rates are low enough that this technique can be used in practical situations where the number of training instances is limited to three or five. With five training instances, we obtain FRR = 1.72% and FAR = 12.22% using just the HMM to perform classification.

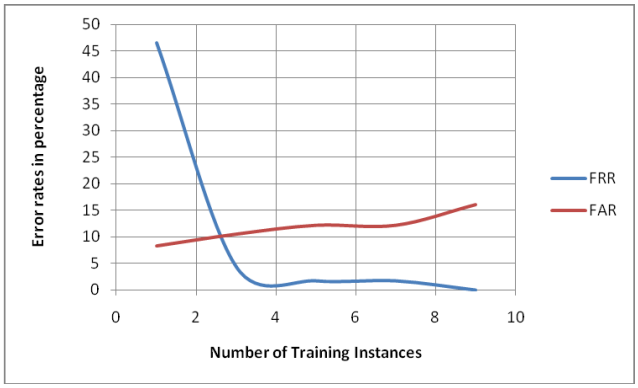

Figure 11. Variation of FAR and FRR expressed as percentage with the number of training instances.

C. *Experimental Results using HMM and other features*

When the proposed technique is used end to end, including elimination using global features and comparison using DTW, we obtain FRR = 3.45% and FAR = 0%. This experiment was carried out with 30 HMM states, 1 Gaussian mixture component, and five training instances. Thus, this system is resilient to forgery attempts and has an acceptable FRR.

VI. CONCLUSION

We have presented a new technique for dynamic signature verification using HMM with Gaussian mixtures to represent output probabilities. The signature features are denoted by the RMS values of acceleration and rotation. Experimental results show good signature verification performance. This makes it possible to use this technique as a means of authentication in business transactions in real time. We are currently investigating the performance with pressure readings as another dimension in Gaussian mixtures. We are also working on reducing the form factor of the hardware unit significantly.

ACKNOWLEDGMENT

The authors are thankful to Research in Motion for supporting this research in the form of a gift to The University of Texas at Dallas.

REFERENCES

[1] S. Nanavati, M. Thieme, and R. Nanavati, *Biometrics: Identity Verification in a Networked World.* New York: Wiley, 2002, pp. 123–131.

[2] Julian Fierrez, Javier Ortega-Garcia, Daniel Ramos, Joaquin Gonzalez-Rodriguez, "HMM-based on-line signature verification: Feature extraction and signature modeling," *Pattern Recognition Letters*, vol. 28, pp. 2325-2334, Dec 2007.

[3] Impedovo, D.; Pirlo, G.; , "Automatic Signature Verification: The State of the Art," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* , vol.38, no.5, pp.609-635, Sept. 2008.

[4] Jayadevan R, Satish R Kolhe, Pradeep M Patil, "Dynamic Time Warping Based Static Hand Printed Signature Verification," *Journal of Pattern Recognition Research*, vol 4, pp. 52-65, 2009.

[5] L. Yang, B. K. Widjaja, and R. Prasad, "Application of hidden Markov models for signature verification," *Pattern Recognit.*, vol. 28, no. 2, pp. 161–170, Feb. 1995.

[6] H. Cardot,M. Revenu, B. Victorri, and M.-J. Revillet, "A static signature verification system based on a cooperating neural networks architecture," *Int. J. Pattern Recognit. Artif. Intell. (IJPRAI),* vol. 8, no. 3, pp. 679–692, 1994.

[7] R. Bajaj and S. Chaudhury, "Signature verification using multiple neural classifiers," *Pattern Recognit*., vol. 30, no. 1, 1997, pp. 1–7.

[8] Baron, R.; Plamondon, R.; , "Acceleration measurement with an instrumented pen for signature verification and handwriting analysis," *Instrumentation and Measurement, IEEE Transactions on* , vol.38, no.6, pp.1132-1138, Dec 1989.

[9] D. J. Hamilton, J. Whelan, A. McLaren, I. Macintyre, and A. Tizzard, "Low cost dynamic signature verification system," in *Proc. Eur. Convention Secur. Detection*, 1995, no. 408, pp. 202–206.

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

[11] Nikos Mamoulis , David W. Cheung , Eamonn Keogh, "Fast and Exact Warping of Time Series Using Adaptive Segmental Approximations," *Machine Learning*, vol. 58, pp. 231-267, Feb 2005.

[12] Tim Oates, Laura Firoiu, and Paul R. Cohen, "Using Dynamic Time Warping to Bootstrap HMM-Based Clustering of Time Series," *Sequence Learning, Lecture Notes in Computer Science*, vol. 1828, pp. 35-52, 2001.

[13] Guorong Xuan; Wei Zhang; Peiqi Chai; , "EM algorithms of Gaussian mixture model and hidden Markov model," Image Processing, 2001. Proceedings. 2001 International Conference on , vol.1, no., pp.145-148 vol.1, 20.