IET Biometrics

Research Article

Online signature verification using i-vector representation

ISSN 2047-4938
Received on 9th April 2017
Revised 21st November 2017
Accepted on 28th November 2017
E-First on 12th February 2018
doi: 10.1049/iet-bmt.2017.0059
www.ietdl.org

Hossein Zeinali[1], Bagher BabaAli[2] ✉, Hossein Hadian[1]
[1]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[2]School of Mathematics, Statistics and Computer Science, University of Tehran, Tehran, Iran
✉ *E-mail: babaali@ut.ac.ir*

Abstract: Signature verification (SV) is one of the common methods for identity verification in banking, where for security reasons, it is very important to have an accurate method for automatic SV (ASV). ASV is usually addressed by comparing the test signature with the enrolment signature(s) signed by the individual whose identity is claimed in two manners: online and offline. In this study, a new method based on the i-vector is proposed for online SV. In the proposed method, a fixed-length vector, called i-vector, is extracted from each signature and then this vector is used for template making. Several techniques such as nuisance attribute projection (NAP) and within-class covariance normalisation (WCCN) are also investigated in order to reduce the intra-class variation in the i-vector space. In the scoring and decision making stage, they also propose to apply a 2-class support vector machine method. Experimental results show the proposed method could achieve 8.75% equal error rate (EER) on SigWiComp2013 database in the best case. On SVC2004 database, it also achieved 5% EER that means 11% relative improvement compared with the best reported result. In addition to its considerable accuracy gain, it has shown significant improvement in the computational cost over conventional dynamic time warping method.

1 Introduction

Identity verification is defined as determining a person's identity based on their physical or behavioural characteristics. There are different such characteristics in each category (i.e. physical/behavioural). The first category is, in fact, the set of biometric characteristics such as fingerprint, eye — especially the iris and cornea — and speech [1]. In the second category, which is based on behavioural characteristics, features such as handwriting and signature are used to identify a person [1]. From the perspective of how a person utters words and his habitual phrases, speech can be placed in the second category too. The methods in the first category are superior in terms of accuracy, and among them fingerprints are more widespread and are very commonly employed in time and attendance systems. However, since it is much more convenient to use signatures, methods based on such characteristics are more widespread in banking, even though they are less accurate than the methods in the first category. As a behavioural biometric, handwritten signature can be influenced by a person's mental and physical conditions, hence it exhibits inconstancy due to stress, emotions, sleepiness, fatigue etc.

The use of handwritten signature as an authentication modality has a long history. In the conventional method of signature verification (SV) — which is common in banks — an operator performs the task of verifying or rejecting a signature. Since the 1970s, due to the increasing concerns of access control, automatic SV (ASV) has been receiving growing research interest to perform this task automatically using computers [2, 3]. By leveraging advances in signal processing and machine learning, ASV is done in two manners: online and offline. In the offline verification, also called static verification, we only have access to the image of the signature [4–6]. In these kinds of methods, we usually normalise the size of the image after some preprocessing and then extract features from the image using a sliding window. The features are then used to compare two signatures. On the other side are the online methods, also called dynamic methods, where informations related to dynamics of the signature are provided as well as the image of the signature [7–9]. Dynamic information includes pressure, velocity, azimuth etc. In these methods, the changes in the vertical and horizontal directions are usually used as shape-related features. These methods have a better performance compared with the offline methods and are more reliable since they use more information extracted from the signature. Apart from these advantages, signature forgery is more difficult in these methods because they use dynamic features such as velocity and azimuth that are very difficult to simulate. Our focus in this research is on the online methods.

There have been many studies on the online SV, which can be grouped into two main categories:

- *Methods based on global features of signature:* These methods try to extract a fixed-length vector from the whole signature, so that signatures can be easily comparable in the vector form. These methods can be further categorised into two subcategories: in the first one, we try to extract the global features from the entirety of signatures. For example, in [10] *Jain et al.* use the number of strokes as a global feature. Authors use other features such as average velocity, average pressure, and the number of times the pen is lifted during the signature in [11]. As a good example, Fierrez-Aguilar *et al.* in [7] introduce 100 global features sorted by their individual discriminative power. A subset of these features is employed in other studies too [8, 12–15]. In the second subcategory, a transformation is applied to the signature to give a fixed-length vector. For instance, a wavelet transform is used in [16] to extract a feature vector from the whole signature. In another study, discrete cosine transform (DCT) is used to obtain the fixed-length feature vector [9]. The proposed method in this paper is categorised in this group.
- *Functional methods:* The methods in this category focus more on comparing signatures and calculate the distance between two signatures. In these methods, each signature is represented using a sequence of local features extracted from it. This category can be further divided into two subcategories as well: the methods in the first one do not perform any kind of modelling. In fact, in these methods, a reference set is kept for each individual, and in the test time the input signature is compared with the reference set in order for decision making. The most common method in this subcategory is the dynamic time warping (DTW) method, which is used in many studies [17–20]. The second category

includes methods which train a probabilistic model for each individual using the signatures in her/his reference set. These methods usually use likelihoods for scoring and decision making. The most common methods in this subcategory are hidden Markov model (HMM) [21–25] and Gaussian mixture model (GMM) [26–28].

The aim of this paper is to propose a method for the online SV based on i-vector. i-Vector was first proposed for speaker recognition application [29] and later was adopted in other applications such as language identification [30, 31], accent identification [32], gender recognition, age estimation, emotion recognition [33, 34], audio scene classification [35] etc. In the main application of this method (i.e. speaker recognition), a fixed-length vector called *i-vector* is extracted from a speech signal of arbitrary duration. In next steps, this vector is used for computing scores and recognition. While i-vector is used mostly in many speech-related applications, it is less known to other fields. In this paper, we adopt the i-vector that is commonly used for speaker recognition to SV. Despite their different domains, voice biometrics and signature biometrics are similar in nature as both need to extract subject specific patterns from captured signal contaminated with variations from various irrelevant sources. As total variability factor analysis is a built-in step of i-vector training, which helps to remove the distracting factors in biometric analysis and extracts a unique identity representation vector, we expect that i-vector should be able to provide a promising solution to the signature identity extraction problem.

We have two motivations for using this method for SV. First, online signatures have variable lengths similar to the speech signals. By using this method, we can have a fixed-length vector which facilitates the next steps in the decision making. Therefore, after extracting temporal features from each signature, we extract an i-vector. Since we acquire a fixed-length vector for each signature, we can place this method in the first category above. The second motivation is that usually signatures of a person are slightly different each time. These differences lead to intra-class variations which in turn increase the false rejection rate (FRR). In different applications of i-vector in speech processing, various methods for reducing intra-class variations have been proposed which can be adopted in this application as well. Similar to the speaker verification case, we also propose to use two different techniques to reduce the intra-class undesirable variations effects. Since for each individual there are several signature samples as the reference set in enrolment phase, we proposed to add them up to the data that is used to train the within-class variation compensation methods. In addition, we proposed to apply a 2-class support vector machine (SVM) to discriminate between i-vectors extracted from genuine and forged signatures. Experimental results revealed the effectiveness of the mentioned ideas on two different databases. Fig. 1 depicts the block diagram of the proposed i-vector-based online SV system.

The rest of this paper is organised as follows: in Section 2, theories related to i-vector and methods of reducing intra-class variations are described. The proposed method is then explained in Section 3. In Section 4, we explain the procedures for extracting features and afterwards in Section 5: first, the experimental setup is described and then in Section 6 the experimental results are presented. Finally, the conclusions of this paper are derived in Section 7.

2 i-Vector-based systems

As explained earlier, currently i-vector in total variability space has become the state-of-the-art approach for speaker recognition [29]. This method that was introduced after its predecessor method, joint factor analysis [36, 37], can be considered as a technique to extract a compact fixed-length representation given a signal with arbitrary length. Then, the extracted compact feature vector can be either used for vector distance-based similarity measuring or as input to any further feature transform or modelling. There are certain steps to extract i-vector from a signal. First, features should be extracted from the input signal and then the Baum–Welch statistics should be extracted from the features [38], and finally i-vector is computed using these statistics. In the following, we explain these steps in details.

2.1 Universal background model (UBM) training

The first step in i-vector extraction pipeline is to create a global model which is called an UBM [39]. For UBM, various models are used based on the application. Usually, GMM is used for this purpose in text-independent speaker verification [29, 40] and HMM is used in text-dependent applications [41–43]. In SV tasks, since the signatures are different for each individual, it is not possible to train a universal HMM. Therefore, we train a GMM from all the extracted features of all individuals in the development set. There should be sufficient training data in the development set for this model to properly cover the feature space.

A GMM is a weighted set of C multivariate Gaussian distributions and formulated as

$$\Pr(\boldsymbol{x}\,|\,\lambda) = \sum_{c=1}^{C} w_c \mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{m}_c, \boldsymbol{\Sigma}_c), \tag{1}$$

where $\boldsymbol{x}$ is a D-dimensional vector with continuous values, w shows the weight for each component of the mixture, and $\mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{m}_c, \boldsymbol{\Sigma}_c)$ shows the Gaussian distribution with mean $\boldsymbol{m}_c$ and covariance matrix $\boldsymbol{\Sigma}_c$. The sum of all weights should be equal to one. Usually, GMM is used with a diagonal covariance matrix in practise and we use a diagonal matrix in this study too [39].

2.2 Extraction of Baum–Welch statistics

In this step, for each feature sequence, the zero and first-order Baum–Welch statistics are computed using the UBM [37, 38]. Given $\boldsymbol{X}_i$ as the entire collection of feature vectors for training signature ith, the zero, and first-order statistics (i.e. N_c and $\boldsymbol{F}_c$) for the cth component of the UBM are computed as follows:

$$N_c(\boldsymbol{X}_i) = \sum_t \gamma_{i,\mathrm{t}}^{c} \tag{2}$$

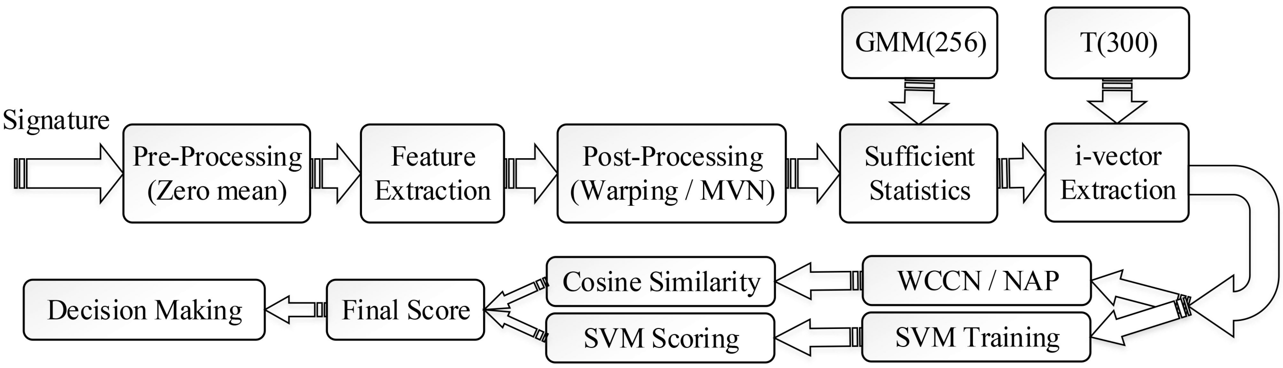

Fig. 1 *Block diagram of the proposed i-vector-based online SV system*

$$F_c(X_i) = \sum_t \gamma^c_{i,\mathrm{t}}(X_{i,\mathrm{t}} - m_c), \tag{3}$$

where $X_{i,\mathrm{t}}$ shows the tth vector of entire features for signature ith, m_c is the mean of cth component, and $\gamma^c_{i,\mathrm{t}}$ is the posterior probability of generating $X_{i,\mathrm{t}}$ by the cth component as follows:

$$\gamma^c_{i,\mathrm{t}} = \Pr(c \mid X_{i,\mathrm{t}}) = \frac{w_c \mathcal{N}(X_{i,\mathrm{t}} \mid m_c, \Sigma_c)}{\sum_{j=1}^{C} w_j \mathcal{N}(X_{i,\mathrm{t}} \mid m_j, \Sigma_j)}. \tag{4}$$

2.3 i-Vector

Let M show the individual dependent mean-supervector that represents the feature vectors of a signature. The term supervector is referred to the DC-dimensional vector obtained by concatenating the D-dimensional mean vectors of the GMM corresponding to a given signature (it can be obtained by classical maximum a posteriori (MAP) adaptation [39]). In the i-vector method [29], this supervector is modelled as follows:

$$M = m + Tw, \tag{5}$$

where m is an individual independent mean-supervector derived from the UBM, T is a low rank matrix, and w is a random latent variable having a standard normal distribution. The i-vector ϕ is the MAP point estimate of the variable w which is equal to the mean of the posterior probability of w given the input signature. In this setting, it is assumed that supervector M has a Gaussian distribution with mean m and covariance matrix TT^{t}.

2.4 Training the parameters of the model

In (5), m and T are the parameters of the model. Usually, the mean-supervector of the UBM is used as m. This supervector is formed by concatenating the means of the UBM components [44]. To train T, the expectation maximisation (EM) algorithm is used [38]. Let the UBM have C components and the dimensions of feature vectors be D. First, the matrix Σ is formed as follows:

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 & \dots & 0 \\ 0 & \Sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Sigma_C \end{bmatrix} \tag{6}$$

where Σ_c is the covariance matrix of the cth component of the UBM. Assuming X_i shows the entire collection of feature vectors for signature ith and $P(X_i \mid M_i, \Sigma)$ denotes the likelihood of X_i calculated with the GMM specified by the supervector M_i and the super-covariance matrix Σ, then the EM optimisation is done by repeating the following two steps:

1. For each training signature, we use the current value of T and compute the vector that maximises the likelihood in the following way:

$$w_i = \underbrace{\arg\max}_{w} P(X_i \mid m + Tw, \Sigma) \tag{7}$$

2. Then we update T by maximising the following equation:

$$\prod_i P(X_i \mid m + Tw_i, \Sigma), \tag{8}$$

By taking the logarithm of (8), the product is replaced with summation and also the likelihood is replaced with log-likelihood which can be calculated for each signature using the following equation: (see (9)) ,where c iterates over all components of the model and t iterates over all feature vectors. T_c is a submatrix of T related to the cth component.

Assuming we have computed the zero and first-order statistics using (2) and (3), we can compute the posterior covariance matrix [i.e. $\mathrm{Cov}(w_i, w_i)$], mean (i.e. $\mathbb{E}[w_i]$) and the second moment (i.e. $\mathbb{E}[w_i w_i^{\mathrm{t}}]$) for w_i using the following relations:

$$\mathrm{Cov}(w_i, w_i) = \left(I + \sum_c N_c(X_i) T_c^{\mathrm{t}} \Sigma_c^{-1} T_c\right)^{-1} \tag{10}$$

$$\mathbb{E}[w_i] = \mathrm{Cov}(w_i, w_i) \sum_c T_c^{\mathrm{t}} \Sigma_c^{-1} F_c(X_i) \tag{11}$$

$$\mathbb{E}[w_i w_i^{\mathrm{t}}] = \mathrm{Cov}(w_i, w_i) + \mathbb{E}[w_i]\mathbb{E}[w_i]^{\mathrm{t}} \tag{12}$$

Finally, if we maximise (8), the following relation is obtained for updating matrix T:

$$T_c = \left(\sum_i F_c(X_i)\mathbb{E}[w_i]^{\mathrm{t}}\right)\left(\sum_i N_c(X_i)\mathbb{E}[w_i w_i^{\mathrm{t}}]\right)^{-1} \tag{13}$$

2.5 Computing the i-vector

As explained in the previous section, w is a random hidden variable with standard normal distribution, where i-vector is the mean of the posterior probability of w given the input signature. To find i-vector, the MAP point estimation of w is used and the formula is the same as (11).

2.6 Methods for reducing the effects of intra-class variations

Several methods have been proposed for reducing the effects of intra-class (within class) variations. For i-vector-based method, the widely used such methods are NAP [29, 44–46], WCCN [29, 47, 48], and linear discriminant analysis (LDA) [29]. Here, we used WCCN and NAP which will be explained in the following section.

2.6.1 Within-CCN: In the WCCN method, we try to find a linear transformation by which we can reduce intra-class variations. To achieve this, we first compute the intra-class covariance matrix using the following relation:

$$S_{\mathrm{w}} = \frac{1}{S}\sum_{s=1}^{S} \frac{1}{N_s}\sum_{n=1}^{N_s} (w_s^n - \bar{w}_s)(w_s^n - \bar{w}_s)^{\mathrm{t}}, \tag{14}$$

where S is the total number of classes, N_s is the number of training samples in class s, w_s^n is the nth sample in class s, and $\bar{w}_s = \frac{1}{N_s}\sum_{n=1}^{N_s} w_s^n$ is the mean of class s. Then, the transform matrix $B \in R^{N \times K}$ can be calculated using the Cholesky decomposition of intra-class covariance matrix:

$$S_{\mathrm{w}}^{-1} = BB^{\mathrm{t}}. \tag{15}$$

Finally, the samples in the new space are calculated using $y = B^{\mathrm{t}}w$.

$$\log(P(X_i \mid m + Tw_i, \Sigma)) = \sum_c \Bigg(\mathrm{N}_c \log \frac{1}{(2\pi)^{D/2}|\Sigma_c|^{1/2}} - \frac{1}{2}\sum_t (X_{i,\mathrm{t}} - T_c w_i - m_c)^t \Sigma_c^{-1} (X_{i,\mathrm{t}} - T_c w_i - m_c)\Bigg), \tag{9}$$

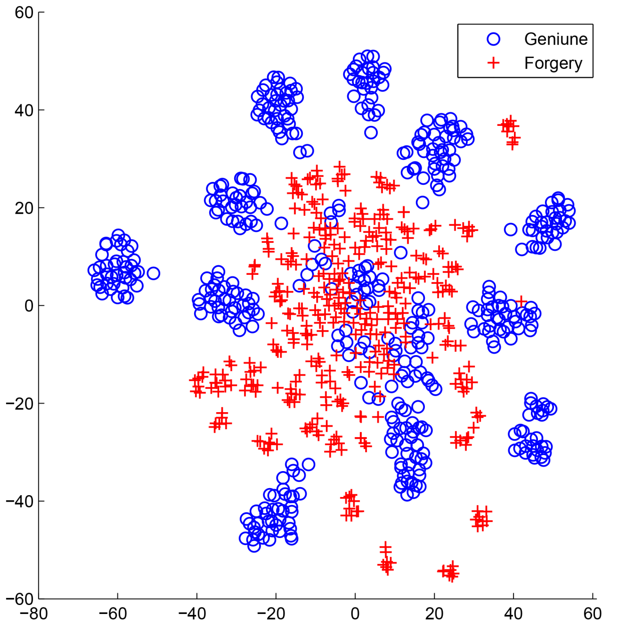

Fig. 2 *t-SNE plot of individuals' signatures for 11 signature types of SigWiComp2013 training set using raw i-vectors (i.e. without applying any transformation)*

2.6.2 Nuisance AP: The aim of this method is to find a transformation by which we can omit the subspaces that cause noisy variations. The transform matrix is calculated using the following relation:

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^{\mathrm{t}}, \tag{16}$$

where $\boldsymbol{R} \in \mathbb{R}^{N \times K}$ is a low rank rectangular matrix, where its columns are the K eigenvectors of $\boldsymbol{S}_{\mathrm{w}}$ with maximum corresponding eigenvalues. After finding transform matrix $\boldsymbol{P} \in \mathbb{R}^{N \times N}$, we use it to project all samples to the new space.

3 Proposed method

The proposed method aims at using the i-vector for online SV. In this method, we first train a GMM using all signatures (i.e. both genuine and forged signatures) in the training set. The forged signatures are used to achieve better modelling of the forged signatures space. After training this UBM, we use it to extract zero and first-order statistics of the training features. Then, using these statistics we train an i-vector extractor using several iterations of the EM algorithm explained in Section 2.4.

After training the i-vector extractor, we extract i-vectors from all signatures in the training set. In this stage, we have extracted several i-vectors for each individual in the training set and we use them to train the intra-class variation reduction methods. Specifically, we train NAP and WCCN transforms from the original signatures in the training set and use them to transform the i-vectors to the new space.

3.1 Creating a template for each individual

After training the required transforms, a template is created for each individual based on her/his reference signatures. To achieve this, first, we extract i-vectors from the signatures and transform them using the trained models (i.e. WCCN and NAP). Then, the average of the reference i-vectors is used as the template for each individual (or representative vector). This is a conventional method in the speaker and language recognition fields based on i-vector [29, 31]. Therefore, we have a representative i-vector for each individual, which we will use for scoring.

3.2 Scoring (similarity measure)

To calculate the similarity between the representative i-vector and the test signature we proceed as follows. First, we extract an i-vector from the test signature and project it to the new space using the trained transforms. Then, we use the cosine similarity to compute the score between this i-vector and each individual's representative i-vector

$$\mathrm{Cos\ Sim}(\boldsymbol{w}_{\mathrm{template}}, \boldsymbol{w}_{\mathrm{test}}) = \frac{\langle \boldsymbol{w}_{\mathrm{template}}, \boldsymbol{w}_{\mathrm{test}} \rangle}{\| \boldsymbol{w}_{\mathrm{template}} \| \| \boldsymbol{w}_{\mathrm{test}} \|}, \tag{17}$$

where the numerator and denominator show the inner product and the product of the i-vectors' norm-2, respectively.

3.3 Decision making

After finding the cosine distance score, a simple threshold is applied to it to make the final decision of accepting/rejecting a verification attempt. If the score was higher than the threshold, the test signature is verified and otherwise it is rejected. We can use different thresholds for each individual or we can assume a global threshold for everyone. In this paper, we used a global threshold to be able to plot the detection error tradeoff (DET) curves. In addition to this simple approach, more advanced learning algorithms such as SVM modelling or LDA can be applied to the i-vectors.

3.4 Decision making based on a binary SVM classifier

The combination of WCCN/NAP and cosine similarity scoring is a common approach to the i-vector preconditioning and decision making in the speaker recognition area. As our experimental results show this approach also works for the SV relatively well. In ASV task, the decision-making step can also be considered as a supervised binary classification process that categorises the input feature vectors as either forged or genuine. Therefore, in SV, we propose to train a binary SVM classifier for each individual to differentiate between her/his forged and genuine signatures. To train an SVM for each individual, we will be in need of her/his forged data samples which are non-trivial in practical applications. Even though in speaker verification, other speakers' data can serve as forged data to train the binary classifier for each speaker properly; in SV, this is not the case.

To overrule this issue, we propose to train an individual independent binary SVM. In this case, there will be one binary SVM that works for all individuals in the evaluation database. Before going through details of the proposed individual independent binary SVM, results of an intuitively related experiment are discussed. Fig. 2 demonstrates the individuals' signatures for 11 signature types of SigWiComp2013 training set in the i-vectors space without applying any transformation using t-distributed stochastic neighbor embedding (t-SNE) [49] method.

As you can see from Fig. 2, genuine signatures have mostly concentrated around some distinctive clusters, but forged signatures have scattered in the space. Hence, it is not possible to train a binary classifier to categorise these two classes in this space properly. Fig. 3 shows the same plot after removing the bias of each signature type (i.e. for each individual, subtracting the mean of only her/his genuine signatures from all her/his signatures). It seems that after this kind of bias subtraction, the genuine and forgery signature samples are perfectly separable. Motivated by this observation, we propose to perform the same procedure for each individual of the training set, and then an individual independent binary SVM is trained using all these bias-removed genuine and forgery data to classify the genuine signatures from forged signatures in the evaluation time. Specifically, in the evaluation time, first the mean of each individual's reference signatures is subtracted from all her/his test signatures, and finally the individual independent SVM classifier categorises these bias-removed signatures.

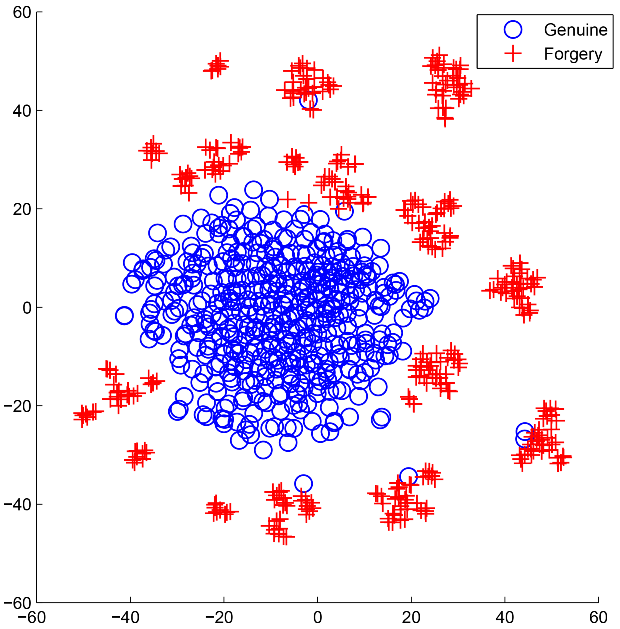

Fig. 3 *t-SNE plot of individuals' signatures for 11 signature types of SigWiComp2013 training set when the bias of each signature type is removed*

4 Feature extraction

4.1 Preprocessing

As proposed in [21] the centres of masses of all signatures are normalised. This is done by setting the mean across x and y axes to zero. The means are calculated using the following equation:

$$[\bar{x}, \bar{y}]^{\mathrm{t}} = \frac{1}{N}\sum_{n=1}^{N}[x_n, y_n]^{\mathrm{t}} \tag{18}$$

4.2 Features

There are many features that are used for online SV. We use several of them which are listed below:

- Vertical and horizontal positions.
- Path-tangent angle.
- Path velocity magnitude.
- Log curvature radius.
- Total acceleration magnitude.

Aside from the features listed above, the first-order derivative of them is used too. This derivative is calculated according to the second-order regression formula in [50].

4.3 Post-processing

To normalise the features, we used two types of post-processing techniques separately.

4.3.1 Mean and variance normalisation: In this technique, the features are normalised, so that the mean and variance across each feature dimension is zero and one, respectively. The mean and variance are calculated for each signature separately. This is a very common method in speaker verification [29, 51].

4.3.2 Feature warping: This technique is more commonly used in speech processing applications, but has been recently employed in online SV as well [52]. The purpose of this technique is to normalise the features, so that they have a standard normal Gaussian distribution. This is done using a moving window where we used a window of size 31.

5 Experiment setup

5.1 Database

Several databases are available for online SV [53–56] that reduce the barrier in this research field. In these databases, the most common signature dynamics features are the position information of the pen, though some of them have pressure information as well. One of the most authentic databases is the online Japanese signature data set which was used in SigWiComp2013 [54]. These signatures have been collected using the HP EliteBook 2730p tablet. Each signature in this database is represented by a sequence of triples. The first two elements in each triple show the position of the pen and the third element shows the state of the pen (whether it is lifted or not). The sampling rate for these signatures is 200 Hz and the resolution is 50 px per centimetre. There are 31 individuals (i.e. signature types) in this database where for each individual there are 42 genuine signature examples and 36 forged signature examples. This database is divided into training and evaluation sets as follows:

Training set: There are 11 signature types (i.e. one signature for each individual) in this set that for each of which there are 42 genuine and 36 forged signature examples. This set is used to train the UBM and i-vector extractor and also for computing the NAP and WCCN transforms as well as for training the applied SVM model which is used in the decision-making stage.
Evaluation set: This set includes 20 signature types for each of which there are 12 reference signatures. These 12 signatures are used to create the reference template for each individual. Apart from these reference signatures, there are 66 test signatures for each signature type which consists of 30 genuine signatures and 36 forged signatures.

Almost all evaluations are carried out on the SigWiComp2013 database. To verify the performance of the proposed method, the final results have been reported in the SVC2004 database as well [55]. The SVC2004 database contains 80 signature types which have been captured by a graphic tablet (WACOM Intuos). It has been divided into two tasks, each of which has 40 signature types. There are 20 genuine and 20 skillfully forged signatures for each signature type, where the first ten genuine signatures are considered as the reference set. We employ task one for models training and task two for evaluation.

5.2 Evaluation criteria

There are two types of mistakes in a typical SV system: a false rejection which happens when a genuine signature is incorrectly classified as forged by someone and a false acceptance is when a forgery signature is considered as signed by the corresponding individual. Several evaluation metrics are available to report the test results. In this paper, we report the equal error rate (EER), i.e. the FRR and false acceptance rate (FAR) when they are equal. In addition, in some cases, the DET curves are plotted to allow for more detailed comparison.

5.3 t-SNE visualisation in the i-vector space

In the SV task, the best representation is the one that perfectly discriminates between genuine and skillfully forged signatures for each individual. In addition, in order to prevent random forgery attacks, the representation also must discriminate between individual signatures. Fig. 4 shows t-SNE plots for genuine and forged signatures for ten individuals in the test set of SigWiComp2013 using raw i-vectors (i.e. without any transforms).

It is clear from Fig. 4 that for almost all individuals, the i-vector representation discriminates between genuine and forged signatures properly; therefore, it is predictable that using a good classifier (e.g. non-linear or even linear) an acceptable performance is achievable. Fig. 5 shows the t-SNE plot for the genuine signatures of the first ten individuals from SigWiComp2013. It is obvious that in this case, the individual signatures are more separated than in the

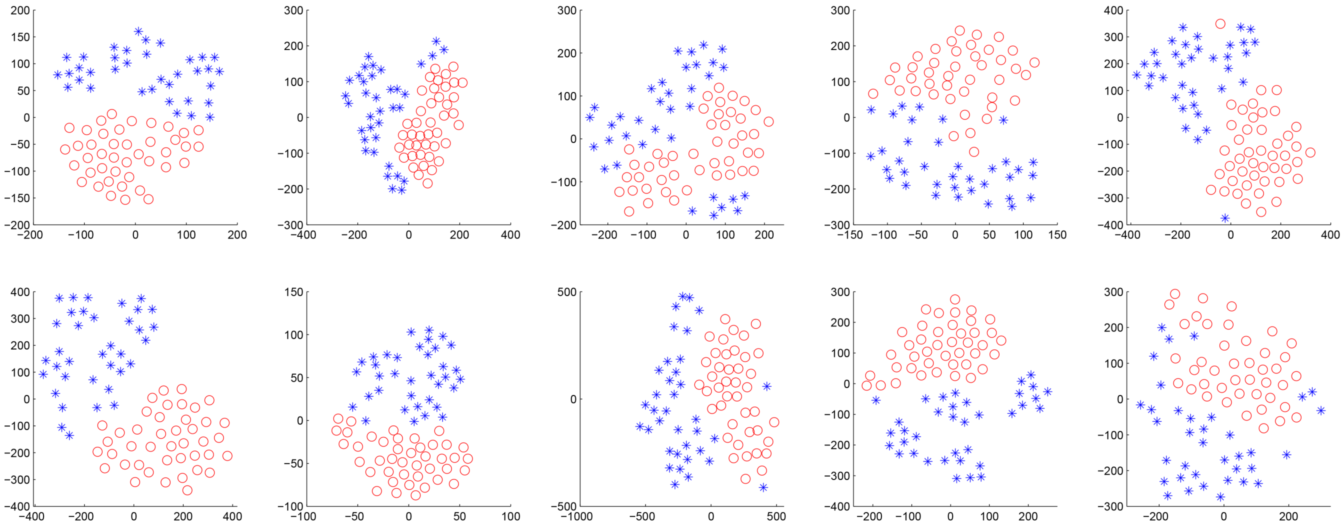

Fig. 4 *t-SNE plots of genuine and forged signatures for the ten first individuals of SigWiComp2013 test set using raw i-vectors. The red circles show the genuine signatures and the blue stars show the forged signatures*

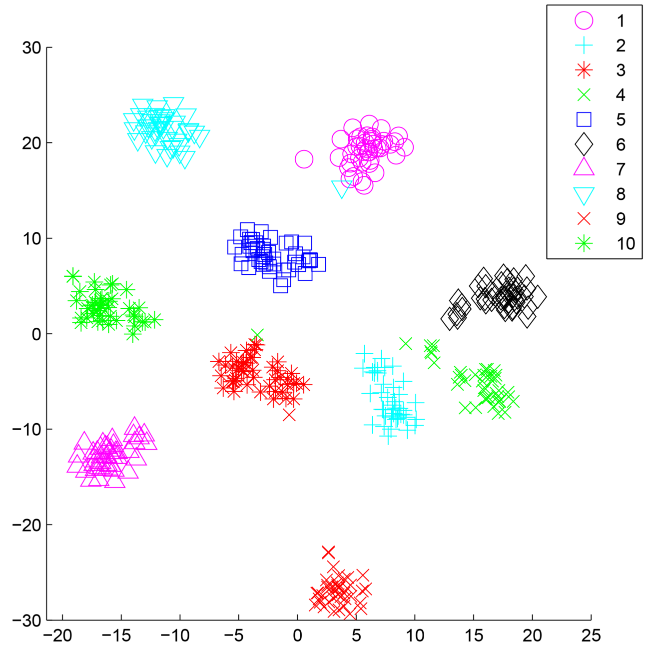

Fig. 5 *t-SNE plot of genuine signatures of the first ten individuals of SigWiComp2013 test set using raw i-vectors*

previous case; therefore, random forgeries rejection would not be a serious concern.

6 Experiments results

6.1 Effects of UBM components count, i-vectors dimensionality, and features post-processing

The first series of experiments were performed to investigate the effects of the number of UBM components, the dimension of i-vectors, and also post-processing techniques. These experiments were carried out for two techniques of feature normalisation separately. Table 1 represents the obtained EERs. In each element of this table, there are two EER values (separated by a slash) that represent the effect of using the feature warping and mean and variance normalisation (MVN) techniques.

First of all, EER values in this table indicate that in most cases the feature warping technique leads to smaller errors than MVN (i.e. the numbers on the right-hand side of slash). Consequently, it is considered as the default post-processing technique from now on. Furthermore, UBM with 64 components is found to have the best EER values mainly for both post-processing techniques. Finally, it is evident that applying i-vectors with size 100–120 resulted in better EER values. One of the factors that affect i-vector dimensionality is the length of the input signal. In the text-independent speaker verification task, the dimensions of i-vectors are considered in the range of 400–600. However, in text-prompted speaker verification task where input signals are significantly shorter, a size around 200 is applied [41]. Regarding signatures, since the signal length is usually short; therefore, an appropriate i-vector size should be <200 which our results confirm it.

Table 1 EER comparison of two post-processing techniques using various UBM component counts and i-vector dimensions

Dimension	Mixture count		
	64	128	256
60	12.71/14.69	13.37/14.52	14.17/16.50
70	11.94/14.19	12.08/14.03	14.69/15.97
80	12.64/13.61	12.87/14.36	13.70/16.25
90	12.87/13.19	13.04/14.03	14.36/15.56
100	12.54/**12.87**	12.92/13.19	14.36/16.39
110	**11.25/12.87**	12.64/14.03	13.53/16.25
120	11.53/13.04	12.38/13.70	13.86/14.72
130	11.39/13.20	12.54/13.06	13.53/15.28
140	12.05/14.03	12.38/13.53	14.03/13.86
150	11.72/14.03	11.94/12.87	13.61/13.06
160	12.50/13.33	12.87/13.53	13.06/14.52

To compare feature warping and MVN techniques more intuitively, DET curves were plotted for the best cases [UBM with 64 components and 110-dimensional (110D) i-vectors] of these two techniques in Fig. 6. According to this figure, it should be noted that when the cost of false acceptance is more than false rejection, the MVN technique performs better than feature warping predominantly.

6.2 NAP and WCCN transforms comparison

This series of experiments are going to evaluate the effect of applying WCCN or NAP transforms for intra-class variations reduction and using no-transform at all. In each cell of Table 2, EER value of not using a transform (on the left-hand side) has been separated from the value of applying the NAP transform on the right-hand side. The results of applying the WCCN transform appear on the left-hand side of each cell of Table 1.

By comparing the results of these two transforms (i.e. WCCN and NAP) from Tables 1 and 2, respectively, we conclude that employing each of these two variation reduction techniques can significantly increase the performance of our proposed method for SV. Furthermore, in most cases, the WCCN transform results in

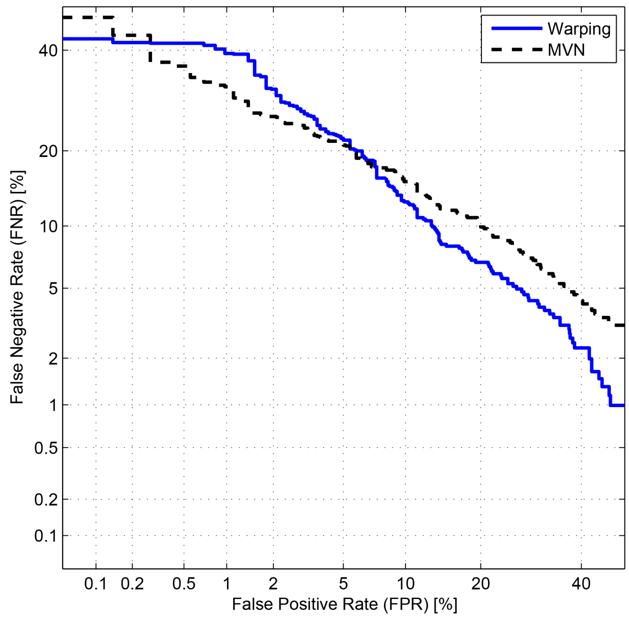

Fig. 6 *DET curve comparison for the feature warping and MVN techniques (UBM with 64 components and 110D i-vectors)*

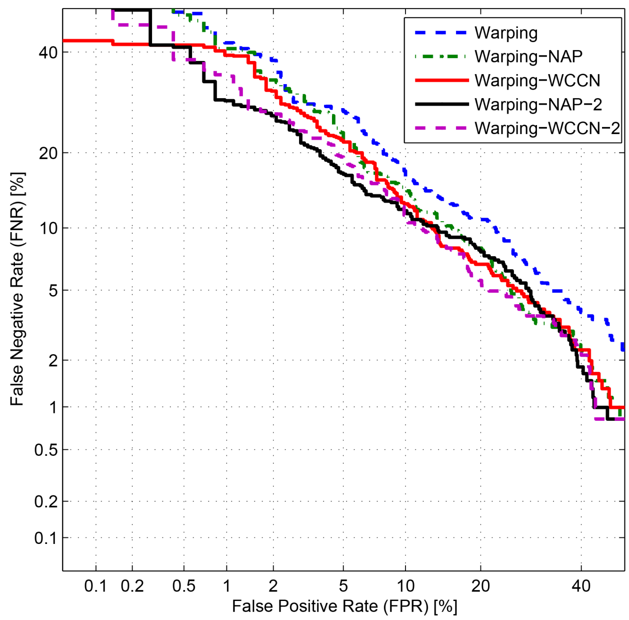

Fig. 7 *DET curve comparison for WCCN, WCCN-2, NAP, NAP-2, and no-transform. In each case, results are reported using the best parameters configuration*

lower EER values than the NAP. This observation is consistent with the reported results in speaker verification domain [29]. Finally, the results indicate that these two transformation techniques perform better for the middle range i-vector dimension (i.e. 100–120). According to our experiments, the larger i-vectors can improve the discrimination ability between the different individuals' signatures, while they are likely to lead to a reduction in the effectiveness of the WCCN/NAP transforms, because already the classes have been well separated in the original space (i.e. before applying any transformation).

6.3 Using reference signatures in WCCN and NAP transforms

In all experiments so far performed, only 11 signature types of the training set had a contribution in the WCCN and NAP transforms calculation. This means we applied the same transforms for all individuals. In this experiment, for each individual in the evaluation set, distinct WCCN and NAP transforms are computed

Table 2 EER comparison based on UBM components count and i-vector dimension in no-transform/NAP format

Dimension	Mixture count		
	64	128	256
60	16.25/13.37	17.66/12.54	15.42/14.85
70	16.81/12.78	17.08/12.50	16.01/14.69
80	16.53/13.47	16.50/13.37	15.56/14.36
90	15.51/13.86	15.14/13.86	15.51/13.53
100	15.28/13.06	15.84/13.37	15.28/14.52
110	14.52/11.81	15.28/12.54	14.86/14.17
120	15.18/12.36	15.18/14.36	15.35/13.53
130	15.00/**11.72**	15.00/13.86	14.86/13.37
140	14.72/12.22	14.86/13.33	14.31/13.70
150	14.58/11.88	15.00/13.19	**13.70**/13.04
160	14.31/12.36	14.85/12.87	**13.70**/12.71

All systems used feature warping.

Table 3 EER comparison based on UBM components count and i-vector dimension in WCCN-2/NAP-2 format

Dimension	Mixture count		
	64	128	256
60	12.05/14.03	13.37/14.86	13.20/14.52
70	11.72/14.17	11.67/11.22	12.71/12.36
80	12.22/12.71	11.81/12.21	12.50/11.94
90	11.81/13.70	11.72/11.72	12.64/13.04
100	12.08/13.53	11.53/12.21	13.47/13.06
110	**10.56**/13.06	11.94/12.05	12.38/12.38
120	11.53/11.88	12.05/11.81	12.71/12.78
130	10.97/12.71	12.05/12.05	12.21/11.88
140	10.97/12.50	12.05/12.71	11.81/10.97
150	10.89/11.06	10.73/12.21	10.73/11.39
160	11.39/11.55	11.22/11.72	11.22/**10.89**

using her/his reference signatures and the 11 signature types of the training set altogether. Therefore, each distinct WCCN and NAP transforms for each individual uses 12 signature types. The results of these experiments are shown in Table 3. We named these new transforms WCCN-2 and NAP-2.

From Table 3, we conclude that the individual dependent WCCN transforms (i.e. one WCCN transform for each individual) result in some gain. However, the amount of gain is different for different cases. Similar to conventional WCCN results in Table 1, the best EER value was achieved by using a 64-component UBM and 110D i-vectors.

On the other hand, gains for the NAP-2 transform were not consistent in all cases. In some cases, one transform for all individuals leads to improvement in the EER, while in most cases individual dependent transform scheme obtains some gain (i.e. NAP-2). Overall, applying distinct transforms leads to improvement in EER in NAP transform case as well.

Fig. 7 demonstrates the best DET curve of the different transform techniques.

6.4 Effects of sampling rate reduction

As explained in Section 5.1, the SigWiComp database has a sampling rate of 200 Hz. However, there are many other databases with a sampling rate of 100 Hz. The purpose of the experiments in this section is to investigate the effects of sampling rate reduction on the performance of the proposed method. Therefore, before extracting features from the signatures, they were downsampled to 100 Hz and then all the previous experiments were performed on them. Table 4 shows the comparative results for different cases. To make this table smaller and clearer, only the results related to UBM with 64 components and 110D i-vectors are shown.

The results in Table 4 show that in most cases, on the contrary, the downsampling improves the performance. The poorest results

Table 4 EER comparison for 100 and 200 Hz sampling rates

Method	Sampling rate	
	100 Hz	200 Hz
no-transform	14.58	14.52
NAP	11.72	11.81
WCCN	10.73	11.25
NAP-2	9.74	13.06
WCCN-2	**9.31**	10.56

These results come from UBM with 64 components and i-vectors with 110 dimensions.

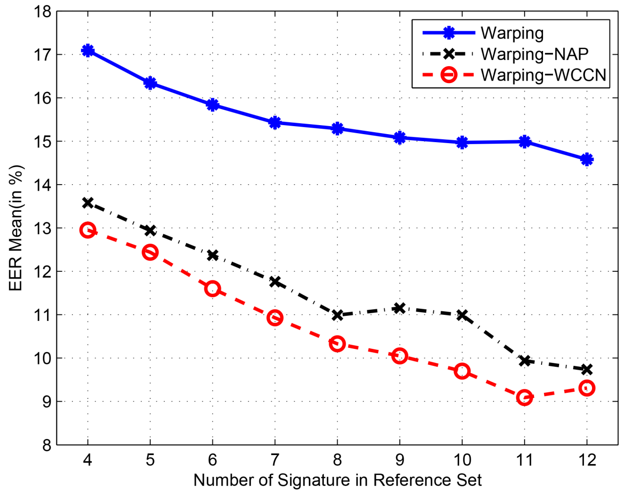

Fig. 8 *Mean of EERs of the proposed method for different sizes of the reference set. In each case, the experiment was repeated 50 times with random subset selection of the main reference set*

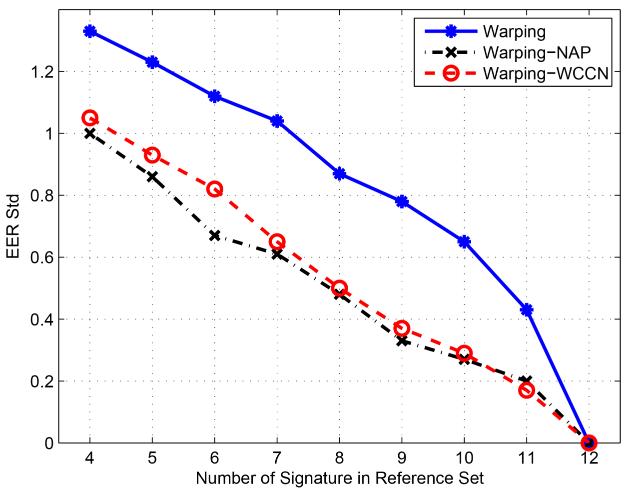

Fig. 9 *Standard deviation of EERs of the proposed method for different sizes of the reference set. In each case, the experiment was repeated 50 times with random subset selection of the main reference set*

are for the cases where no-transform was used and the best improvements are obtained in cases where distinct transforms were used. Similar to the previous experiments, the best result was achieved for the WCCN case with distinct transforms (WCCN-2).

6.5 Effects of reference set size on the performance

This experiment aims to investigate the effect of the number of signatures in an individual's reference set on the performance. In the SigWiComp database, there are 12 reference signatures for each individual. Here, we randomly selected an *n*-element subset of that set and used it for computing the template for each individual. We repeated this experiment for 50 times and computed the mean and standard deviation of EER. Figs. 8 and 9 show the plots for mean and standard deviation obtained from this experiment. These results were obtained using an UBM with 64 components, 110D i-vectors, and the sampling rate of 100 Hz. As it can be seen from these figures, in almost all cases, the increase in the size of the reference set decreases both the mean and standard deviation of the EERs.

6.6 Results of the proposed SVM-based techniques

In this section, we report the results of the proposed SVM-based decision-making approach. Here, similar to the WCCN-2/NAP-2 techniques, each individuals' reference set is added to the SVM training data to train an individual dependent SVM model, named SVM-2. Table 5 represents the results of applying conventional SVM and SVM-2-based techniques. By comparing these EER values with corresponding values in Table 1, we realised that both SVM-based techniques outperform the baseline approach. SVM-2 performed about 0.2% absolutely better than the conventional SVM, though this improvement is not as much as the improvement made by WCCN-2/NAP-2 versus WCCN/NAP. Overall, the absolute EER improvements made by SVM and SVM-2 are about 2 and 2.2%, respectively; therefore, we conclude that the proposed SVM-based techniques are better than the WCCN/NAP techniques in the SV domain.

6.7 Comparison with other methods

In this section, we compare the results of our i-vector-based method with results of several other methods. Generally, there are two approaches for comparing results of different studies. The first is to implement all methods and report the results. The other approach is to compare only the results that are obtained by evaluating a standard database with a fixed testing scenario. In cases where the testing conditions are the same for all studies, the second approach is better and fairer. That is because often there are subtleties in many studies which are not clearly explained in the papers and this makes it difficult to obtain the same results reported in this paper. Therefore, we used the second approach and compared our results with the results reported in the 2013 contest [54]. To make the comparison fairer, for each method we selected the worst case, the average case, and the best case from the achieved results. Table 6 shows these comparative results. In this table, results are reported based on accuracy, FAR, and FRR unlike the previous results.

The results in Table 6 clearly show that the proposed method is significantly better than other methods in practise. As a summary, SVM has reduced the error 56% in the worst case and 68% in the best case relatively.

6.8 Results on SVC2004 database

In this section, we are going to report the results of the proposed method on the SVC2004 database to assess its generalisability on other databases. For this database, we only report the results of the best method which were obtained using SVM-based techniques. Table 7 shows these results for both SVM and SVM-2.

Results of Table 7 make it clear that the bigger models perform better. This is slightly in contrast with our previous results on the SigWiComp2013 database, where the middle range models performed better. We believe the main reason for that is the number of individuals in the training data. The bigger model needs more training data to estimate its parameters properly. Here, there are 40 individuals in the training data, whereas there are 11 individuals in the previous database. Therefore, the parameter estimation has been performed better here, and consequently better performance has been achieved using the biggest model.

Furthermore, less improvement has been obtained by adding individual reference set to training data. This is due to the size of training data as well. For the SigWiComp2013 database, adding data would increase the size of training data about 5%, whereas for SVC2004 the increase would be <1%. Consequently, the effect of adding reference set is not so much.

Table 5 EER comparison based on UBM components count and i-vector dimension for the proposed SVM method in SVM/SVM-2 format

Dimension	Mixture count		
	64	128	256
60	12.2/12.1	10.7/10.4	11.1/10.9
70	12.4/12.1	10.1/10.0	10.4/10.4
80	11.6/11.4	10.6/10.1	10.4/10.3
90	10.9/10.6	9.57/9.24	9.44/9.41
100	10.4/10.3	9.24/9.17	9.24/9.24
110	9.90/9.57	9.41/9.03	10.3/10.1
120	9.90/9.57	9.03/**8.75**	9.41/9.41
130	9.90/9.44	9.57/9.08	9.24/9.17
140	9.74/9.58	8.89/**8.75**	9.08/8.89
150	10.4/10.2	9.03/8.91	8.89/**8.75**
160	9.90/9.90	9.31/8.91	9.17/9.17

All systems used feature warping.

Table 6 Comparison results of the proposed method with three other methods on the same database [54]

Method	Accuracy	FAR	FRR
Qatar University	70.55	30.22	29.56
Sabanci University-1	72.55	27.37	27.56
Sabanci University-2	72.47	27.50	27.56
NAP-2 worst case	84.92	15.14	15.02
NAP-2 average	87.62	12.41	12.35
NAP-2 best case	89.06	10.97	10.89
WCCN-2 worst case	86.58	13.47	13.37
WCCN-2 average	88.05	11.99	11.91
WCCN-2 best case	89.37	10.69	10.56
SVM-2 worst case	87.93	12.08	12.05
SVM-2 average	90.22	9.81	9.75
SVM-2 best case	**91.25**	**8.75**	**8.75**

6.9 Comparison with others reported results on the SVC2004 database

Similar to the SigWiComp database, here we compare the proposed method with several other methods on the SVC2004 database. The comparison results are shown in Table 8. Note that some of the methods only used five signatures as the reference set, so their results are not fairly comparable with the other methods.

Here also the proposed method outperforms all the others, but its EER differences with them are not as much as in the SigWiComp2013 database. The main reason for that is the average length of signatures in these two databases. The average length of signature for SVC2004 is about 230 sample points, whereas this number for SigWiComp2013 is about 470 sample points (after downsampling to 100 Hz). In text-independent speaker verification, it has been proved that the i-vector method performs better when signals are longer. It seems that it is also true for SV. On the other hand, the DTW-based method cannot align the whole two long signals perfectly because of error propagation. When an error happens in the alignment, which is likely, it could be propagated to next samples and affect the overall performance. This is not probable to happen for the i-vector method because it does not perform any hard alignment such as DTW.

7 Conclusion

In recent years, i-vectors have achieved the best results for speaker verification. In this paper, we aimed at adopting this method for the application of online SV. As a result, we proposed a method based on i-vector which achieved better results compared with previous methods on the SigWiComp2013 database. In this method, we used the techniques of NAP and WCCN to reduce intra-class variations which improved the results remarkably. In addition, we further improved the results by using separately computed feature transformations for each individual.

Table 7 EER comparison based on UBM components count and i-vector dimension for the proposed SVM-based technique on SVC2004 database in SVM/SVM-2 format

Dimension	Mixture count		
	64	128	256
50	7.75/7.38	8.25/7.78	7.38/7.13
100	7.78/7.50	7.75/7.38	6.63/6.38
150	6.94/6.94	6.63/6.61	5.56/5.25
200	7.22/7.22	6.50/6.39	5.38/5.38
250	7.22/7.22	6.25/6.13	5.28/5.28
300	7.09/6.15	6.25/6.11	**5.00/5.00**

Table 8 Comparison of the results of the proposed method with several others reported results on the SVC2004 database

Method	Number of samples	EER
DTW [17]	5	6.96
HMM [21]	5	6.90
DTW + HMM [57]	5	10.9
wavelet packet [58]	5	6.65
wavelet transform + DCT [59]	10	6.37
length norm + fractional distance [18]	10	5.82
DCT + sparse representation [9]	10	5.61
proposed SVM-2	10	5.00

Second column shows the number of samples used as the individual's reference set.

We also proposed a 2-class SVM-based technique to discriminate genuine and forged signatures which improved the performance of the i-vector method considerably. In this case also applying individual dependent SVM models resulted in some gain in the EER value. On SigWiComp2013 database, this method achieved 8.75% EER that is the best reported result on this database so far.

To verify the consistency of improvement of the proposed i-vector-based method on other databases, the SVC2004 database was also evaluated. On this database, we obtained some improvement over other reported results, but not as much as the SigWiComp2013 database. The individual dependent SVM-based method reaches 5% EER in the best case. The reason for smaller improvements on SVC2004 is the length of signatures. The average length of signatures in SVC2004 is about half of the SigWiComp2013 database. It has been proved that i-vector works better for longer signals, and here we verified that is also correct for SV. Finally, we concluded that the i-vector-based method is able to achieve the best results for long signatures.

One of the main advantages of the proposed method is that it represents each signature with a fixed-length vector. This enables us to cope with this problem such as other classification problems in machine learning which in turn will lead to newer methods being proposed for this problem. Furthermore, possible future work would be to investigate the use of a deep neural network classifier.

The final note about the proposed method is its speed. In the proposed method, in the test time, after extracting some features from the input signature, i-vector is computed using a few matrix multiplications. Scoring and decision making are also fast, because they are a simple cosine similarity measuring and a thresholding. Thus, the proposed method is suitable for the practical applications, where speed is critical. DTW-based methods usually compare the input signature with all reference signatures which makes it much slower than our proposed method. The i-vector-based method, in the worst case (i.e. SVM-based method), needs about 30 ms on average for one SV, whereas for the linear DTW method (i.e. fast version) this number is about 3200 ms.

8 Acknowledgment

The authors thank Muhammad Imran Malik for providing them with the data set to do the experiments.

9 References

[1] Li, S.Z., Jain, A.: '*Encyclopedia of biometrics*' (Springer Publishing Company, Incorporated, New York, US, 2015, 2nd edn.)

[2] Plamondon, R., Lorette, G.: 'Automatic signature verification and writer identification: the state of the art', *Pattern Recognit.*, 1989, **22**, (2), pp. 107–131

[3] Impedovo, D., Pirlo, G.: 'Automatic signature verification: the state of the art', *IEEE Trans. Syst. Man Cybern. C*, 2008, **38**, (5), pp. 609–635

[4] Kalera, M.K., Srihari, S., Xu, A.: 'Offline signature verification and identification using distance statistics', *Int. J. Pattern Recognit. Artif. Intell.*, 2004, **18**, (07), pp. 1339–1360

[5] Singh, J., Sharma, M.: 'Offline signature verification using neural networks', *i-Manager's J. Inf. Technol.*, 2012, **1**, (4), p. 35

[6] Daramola, S.A., Ibiyemi, T.S.: 'Offline signature recognition using hidden Markov model (HMM)', *Int. J. Comput. Appl.*, 2010, **10**, (2), pp. 17–22

[7] Fierrez-Aguilar, J., Nanni, L., Lopez-Peñalba, J., *et al.*: 'An on-line signature verification system based on fusion of local and global information'. Audio and Video-based Biometric Person Authentication, 2005, pp. 523–532

[8] Nanni, L.: 'An advanced multi-matcher method for on-line signature verification featuring global features and tokenised random numbers', *Neurocomputing*, 2006, **69**, (16), pp. 2402–2406

[9] Liu, Y., Yang, Z., Yang, L.: 'Online signature verification based on DCT and sparse representation', *IEEE Trans. Cybern.*, 2015, **45**, (11), pp. 2498–2511

[10] Jain, A.K., Griess, F.D., Connell, S.D.: 'On-line signature verification', *Pattern Recognit.*, 2002, **35**, (12), pp. 2963–2972

[11] Lee, L.L., Berger, T., Aviczer, E.: 'Reliable online human signature verification systems', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1996, **18**, (6), pp. 643–647

[12] Nanni, L., Lumini, A.: 'Ensemble of Parzen window classifiers for on-line signature verification', *Neurocomputing*, 2005, **68**, pp. 217–224

[13] Lei, H., Govindaraju, V.: 'A comparative study on the consistency of features in on-line signature verification', *Pattern Recognit. Lett.*, 2005, **26**, (15), pp. 2483–2489

[14] Richiardi, J., Ketabdar, H., Drygajlo, A.: 'Local and global feature selection for on-line signature verification'. 2005 Proc. Eighth Int. Conf. Document analysis and recognition, 2005, pp. 625–629

[15] Nanni, L.: 'Experimental comparison of one-class classifiers for online signature verification', *Neurocomputing*, 2006, **69**, (7), pp. 869–873

[16] Lejtman, D.Z., George, S.E.: 'On-line handwritten signature verification using wavelets and back-propagation neural networks'. 2001 Proc. Sixth Int. Conf. Document Analysis and Recognition, 2001, pp. 992–996

[17] Kholmatov, A., Yanikoglu, B.: 'Identity authentication using improved online signature verification method', *Pattern Recognit. Lett.*, 2005, **26**, (15), pp. 2400–2408

[18] Vivaracho-Pascual, C., Faundez-Zanuy, M., Pascual, J.M.: 'An efficient low cost approach for on-line signature recognition based on length normalization and fractional distances', *Pattern Recognit.*, 2009, **42**, (1), pp. 183–193

[19] Sato, Y., Kogure, K.: 'Online signature verification based on shape, motion, and writing pressure'. Proc. Sixth Int. Conf. Pattern Recognition, 1982, pp. 823–826

[20] Martens, R., Claesen, L.: 'Dynamic programming optimisation for on-line signature verification'. 1997 Proc. Fourth Int. Conf. Document Analysis and Recognition, 1997, vol. **2**, pp. 653–656

[21] Fierrez, J., Ortega-Garcia, J., Ramos, D., *et al.*: 'HMM-based on-line signature verification: feature extraction and signature modeling', *Pattern Recognit. Lett.*, 2007, **28**, (16), pp. 2325–2334

[22] Dolfing, J., Aarts, E., Van Oosterhout, J.: 'On-line signature verification with hidden Markov models'. 1998 Proc. Fourteenth Int. Conf. Pattern Recognition, 1998, vol. **2**, pp. 1309–1312

[23] Van, B.L., Garcia-Salicetti, S., Dorizzi, B.: 'On using the Viterbi path along with HMM likelihood information for online signature verification', *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2007, **37**, (5), pp. 1237–1247

[24] Rúa, E.A., Castro, J.L.A.: 'Online signature verification based on generative models', *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2012, **42**, (4), pp. 1231–1242

[25] Yang, L., Widjaja, B., Prasad, R.: 'Application of hidden Markov models for signature verification', *Pattern Recognit.*, 1995, **28**, (2), pp. 161–170

[26] Richiardi, J., Drygajlo, A.: 'Gaussian mixture models for on-line signature verification'. Proc. 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, 2003, pp. 115–122

[27] Miguel-Hurtado, O., Mengibar-Pozo, L., Lorenz, M.G., *et al.*: 'Online signature verification by dynamic time warping and Gaussian mixture models'. 2007 41st Annual IEEE Int. Carnahan Conf. Security Technology, 2007, pp. 23–29

[28] Humm, A., Hennebert, J., Ingold, R.: 'Gaussian mixture models for chasm signature verification'. Machine Learning for Multimodal Interaction, 2006, pp. 102–113

[29] Dehak, N., Kenny, P., Dehak, R., *et al.*: 'Front-end factor analysis for speaker verification', *IEEE Trans. Audio Speech Lang. Process.*, 2011, **19**, (4), pp. 788–798

[30] Dehak, N., Torres-Carrasquillo, P.A., Reynolds, D.A., *et al.*: 'Language recognition via i-vectors and dimensionality reduction'. InterSpeech, 2011, pp. 857–860

[31] Martınez, D., Plchot, O., Burget, L., *et al.*: 'Language recognition in i-vectors space'. InterSpeech, 2011, pp. 861–864

[32] Bahari, M.H., Saeidi, R., Van Leeuwen, D., *et al.*: 'Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech'. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 7344–7348

[33] Xia, R., Liu, Y.: 'Using i-vector space model for emotion recognition'. InterSpeech, 2012

[34] Khaki, H., Erzin, E.: 'Continuous emotion tracking using total variability space'. InterSpeech, 2015

[35] Eghbal-zadeh, H., Lehner, B., Dorfer, M., *et al.*: 'CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks', 2016

[36] Kenny, P., Boulianne, G., Ouellet, P., *et al.*: 'Joint factor analysis versus eigenchannels in speaker recognition', *IEEE Trans. Audio Speech Lang. Process.*, 2007, **15**, (4), pp. 1435–1447

[37] Kenny, P., Ouellet, P., Dehak, N., *et al.*: 'A study of interspeaker variability in speaker verification', *IEEE Trans. Audio Speech Lang. Process.*, 2008, **16**, (5), pp. 980–988

[38] Kenny, P., Boulianne, G., Dumouchel, P.: 'Eigenvoice modeling with sparse training data', *IEEE Trans. Speech Audio Process.*, 2005, **13**, (3), pp. 345–354

[39] Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: 'Speaker verification using adapted Gaussian mixture models', *Digit. Signal Process.*, 2000, **10**, (1), pp. 19–41

[40] Zeinali, H., Mirian, A., Sameti, H., *et al.*: 'Non-speaker information reduction from cosine similarity scoring in i-vector based speaker verification', *Comput. Electr. Eng.*, 2015, **48**, pp. 226–238

[41] Zeinali, H., Kalantari, E., Sameti, H., *et al.*: 'Telephony text-prompted speaker verification using i-vector representation'. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4839–4843

[42] Zeinali, H., Sameti, H., Burget, L., *et al.*: 'i-vector/HMM based text-dependent speaker verification system for RedDots challenge'. InterSpeech, 2016, pp. 440–444

[43] Zeinali, H., Sameti, H., Burget, Č.J., *et al.* 'Text-dependent speaker verification based on i-vectors, deep neural networks and hidden Markov models', *Comput. Speech Lang.*, 2017, **46**, pp. 53–71

[44] Campbell, W.M., Sturim, D.E., Reynolds, D.A., *et al.*: 'SVM based speaker verification using a GMM supervector kernel and NAP variability compensation'. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2006, pp. 97–100

[45] Solomonoff, A., Quillen, C., Campbell, W.M.: 'Channel compensation for SVM speaker recognition'. Odyssey – The Speaker and Language Recognition Workshop, 2004, vol. **4**, pp. 219–226

[46] Solomonoff, A., Campbell, W.M., Boardman, I.: 'Advances in channel compensation for SVM speaker recognition'. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2005, pp. 629–632

[47] Hatch, A.O., Kajarekar, S.S., Stolcke, A.: 'Within-class covariance normalization for SVM-based speaker recognition', InterSpeech, 2006, p. 1874

[48] Dehak, N., Kenny, P., Dehak, R., *et al.*: 'Support vector machines and joint factor analysis for speaker verification'. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2009, pp. 4237–4240

[49] Maaten, L.V.D., Hinton, G.: 'Visualizing data using t-SNE', *J. Mach. Learn. Res.*, 2008, **9**, (Nov), pp. 2579–2605

[50] Young, S., Evermann, G., Gales, M., *et al.*: '*The HTK book*', vol. **2** (Cambridge University Press, Cambridge University, London, UK, 1997)

[51] Zeinali, H., Sameti, H., Burget, L.: 'HMM-based phrase-independent i-vector extractor for text-dependent speaker verification', *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2017, **25**, (7), pp. 1421–1435

[52] Nautsch, A., Rathgeb, C., Busch, C.: 'Bridging gaps: an application of feature warping to online signature verification'. 2014 Int. Carnahan Conf. Security Technology (ICCST), 2014, pp. 1–6

[53] Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., *et al.*: 'MCYT baseline corpus: a bimodal biometric database', *IEE Proc., Vis. Image Signal Process.*, 2003, **150**, (6), pp. 395–401

[54] Malik, M.I., Liwicki, M., Alewijnse, L., *et al.*: 'ICDAR 2013 competitions on signature verification and writer identification for on-and offline skilled forgeries (SigWiComp2013)'. 2013 12th Int. Conf. Document Analysis and Recognition (ICDAR), 2013, pp. 1477–1483

[55] Yeung, D.-Y., Chang, H., Xiong, Y., *et al.*: 'SVC2004: first international signature verification competition', Biometric Authentication, 2004, pp. 16–22

[56] Kholmatov, A., Yanikoglu, B.: 'SUSIG: an on-line signature database, associated protocols and benchmark results', *Pattern Anal. Appl.*, 2009, **12**, (3), pp. 227–236

[57] Fierrez-Aguilar, J., Krawczyk, S., Ortega-Garcia, J., *et al.*: 'Fusion of local and regional approaches for on-line signature verification Advances in Biometric Person Authentication, 2005, pp. 188–196

[58] Wang, K., Wang, Y., Zhang, Z.: 'On-line signature verification using wavelet packet'. 2011 Int. Joint Conf. Biometrics (IJCB), 2011, pp. 1–6

[59] Nanni, L., Lumini, A.: 'A novel local on-line signature verification system', *Pattern Recognit. Lett.*, 2008, **29**, (5), pp. 559–568