Information Sciences 352–353 (2016) 188–202

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Two strategies to optimize the decisions in signature verification with the presence of spoofing attacks

Shilian Yu[a,b], Ye Ai[a,b], Bo Xu[a,b], Yicong Zhou[c], Weifeng Li[a,b,*], Qingmin Liao[a,b], Norman Poh[d]

[a] Department of Electronic Engineering/Graduate School at Shenzhen, Tsinghua University, China
[b] Shenzhen Key Laboratory of Information Science and Technology, Shenzhen, China
[c] Department of Computer and Information Science, University of Macau, Macau, China
[d] Department of Computer Science, University of Surrey, Guildford, Surrey, United Kingdom

ARTICLE INFO

Article history:
Received 19 January 2015
Revised 23 February 2016
Accepted 4 March 2016
Available online 10 March 2016

Keywords:
Biometric authentication
Spoofing attack
Liveness score
Brute-force
Probabilistic

ABSTRACT

A conventional biometric authentication system is often designed to distinguish genuine accesses from zero-effort impostor attacks. However, when operating in an adversarial environment, the system has to be robust against presentation attacks such as spoofing. An effective solution to reduce the impact of spoofing attack is to consider both the matching score and liveness score when making the accept/reject decision. In this paper, we consider the joint decision space of matching and liveness scores in the presence of *both* spoofing attack and zero-effort attack, with application to signature verification. Our investigation aims to understand how decision thresholds in the above space should be optimized. This leads to two dichotomies of methods, namely brute-force approach versus probabilistic approach; and single threshold versus double-threshold approach. This view leads to three novel methods that have never been reported. Based on the experimental results carried out on an off-line signature database, the novel methods turn out to outperform simpler methods with only matching score.

1. Introduction

Biometric authentication now permeates our daily lives, from automatic border control to unlocking our smart phones. Along with its convenience, potential vulnerabilities and various kinds of attacks have been reported. One of the most commonly reported attacks is *spoofing* attack, which is also called *non-zero effort* attack. In a spoofing attack, an attacker tries to cheat the biometric system to gain illegitimate access by producing fake biometric traits of the victim that he/she tries to impersonate. Since a conventional biometric authentication system is designed to distinguish genuine samples from *zero-effort* impostor samples, the system is often unable to judge whether a submitted sample is a *live* one or is made from a spoofed material. Examples of spoofing attack abound for various biometric modalities: gummy fingers in fingerprint authentication [15], face prints in 2D face authentication [13], masks in 3D face authentication [8], synthesized voice forgery [17], etc.

* Corresponding author at: Shenzhen Key Laboratory of Information Science and Technology, Shenzhen, China. Tel.: +86 755 26036564.
E-mail address: li.weifeng@sz.tsinghua.edu.cn (W. Li).

http://dx.doi.org/10.1016/j.ins.2016.03.005
0020-0255/

Signature as one of the most popular biometric traits plays an important role in person recognition. It has many applications such as bank cheques authentication, attendance monitoring, and endorsement or confirmation of documents, often in a legally binding manner. For some applications, especially those where the identity of the signer needs to be ascertained, automatic signature verification is indisputably the most natural biometric modality that can fulfil this role.

According to the data acquisition mechanism, signature verification systems can be divided into online verification and off-line verification. Online systems have a higher signature recognition rate because of the dynamic information including writing speed, stroke length and pen pressure. However, the system requires special electronic devices to capture these information at the same time of writing. On the other hand, off-line systems are designed to compare a static image with a template stored in the database without any dynamic information. Compared to the online systems, off-line systems are more practical because they do not require the presentation of signers or any equipment. Thus, we focus on the off-line signature verification in this paper.

Saikia and Sarma [21] define three basic types of forgeries in signature verification systems; they are random forgery, simple forgery and skilled forgery. "Spoofing" in this paper is associated with *skilled* forgery, where the forger has access to the samples of the genuine signature and thus he/she is able to reproduce it.

To deal with the spoofing attacks in biometric systems, we use two very common threshold-based schemes to make the final decision. One of them simply makes the decision from the matching score d and the liveness score s by thresholding the two measurements:

$$\text{decision}(d,s) = \begin{cases} accept & \text{if } d < T_d \quad \text{and} \quad s > T_s \\ reject & \text{otherwise}, \end{cases} \tag{1}$$

where T_d is a threshold applied to d and T_s is a threshold applied to s. Since we consider the distance between the template and the query sample as our matching score d, the less d is, the higher the degree of matching is.

An alternative way is to estimate the posterior probability that the sample is genuine or a *match* comparison, and that it is also from a live sample, $P(M=1, L=1|d,s)$ based on the measurements d and s. It is then straightforward to apply a threshold to this posterior probability in order to make the final decision:

$$\text{decision}(d,s) = \begin{cases} accept & \text{if } P(M=1, L=1|d,s) > P_T \\ reject & \text{otherwise}, \end{cases} \tag{2}$$

where $M=1$ and $L=1$ synergistically represent the traits of this sample being genuine and authentic. More details about our terminology will be discussed in Section 3. This approach is also referred to as *single threshold probabilistic approach*.

A similar variant, which is referred to as *double threshold probabilistic approach*, is to optimize two thresholds for $P(M=1|d)$ and $P(L=1|s)$ respectively, as shown in Eq. (3).

$$\text{decision}(d,s) = \begin{cases} accept & \text{if } P(M=1|d) > P_{Td} \quad \text{and} \\ & P(L=1|s) > P_{Ts} \\ reject & \text{otherwise}. \end{cases} \tag{3}$$

Although the methodology is general, we will conduct our experiments on off-line signature authentication as a case study.

In order to make the final accept/reject decision in biometric systems using a matching score and a liveness score, from the introductory presentation so far, we can identify two decision schemes to optimizing the decision threshold, namely, brute-force and probabilistic optimization. The first strategy consists of exhaustively searching for an optimal solution by minimizing a performance criterion. The second capitalizes on the logic construct of Eq. (1) but in probabilistic sense. Although both strategies rely on the same logic construct, the brute-force optimization does not commit to an assumption that its probabilistic version does; that is, the latter assumes that both the matching score and the liveness score are independent of each other. Since there are two approaches, i.e. single threshold and double threshold in the probabilistic strategy, we have to systematically investigate all three different methods, as represented by the three equations above. The effectiveness of these methods will be systematically compared in the presence of both spoofing attack and zero-effort attack.

In this paper, we argue that although complicated methods have been proposed in the literature, one should not dismiss simple, yet straightforward strategies such as threshold-based methods because they are extremely easy to implement. Therefore, the key question is not about implementation, but about finding out if there is an optimal way to optimize the thresholds. Our key contribution is, therefore, to advance the understanding of the nature of thresholds optimization; and to provide recommendations and practices with regards to their implementation. Our second contribution is to propose two probabilistic variant of thresholding schemes. We conjecture that these variants are useful and should compare them favourably with the plain thresholding strategy because probability axioms can handle the inherent uncertainty in the choice of thresholds.

This paper is organized as follow: Section 2 describes related work in signature verification and the liveness detector used in some biometric traits. Section 3 illustrates some concepts and the notation used in our study. Section 4 presents our study methodology. Section 5 presents a case study on signature verification followed by conclusions in Section 6.

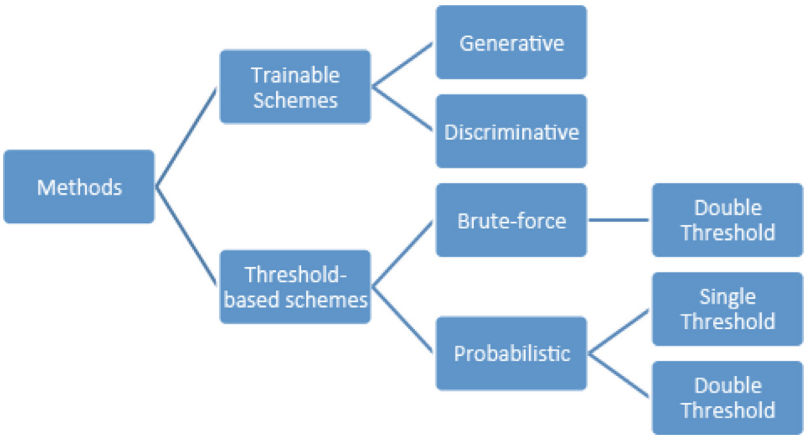

Fig. 1. Taxonomy of the fusion strategies of the matching score and the liveness score.

2. Related work

We shall cover two bodies of literature related to our study here, one on the problem of combining liveness and matching scores; and another on the off-line signature verification problem that is used as a case study.

2.1. Combining liveness and matching scores

Recently, the question of how to counteract spoofing attacks has grabbed serious attention of biometric system designers. In order to cope with spoofing attacks, one approach is to use a multimodal biometric system that combines information coming from different biometric traits [2,4,5,20]. Recent studies, however, show that a multimodal system may still be vulnerable to spoof attacks when one or more of its modalities are compromised. Another approach is to design a liveness detection module [7] that aims to judge whether or not a submitted sample is live. The module output is a liveness score which allows the system integrator to make the final accept/reject decision along with the matching score.

With the development of liveness detection, this leads a new variant of fusion problem that aims to combine both the biometric output (matching score) and the liveness score. Fig. 1 shows a taxonomy of the fusion strategies. These strategies can be broadly categorized as trainable and threshold-based scheme. A trainable scheme deploys a classifier in order to combine the two scores. We survey some of the methods below.

Trainable schemes. Marasco et al. [14] designed and assessed four different ways of combining liveness measure values with matching scores and deployed a Bayesian Belief Network (BBN) to model the relationship between these two scores. Rattani and Poh [19] proposed a Bayesian classifier which was implemented using a mixture of Gaussians, Gaussian Copula, and Quadratic Discriminant function. Instead of using BBN, Chingovska et al. [3] employed logistic regression as a discriminative classifier. These trainable schemes are shown in the upper half of Fig. 1.

Threshold-based schemes. To our best knowledge, due to the obvious implementation, there is no study addressing the effectiveness of threshold-based implementation. From our analysis in the introduction section, it transpires that there is a brute-force method and a probabilistic method; and each of which can be further refined into single versus double-threshold optimization strategies.

2.2. Offline signature verification

Offline signature verification has been studied in some literatures. As a typical pattern recognition system, it has the following steps: Data Acquisition, Preprocessing, Feature Extraction, Classification(also called Verification). To distinguish the genuine signature from forgeries in offline signature verification system, many approaches are proposed.

Ahmad et al. [1] used a Hidden Markov Model (HMM) to build a reference model for each local feature. There were three statistical layers in the phase of verification. A Bayesian inference technique was then used to decide whether to accept a given sample as being genuine or not. The experiments conducted on random and spoofing reported a False Acceptance Rate of 22% and 37% respectively for both attacks.

Vargas et al. [23] proposed an approach based on grey level information using texture features. Their method used a co-occurrence matrix which was further represented by Local Binary Pattern (LBP) as a feature extraction method. Genuine signatures and random forgeries were then used to train an SVM model. When the model was tested with random and spoofing samples their method was reported to achieve an Equal Error Rate (EER) of 12.82% on spoofing signatures.

Fang et al. [9] described an approach to detect spoof forgeries in offline signature verification. Based on a smoothness criterion, their findings suggested that spoofed signatures mostly contained cursive graphic patterns that were less smooth

Table 1
The four classes of a biometric verification system and the desirable actions to take when considering spoofing attacks.

Action	Source of origin	Liveness state	Attack type
Accept	Same(M = 1)	Live(L = 1)	Genuine
Reject	Different(M = 0)	Live(L = 1)	Zero-effort(Random)
Reject	Same(M = 1)	Replica(L = 0)	Spoofing(Forgeries)
Reject	Different(M = 0)	Replica(L = 0)	Improbable

than the genuine ones on a detailed scale. The authors then proposed a smoothness index from such signatures and combined it with other global features for verification.

More recent research results in offline signature verification can be referred in [6,12,25].

3. Problem statement and terminology

This section lays the foundation for the basic concepts and terminology used in our study.

A biometric system typically produces a similarity score by comparing a template and a query sample. This comparison produces a somewhat complex relationship, depending on two factors: (1) the source of origin of the biometric trait and (2) the liveness state. *Source of origin* is defined as the provenance of a biometric trait, *regardless of its liveness state*, i.e., both concepts are independent. By the *liveness state*, we mean that the sample can either be considered a live one or a fake one.

We avoid the terms *match* and *nonmatch* that are commonly used in biometric comparison because these terms are associated with the assumption or connotation that the pair of biometric traits being compared are live. Therefore, in our terminology, a comparison between a stored template and a spoofing query sample of a victim is considered the same source of origin because both samples came from the same biometric trait.

All samples in our biometric authentication system have two scores and two labels:

- Matching score $d \in \mathbb{R}$. d judges whether the content of a sample is from the same source of origin. We interpret the score d as a distance score, such that a low value implies a same source of origin whereas a high value implies a different one.
- Liveness score $s \in \mathbb{R}$. Liveness score s judges whether this sample is an autograph or a replica from an impersonator. The bigger s is, the lower the probability of spoofing is.
- $M \in \{1, 0\}$ is the state of source of origin which can either be the *same* source, or two *different* sources, *regardless of the liveness states of the sample pair*. For example, a spoofing sample is considered the same source of origin since it comes from the same original biometric trait that the victim's.
- $L \in \{1, 0\}$ is the state of liveness; and a sample is either a *live* one or a *replica*.

More specifically, in the case of signature authentication, where our goal is to compare a signature reference with a query sample, the source of origin is considered the same if both of them display the signature of the same person. Conversely, if signatures on the reference and the query sample are from two different people, then, their sources are considered different. A sample is referred to as a live one if it is the authentic signer who has produced the signature. On the other hand, a sample is considered a spoof or a replica, if it has been produced by a skilled forger trying to imitate the genuine signature.

The term "liveness" is borrowed from the fingerprint liveness detection literature [24]. In behavioral biometrics, samples are dynamic and it is arguably easier to produce fake samples by skillful forgers. Therefore, all signatures would be considered "live" in the usual sense of the word. However, we prefer to keep to the same terminology because the methods being proposed are applicable to the general fusion problem of matching and liveness scores.

For each sample, a *joint observation* of d and s is a result of the following two dichotomies of events: same versus different sources; and live versus replicated samples. These dichotomies can then be classified into one of the four classes according to the joint observation. As shown in Table 1, there is a desirable action that to take for each of these four classes.

For convenience, we use *random* samples to represent zero-effort samples in our offline signature verification experiments.

We will not consider the fourth class of attack in our experiments because in authentication, an attacker has all the incentives to gain unauthorised access. By signing differently, the source of origin (of the query sample) will be classified as different (from the reference signature) by a conventional biometric system. As a result, this reduces to a zero-effort attack in our case. Although this does not generalize to fingerprint authentication where a fabricated material can be used, the access request is likely to be denied by a biometric system for the improbable case, hence, sparing the need for liveness detection.

4. Proposed approach

In this section we present three threshold optimization strategies, namely the brute-force approach, and two probabilistic methods.

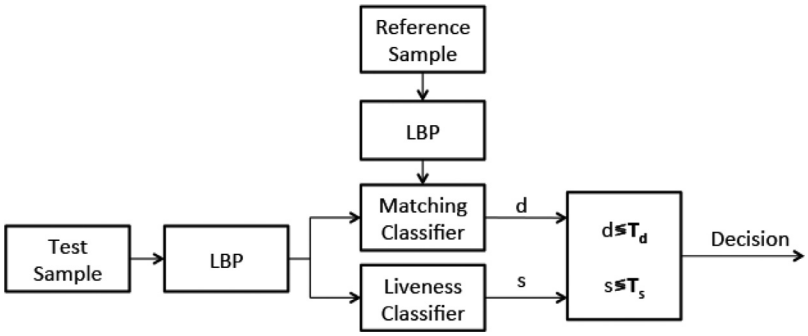

Fig. 2. Flow chart of off-line signature verification system with double threshold brute-force optimization algorithm.

Before presenting them, let us introduce the matching score here, as well as the performance metrics; the discussion on liveness score will be deferred to Section 5.2.2.

Matching score d of a query sample feature Q is obtained by calculating the distance between Q and the reference sample feature R. Since Q and R are histograms, a commonly used metric to characterize their degree of similarity (or rather difference) is by the chi-square distance. The mathematical formula of chi-square distance is as follows:

$$d(Q,R) = \sum_{i=1}^{I} \frac{(Q_i - R_i)^2}{Q_i + R_i}, \tag{4}$$

where I represents the dimension of sample feature. In order to obtain the liveness score, we train a liveness classifier using AdaBoost. More details can be found in Section 5.2.2.

In this case, class of positive samples is defined by *genuine*, which corresponds to the samples of state *same* and *live*. The remaining classes include the random samples and spoofing samples, as shown in Table 1, are collectively referred to as the negative class. Thus, false acceptance rate (FAR) is defined by:

$$\text{FAR} = \frac{\text{number of wrongly classified negative examples}}{\text{number of total examples}} = f(t);$$

whereas false rejection rate (FRR) by:

$$\text{FRR} = \frac{\text{number of wrongly classified positive examples}}{\text{number of total examples}} = g(t).$$

We note that FAR and FRR are functions of the decision threshold, so as the metrics derived from them. $f(t)$ is a monotonically decreasing function of a decision threshold t while $g(t)$ is a monotonically increasing function based on t. We adopt half total error rate (HTER), defined as the average of FAR and the FRR:

$$\text{HTER} = \frac{\text{FAR+FRR}}{2} = \frac{f(t) + g(t)}{2} = h(t)$$

HTER is also a function of t, denoted as $h(t)$. We use HTER as our optimization criterion.

Although HTER is used here, other criterion, such as equal error rate (EER), weighted error rate (WER), etc, can also be used. EER finds the thresholds that make FAR equal with FRR. WER [18] weighs FAR and FRR in different proportions. One can recognize that WER is a more general criterion than HTER because the latter weighs the two errors in equal proportions.

Typically (as used by our experiments below), the data set is divided into three partitions: the training set (S_{tr}) for liveness classifier, the optimization set (S_{op}) for optimizing thresholds and the testing set (S_{te}). From the training set, positive and negative features are obtained in order to train a liveness classifier using AdaBoost. The optimization set is used to find the optimal accept/reject decision threshold by minimizing HTER (or EER). Finally, we apply the threshold found on optimization set S_{op} to the testing set S_{te} in order to measure the performance on the unseen test data set.

4.1. Brute-force approach

In the brute-force optimization, we simply search all possible threshold pairs in the space $\mathcal{D} \times \mathcal{S}$, spanned by the matching score d and the liveness score s, for the solution that minimizes HTER. The system architecture of our particular implementation is shown in Fig. 2; and the general optimization strategy is shown in Algorithm 1. Note that this optimization strategy is sensitive to the proportion between the random and spoof samples. However, since this proportion is not known, an agnostic strategy is to set two attack types to equal proportion during the optimization. In this way, the resultant optimal decision thresholds will satisfy the attack-agnostic assumption, as required.

Algorithm 1 Brute-force optimization.

```
HTER_min = ∞
for d ∈ 𝒟 do
  for s ∈ 𝒮 do
    HTER = Evaluate performance with (d, s)
    if HTER_min ≥ HTER then
      HTER_min = HTER
      T_d = d
      T_s = s
    end if
  end for
end for
return (T_d, T_s)
```

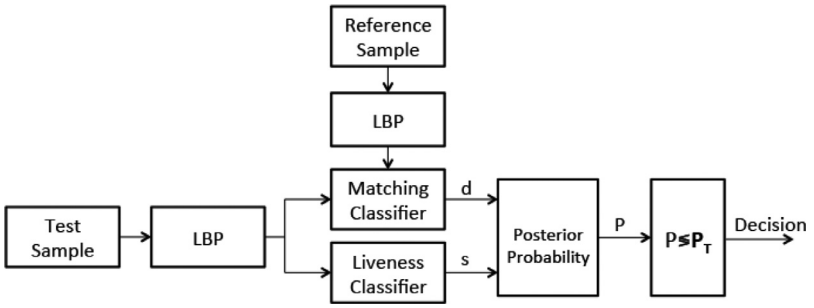

Fig. 3. Flow chart of off-line signature verification system with single threshold probabilistic optimization algorithm.

4.2. Probabilistic approach by single threshold

In the single-threshold probabilistic approach, we estimate the posterior probability of a query sample being from the same source and live under its matching score d and liveness score s, i.e. $P(M=1, L=1|d,s)$. Fig. 3 illustrates a realization of the architecture implementing the single-threshold probabilistic method. By assuming that the posterior probability representing same source of origin and the probability of being live are independent, the above probability $P(M=1, L=1|d,s)$ can be expressed by

$$P(M=1, L=1|d,s) = P(M=1|d)P(L=1|s). \tag{5}$$

The two posterior probabilities can be further calculated using the Bayesian theorem:

$$P(M=1|d) = \frac{P(d|M=1)P(M=1)}{P(d)}, \tag{6}$$

$$P(L=1|s) = \frac{P(s|L=1)P(L=1)}{P(s)}. \tag{7}$$

The prior probabilities $P(M=1)$ and $P(L=1)$ can be easily obtained by counting the proportion of their corresponding samples in training data. In contrast, we need to estimate the probability density functions (PDFs) to obtain the normalizing constants $P(d)$ and $P(s)$ and the likelihoods $P(d|M=1)$ and $P(s|L=1)$. During optimization stage, we first obtain the d and s of all samples in S_{op}. Then we count the proportion of corresponding samples to obtain the prior probabilities $P(M=1)$ and $P(L=1)$. After that, the histograms of d, s, d conditioned on the samples being same and s conditioned on the samples being live are calculated. Finally, cubic spline interpolation is used to estimate the probability density functions. Cubic spline interpolation is a data-interpolation technique which uses many separate cubic polynomials to obtain a piecewise continuous curve. The prior probabilities and the four PDFs will be used in the testing stage as well. Similar to brute-force optimization, we search from 0 to 1 to select the optimal threshold P_T. Algorithm 2 summarizes the single-threshold probabilistic procedure.

4.3. Probabilistic approach by double threshold

Similar to the brute-force and the single threshold probabilistic optimization scheme, we also propose a double threshold probabilistic optimization scheme, the architecture of which is shown in Fig. 4. The difference between this scheme and the single threshold probabilistic one is that, in this scheme, we optimize two thresholds: P_{Td} for $P(M=1|d)$ and P_{Ts} for $P(L=1|s)$. This algorithm is shown in Algorithm 3.

Algorithm 2 Single threshold probabilistic optimization.

calculate $P(M=1)$, $P(L=1)$
estimate PDFs $P(d)$, $P(s)$, $P(d|M=1)$ and $P(s|L=1)$
obtain $P(M=1, L=1|d, s)$
$\text{HTER}_{min} = \infty$
for $P \in [0, 1]$ **do**
  HTER = Evaluate performance with P
  if $\text{HTER}_{min} \geq \text{HTER}$ **then**
    $\text{HTER}_{min} = \text{HTER}$
    $P_T = P$
  end if
end for
return P_T

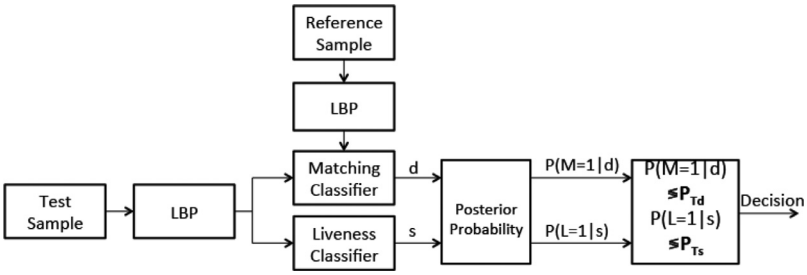

Fig. 4. Flow chart of off-line signature verification system with double threshold probabilistic optimization algorithm.

Algorithm 3 Double threshold probabilistic optimization.

calculate $P(M=1)$, $P(L=1)$
estimate PDFs of $P(d)$, $P(s)$, $P(d|M=1)$ and $P(s|L=1)$
obtain $P(M=1|d)$ and $P(L=1|s)$
$\text{HTER}_{min} = \infty$
for $P_d \in [0, 1]$ **do**
  for $P_s \in [0, 1]$ **do**
    HTER = Evaluate performance with P_d and P_s
    if $\text{HTER}_{min} \geq \text{HTER}$ **then**
      $\text{HTER}_{min} = \text{HTER}$
      $P_{Td} = P_d$
      $P_{Ts} = P_s$
    end if
  end for
end for
return (P_{Td}, P_{Ts})

Table 2
The main properties of the off-line signature database GPDS960GRAYsignature.

Type	Persons	Samples /person	Interval	Forgers	Samples /forger	Total
Genuine	881	24	A Single Day	0	0	21144
Imitation	881	30	A Single Day	10	3	26317

5. Experiments

5.1. Database

The two proposed approaches were evaluated on the offline signature database GPDS960GRAY signature. The main characteristics of the database are summarized in the Table 2. It consists of 21,144 genuine signatures and 26,317 imitations, totally 47,485 signatures. The data is from 881 individuals, each of whom contributes 24 genuine signatures, plus about 30 forged signatures for each of them. The 24 genuine specimens of each signer were collected in a single-day writing sessions.

Fig. 5. An example of samples in database. (a) genuine sample. (b) spoofing sample 1. (c) spoofing sample 2. (d) random sample.

Table 3
The different experimental settings of different approaches.

Approach	Classifier	Observation	Threshold	Dataset
Single threshold brute-force (Conventional)	Matching	d	T_d	S_{op}, S_{te}
Double threshold brute-force (Proposed)	Matching, Liveness	d, s	T_d, T_s	S_{tr}, S_{op}, S_{te}
Single threshold probabilistic (Conventional)	Matching	$P(M=1\|d)$	P_{Td}	S_{op}, S_{te}
Single threshold probabilistic (Proposed)	Matching, Liveness	$P(M=1,L=1\|d,s)$	P_T	S_{tr}, S_{op}, S_{te}
Double threshold probabilistic (Proposed)	Matching, Liveness	$P(M=1\|d), P(L=1\|s)$	P_{Td}, P_{Ts}	S_{tr}, S_{op}, S_{te}

The forgeries were produced from the static image of the genuine signature. Each forger was allowed to practice the signature for as long as he/she wished. Each forger imitated 3 signatures of 5 signers in a single-day writing session. The genuine signatures shown to each forger are chosen randomly from the 24 genuine ones. Therefore for each genuine signature there are 30 skilled forgeries made by 10 forgers from 10 different genuine specimens. Fig. 5 illustrates the signatures of a random selected person in the database. While the first sample is genuine, the other two samples are forgeries, and the last one is a random sample of another person that is paired to the first sample for the purpose of simulating a random attack.

5.2. Experimental setup

In our experiments, we select 300 signers from the database. Every signer has 24 genuine signatures and about 30 forgeries. In order to evaluate the performance of our approaches in the presence of both spoofing attack and zero-effort attack, for each signer, we select 30 genuine signatures from other signers to simulate zero-effort attacks; these samples are also referred to as *random* samples. For consistency, we will adopt the terminology of *genuine* samples, *spoofing* samples and *random* samples in the following discussion, as shown in Table 1.

As discussed in Section 4, the proposed approaches consist of three stages: liveness classifier training, threshold optimization and testing. Therefore, the data set is divided into three partitions: the training set S_{tr} for liveness classifier, the optimization set S_{op} for determining the thresholds and the testing set S_{te} which is reserved uniquely for estimating the generalization performance. We employ the signature samples of the first 100 signers as S_{tr} to train the liveness classifier. Signature samples of another 100 signers are used as S_{op} to optimize the thresholds. Finally, we apply the threshold obtained on S_{op} to the signature samples from the last 100 signers, i.e. the testing set S_{te}, to measure the performance on the unseen testing set. The numbers of spoofing samples and random samples are different in threshold optimization stage but the same in testing stage. The first genuine sample of each signer is regarded as the reference sample.

For comparison, we conduct two other experiments where we take out the liveness classifier. Thus these baseline experiments only have two stages, threshold optimization and testing, without the liveness classifier training.

The experimental settings of the two compared approaches and the three proposed approaches are shown in Table 3. The terms 'Proposed' and 'Conventional' in parentheses are used to distinguish our approaches from the baseline approaches. The content in the second column indicates that a particular approach uses both matching classifier and liveness classifier or only the matching classifier. The third and the fourth columns show the observations and corresponding thresholds for each approach, respectively. Finally, the fifth column shows the datasets used by each approach.

In the *single threshold brute-force (Conventional)* experiment, the only observation used is the matching score d. Thus the system corresponds to a conventional biometric authentication system as we note in parentheses. In the same way, the posterior probability $P(M=1,L=1|d,s)$ in the *single threshold probabilistic (Proposed)* method reduces to $P(M=1|d)$ in the *single threshold probabilistic (Conventional)* method. As indicated in the third row, this method does not use the S_{tr} set because it does not require liveness detection, but still requires the S_{op} set for threshold optimization and for estimating the posterior probability $P(M=1|d)$, as well as S_{te} for performance assessment.

5.2.1. Feature extraction

We use local binary pattern (LBP) to represent the texture of signature in our study. The LBP operator is a powerful gray level invariant texture measurement which was originally designed for texture description [16]. The operator labels every pixel of an image by thresholding its 3×3 neighbor values with it and considering the result as a binary number. Let us represent an image by $I(x, y)$ where x and y represents the x and y coordinates. The LBP operator transforms the input image

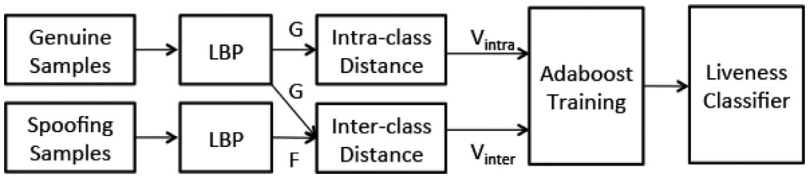

Fig. 6. A flow chart illustrating how a signature liveness classifier with LBP features is trained using Adaboost.

$I(x, y)$ to $I_{LBP}(x, y)$ by:

$$I_{LBP}(x_c, y_c) = \sum_{p=0}^{7} s(I(x_p, y_p) - I(x_c, y_c))2^p, \tag{8}$$

where

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}, \tag{9}$$

and $I(x_c, y_c)$ denotes the the center pixel and $I(x_p, y_p)$ corresponds to the 8-neighborhood around $I(x_c, y_c)$. We then model the distribution of the local pattern by a spatial histogram. In order to maintain the location of different structures inside the image, we divide the image into four equal vertical blocks and three equal horizontal blocks which overlapped by 60% [10]. For each block, we calculate its histogram $\{h^i_{LBP}\}^{12}_{i=1}$. We obtain many 8-bit binary codes after using LBP algorithm, there are 256 kinds of codes in total, representing 256 kinds of gray levels. Since the background of the signature is white and carries no useful information, thus the bin corresponding to the code of the background is not considered in the histogram, causing each block has 255 bins. And the final dimension of the LBP feature of a signature image is $255 \times 12 = 3060$.

5.2.2. Liveness classifier

We trained a liveness classifier in order to obtain the liveness score. The flow chart of training stage is shown in Fig. 6. We calculate the distance vector between genuine features $\boldsymbol{g_i}$ and $\boldsymbol{g_j}$ as their absolute element-wise difference:

$$\boldsymbol{v_{ij}^{intra}} = |\boldsymbol{g_i} - \boldsymbol{g_j}|, \tag{10}$$

thus obtaining an intra-class distance vector as our positive features. Similarly, we calculate distance vector between genuine features $\boldsymbol{g_i}$ and spoofing features $\boldsymbol{f_j}$ as:

$$\boldsymbol{v_{ij}^{inter}} = |\boldsymbol{g_i} - \boldsymbol{f_j}|, \tag{11}$$

which produces an inter-class distance vector as our negative features. AdaBoost is an adaptive boosting algorithm that can improve the performance of a weak learner by iteratively refining the learner in order to find a small number of weak classifiers and then combine them to form a strong one [11,22]. We employ the AdaBoost algorithm to train a strong classifier as our liveness classifier.

Although there may be an issue with the different level of skill of the forgers, we did not look into this in this context of study, but will study this in the future.

5.2.3. Scatter plot

Fig. 7 shows the scatter plot of two scores of the three classes of samples, genuine, spoofing and random. One can discover that:

- Genuine samples have a very low matching score d and a very high liveness score s.
- Since LBPs are used, genuine and spoofing samples differ in their liveness textures, causing spoofing samples have a much lower liveness score s than genuine ones.
- Random samples have a much higher matching score d than genuine samples.
- Spoofing samples have a lower matching score d than random samples.
- Although spoofing samples and random samples both have lower liveness scores than genuine samples, the liveness scores of random samples are not significantly higher than that of spoofing samples. We interpret this appearance as the result of using absolute distance as the liveness feature. Both the features of spoofing samples and random samples are generally characterised by low liveness scores. However, we argue that this characteristic is not detrimental to the performance of our system since there is no need to distinguish spoofing samples from random samples in the final decision. Recall that the prime objective of authentication is only to separate the genuine samples from zero-effort and spoof attacks.

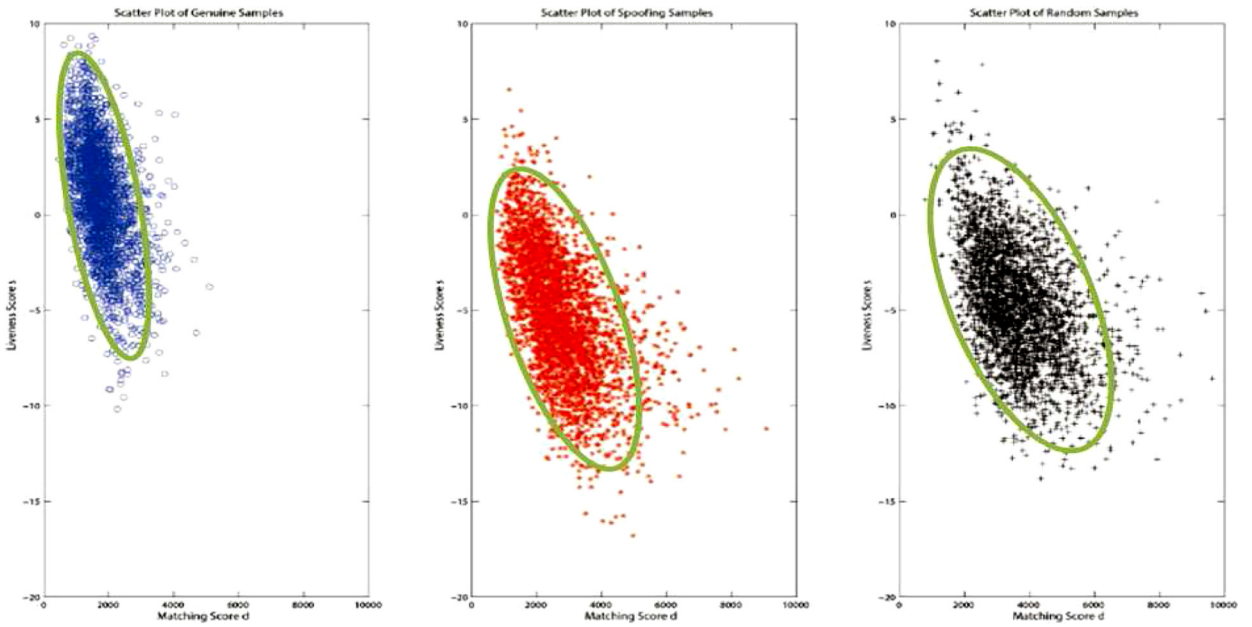

Fig. 7. Scatter plot of genuine samples (left), spoofing samples (middle) and random samples (right) in the threshold optimization stage. The X-axis is the liveness score and the Y-axis is the matching score.

5.2.4. Brute-force, single threshold probabilistic and double threshold probabilistic optimization

Some intermediate results during their threshold optimization stage for the three proposed methods, namely, brute-force optimization, single-threshold probabilistic optimization and double-threshold probabilistic optimization are shown in Figs. 8, 9 and 10, respectively. Since the coordinates of Figs. 8 and 10 are different, these two figures appear to be similar but not the same.

In all these diagrams, one can recognize that HTER increases at either end of the matching score space of d; and that there is an extreme point where HTER attains its minimum value, as we marked in the figures. For instance, for the matching score d in the brute-force optimization, if T_d is too high, too many negative samples will be classified into positive, which leads to a very high FAR. Conversely, if T_d is too low, the FRR will be very high. Thus it's reasonable the HTER obtains its minimum value in the middle area.

5.2.5. Prior probabilities and probability density functions

In the threshold optimization stage in the experiments of three proposed approaches, we have 2300 genuine samples, 2977 spoofing samples and 3000 random samples in S_{op}. Thus,

$$P(M=1)=\frac{2300+2977}{2300+2977+3000}=0.6375$$

and

$$P(L=1)=\frac{2300+3000}{2300+2977+3000}=0.6403.$$

The probability density functions (PDFs) we fitted are shown in Figs. 11 and 12. One can see that the probability density functions are different when $M=1$ and $M=0$ in Fig. 11. The peak of PDF of $M=1$ locates at about 2,000, while that of $M=0$ locates at about 3,000. Similarly, the peak of PDF of $L=1$ locates at about -3, while that of $L=0$ locates at about -5, and the shape of PDFs differ.

5.3. Results

The results of our proposed methods and baseline methods are shown in Table 4. We also augment the results using logistic regression to train liveness classifier. The term 'Total' means the presence of both spoofing samples and random samples in testing set, namely the whole S_{te}. *Spoofing* means testing samples only consist of genuine samples and spoofing samples. Similarly, *Random* means there are only genuine samples and random samples in testing set. FAR, FRR and HTER are given in the form of percentage.

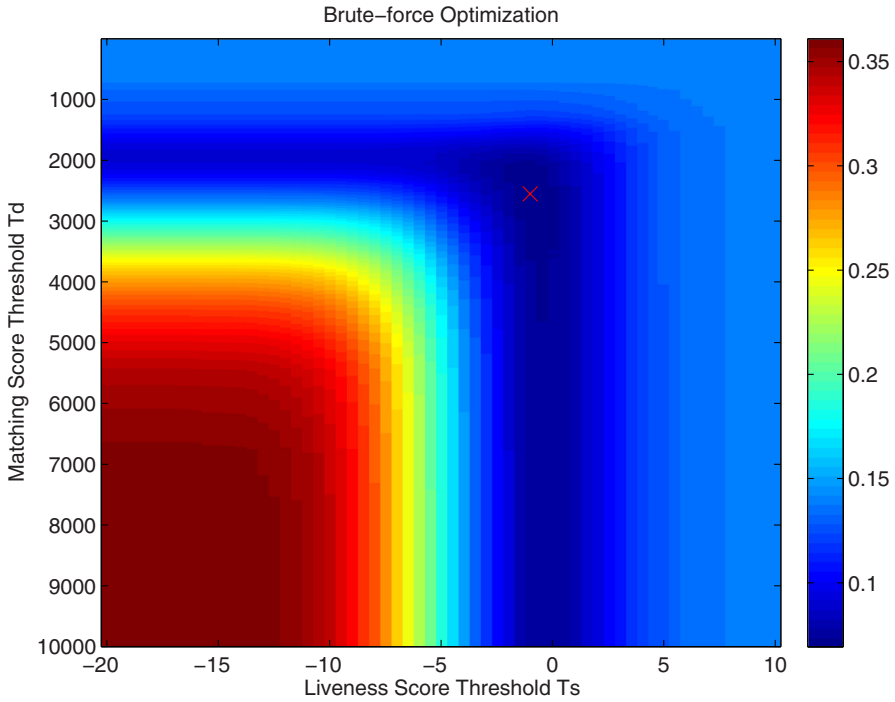

Fig. 8. Brute-force optimization in the threshold optimization stage. The vertical bar shows the value of HTER in the (d, s) space. The red cross shows the location of the minimum HTER in the optimization set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

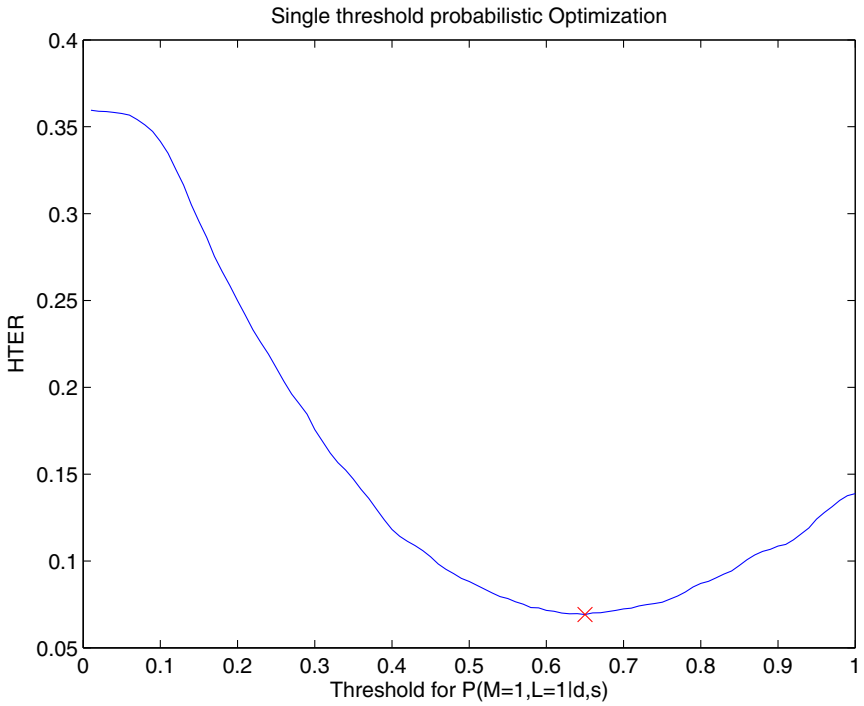

Fig. 9. Single threshold probabilistic optimization in the threshold optimization stage.

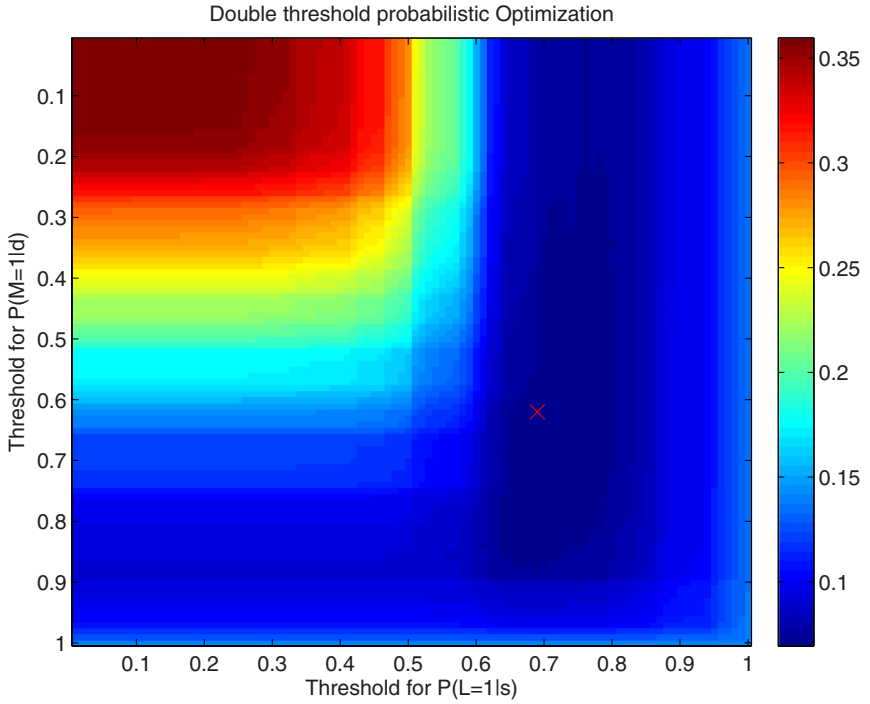

Fig. 10. Double threshold probabilistic optimization in the threshold optimization stage. The vertical bar shows the value of HTER in the (d, s) space. The red cross shows the location of the minimum HTER in the optimization set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

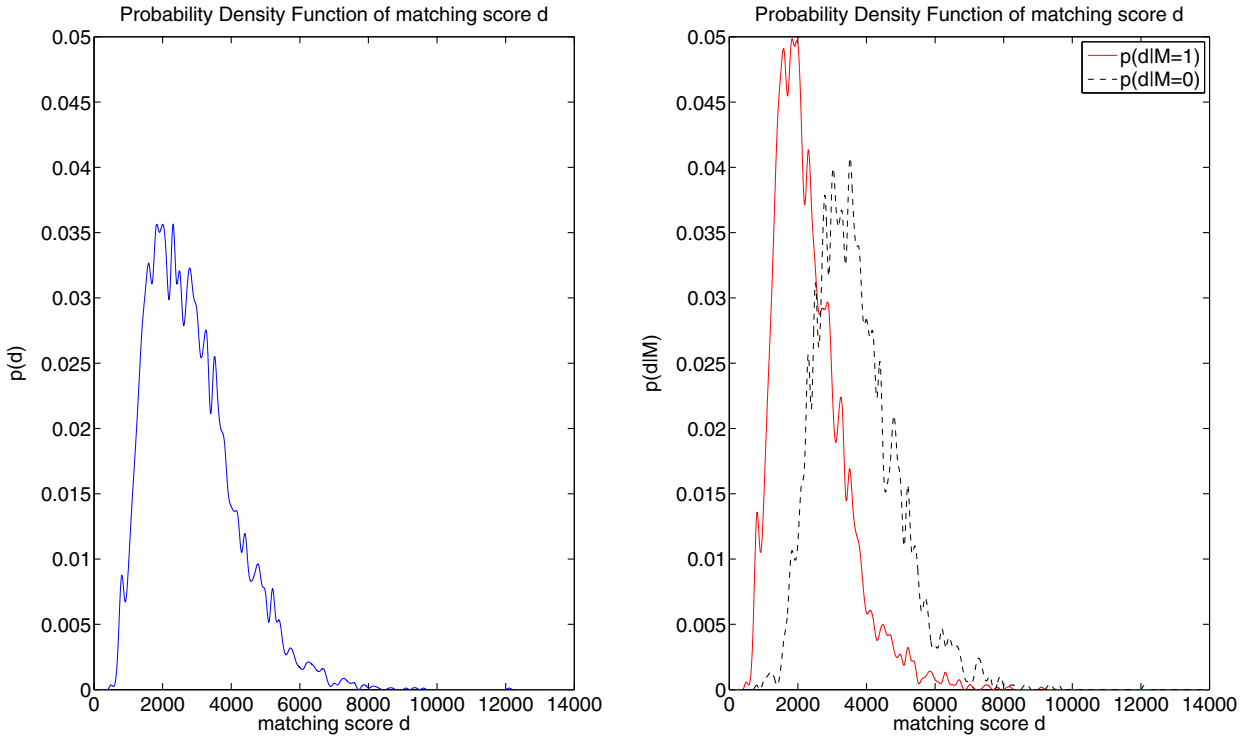

Fig. 11. The estimated probability density functions of $P(d)$, $P(d|M = 1)$ and $P(d|M = 0)$.

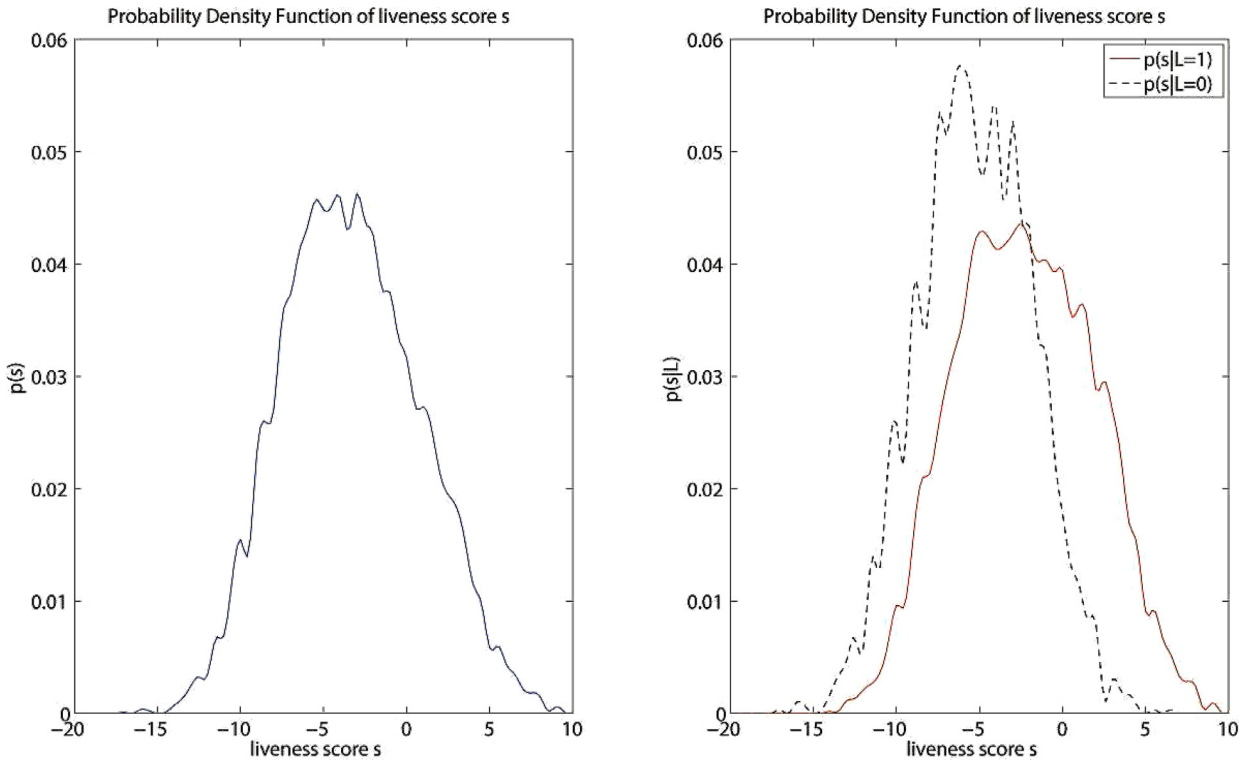

Fig. 12. The estimated probability density functions of $P(s)$, $P(s|L=1)$ and $P(s|L=0)$.

Table 4
Performance of proposed approaches and conventional approaches.

Approach	Samples	FAR(%)	FRR(%)	HTER(%)
Single threshold brute-force (Conventional)	Total	9.10	9.28	9.19
	Spoofing	12.68	14.53	13.60
	Random	1.57	14.53	8.05
Double threshold brute-force (Proposed)	Total	7.12	8.13	7.63
	Spoofing	7.89	12.74	10.31
	Random	3.26	12.74	8.00
Double threshold brute-force (Logistic)	Total	8.45	8.17	8.31
	Spoofing	8.87	12.79	10.83
	Random	4.36	12.79	8.58
Single threshold probabilistic (Conventional)	Total	8.82	9.31	9.07
	Spoofing	12.34	14.58	13.46
	Random	1.47	14.58	8.03
Single threshold probabilistic (Proposed)	Total	**5.71**	8.81	**7.26**
	Spoofing	**7.06**	13.79	**10.42**
	Random	**1.89**	13.79	**7.84**
Single threshold probabilistic (Logistic)	Total	6.27	10.02	8.14
	Spoofing	6.43	15.70	11.07
	Random	3.38	15.70	9.54
Double threshold probabilistic (Proposed)	Total	7.58	**7.92**	7.75
	Spoofing	8.11	**12.40**	10.25
	Random	3.75	**12.40**	8.08

5.3.1. Conventional versus Proposed

The 'FAR', 'FRR' and 'HTER' all show that our three proposed approaches outperform the two conventional approaches on both 'Total' and 'Spoofing' samples. However, when testing set consists of only genuine samples and random samples, the proposed approaches obtain a higher FAR, as the results of 'Random' show. This finding indicates that the introduction of liveness score improves the performance of biometric authentication system under spoofing attacks significantly, but loses some performance on the random samples. This is understandable because some genuine samples will be rejected due to their low liveness scores which will lead to a higher FAR. Yet, the FRR decreases. On the whole, the introduction of

liveness score and the schemes of threshold optimization improve the performance of biometric authentication system in the presence of both spoofing samples and random samples.

5.3.2. Brute-force versus probabilistic

Since the three proposed approaches capitalize on the same logic construct, their performances are close, especially the brute-force one and the double threshold probabilistic one. These two approaches achieve very similar performance. We can just comprehend the two observations (i.e. $P(M = 1|d)$ and $P(L = 1|s)$) in the double threshold probabilistic approach as another form of d and s in the brute-force approach. Their subtle but important difference, however, is that, changing the prior is possible with the probabilistic approach but for the brute-force approach, another round of optimization is required with the correct proportion of training data that reflects the desired attack priors.

We also note that the single threshold probabilistic approach performs better than the other two proposed approaches on 'Random' and 'Total', but poorer on 'Spoofing'. Since this approach fuses $P(M = 1|d)$ and $P(L = 1|s)$ into a new measurement $P(M = 1, L = 1|d, s)$, its performance is similar to or better than the conventional approach on 'Random' samples, but not as good as other proposed approaches on 'Spoofing' samples. Overall, the single threshold probabilistic approach performs the best in this case. We have printed the best results among different approaches in bold according to the performance on the Total.

The disadvantage of the single threshold probabilistic approach is that it needs to estimate the four PDFs which depend on the optimization set S_{op} when comparing to brute-force scheme. We need to clarify that the prior probabilities $P(M = 1)$ and $P(L = 1)$ has no significant impact on the performance. Although the priors of the data have been used, these priors can also be set to, e.g. $P(M = 1) = P(L = 1) = 0.5$. The key point is that these prior probabilities have to be consistent in both optimization stage and testing stage. Otherwise the threshold so-obtained will not be optimal.

5.3.3. Logistic regression versus Adaboost

For the comparison, we have also used logistic regression to train the liveness classifier in place of Adaboost. We use Adaboost to obtain the value of s, which is optimized for a threshold to determine the test performance together with the score of d. But using logistic regression, we compute a probabilistic value to decide a query sample is a live one or a fake one. In addition, the threshold for logistic regression is 0.5 in common, so the optimization for the probability s is needless.

To compare the performance of logistic regression with our proposed optimization schemes, we search the threshold in [0,1] instead of using 0.5 as the threshold of s during the optimization stage. In addition, we use Brute-force and single threshold probabilistic approaches to optimize the threshold of s and d, as well as for their joint probability. In addition, we have also explored different parameters when optimizing the weights of logistic regression, e.g., changing the number of iterations and the learning rate. Despite these attempts, we found that the performance of logistic regression is still inferior compared to Adaboost, as indicated by higher FAR, FRR and HTER of the former. Based on this observation, we believe that Adaboost is more suitable for our optimization schemes.

6. Conclusions

In order to render a biometric system robust against presentation attacks, it is often necessary to consider the liveness of a biometric sample, or more generally, the probability that a sample is forged or the system is subject to a spoof attack. This can be viewed as a fusion problem involving two measurements, namely, the verification (matching) score and the liveness score.

Although complicated machine-learning based methods have been proposed in the literature, such as Bayesian network and logistic regression, in this paper, we have opted to investigate simpler threshold-based schemes because these methods are extremely easy to implement. However, the question being posed here is not about implementation; but about recommendations and the best practices with regards to the optimization strategies required in order to attain the best possible performance.

The thresholds-based scheme can be divided into the following dichotomies: brute-force versus probabilistic strategy, and single versus double thresholds. While the brute-force strategy aims at optimizing the decision thresholds directly, the probabilistic strategy converts the observations (matching score or liveness score, or both) into a probabilistic space that relates to the events of interest (same versus different person, liveness state, or both aspects) more explicitly.

We have explored all possible variants of scheme systematically, tested them using an offline signature database, and built our own liveness/spoof detection classifier, and a baseline signature verification classifier.

Among the different variants, we consider three to be novel, namely, the double-threshold brute-force method, the single-threshold probabilistic method, and the double-threshold probabilistic method. These methods turn out to perform better than their baseline counterparts since they can lower the errors in the presence of spoof attacks, thus demonstrating the validity of our proposal.

7. Future work

Although the experimental results show that the proposed approaches perform better in the presence of both spoofing samples and random samples, these methods may also cause a slight increase in FAR under the conventional zero-effort

attack. This calls for a future research direction in better addressing this shortcoming. Apart from this, the current investigation can be extended in a number of ways, including, but not limited to the following directions:

- Measuring and understanding the correlation between d and s.
- Considering multiple liveness measures.
- Studying the effect of the skill of the forgers in offline signature authentication case.
- Applying the proposed technique to other biometric modalities.
- Measuring the performance of the proposed framework on several data sets.

References

[1] S.M.S. Ahmad, A. Shakil, M.A. Faudzi, R.M. Anwar, M.A.M. Balbed, A hybrid statistical modelling, normalization and inferencing techniques of an off-line signature verification system, in: Proceedings of the Computer Science and Information Engineering, WRI World Congress on, vol. 6, IEEE, 2009, pp. 6–11.
[2] Z. Akhtar, G. Fumera, G.L. Marcialis, F. Roli, Evaluation of multimodal biometric score fusion rules under spoof attacks, in: Proceedings of the 5th IAPR International Conference on Biometrics (ICB), 2012, pp. 402–407.
[3] A. Anjos, I. Chingovska, S. Marcel, Anti-spoofing in action: joint operation with a verification system, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Biometrics, 2013, pp. 98–104.
[4] B. Biggio, Z. Akhtar, G. Fumera, G.L. Marcialis, F. Roli, Security evaluation of biometric authentication systems under real spoofing attacks, IET Biometrics 1 (1) (2012) 11–24.
[5] K.I. Chang, K.W. Bowyer, P.J. Flynn, An evaluation of multimodal 2D+3D face biometrics, IEEE Trans. Pattern Anal. Mach. Intell. 27 (4) (2005) 619–624.
[6] A. Chugh, C. Jain, P. Singh, P. Rana, Learning approach for offline signature verification using vector quantization technique, in: Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India (CSI), Vol. 1, Springer, 2015, pp. 337–344.
[7] M. Drahansky, D. Lodrová, Liveness detection for biometric systems based on papillary lines, Int. J. Secur. Appl. 2 (4) (2008) 29–37.
[8] N. Erdogmus, S. Marcel, Spoofing face recognition with 3d masks, IEEE Trans. Inf. Forens. Secur. 9 (7) (2014) 1084–1097.
[9] B. Fang, Y. Wang, C.H. Leung, Y.Y. Tang, P.C. Kwok, K. Tse, Y. Wong, A smoothness index based approach for off-line signature verification, in: Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR'99., IEEE, 1999, pp. 785–787.
[10] M.A. Ferrer, J. Vargas, A. Morales, A. Ordóñez, Robustness of offline signature verification based on gray level features, IEEE Trans. Inf. Forens. Secur. 7 (3) (2012) 966–977.
[11] Y. Freund, R.E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, Comput. Learn. Theory (1995) 23–37.
[12] Y. Guerbai, Y. Chibani, B. Hadjadji, The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters, Pattern Recognit. 48 (1) (2015) 103–113.
[13] J. Maatta, A. Hadid, M. Pietikainen, Face spoofing detection from single images using texture and local shape analysis, IET Biometrics 1 (1) (2012) 3–10.
[14] E. Marasco, Y. Ding, A. Ross, Combining match scores with liveness values in a fingerprint verification system, in: Proceedings of the the Fifth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2012, pp. 418–425.
[15] T. Matsumoto, H. Matsumoto, K. Yamada, S. Hoshino, Impact of artificial "gummy" fingers on fingerprint systems, in: Proceedings of the SPIE, Optical Security and Counterfeit Deterrence Techniques IV, vol. 4677, 2002, pp. 275–289.
[16] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognit. 29 (1) (1996) 51–59.
[17] P. Perrot, G. Aversano, R. Blouet, M. Charbit, G. Chollet, Voice forgery using ALISP: indexation in a client memory, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, 2005, pp. 17–20.
[18] N. Poh, S. Bengio, Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication, Pattern Recognit. 39 (2) (2006) 223–233.
[19] A. Rattani, N. Poh, Biometric system design under zero and non-zero effort attacks, in: Proceedings of the IEEE International Conference on Biometrics (ICB), 2013, pp. 1–8.
[20] A. Ross, A. Jain, Information fusion in biometrics, Pattern Recognit. Lett. 24 (13) (2003) 2115–2125.
[21] H. Saikia, K.C. Sarma, Approaches and issues in offline signature verification system, Int. J. Comput. Appl. 42 (16) (2012) 45–52.
[22] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Mach. Learn. 37 (3) (1999) 297–336.
[23] J.F. Vargas, M.A. Ferrer, C. Travieso, J.B. Alonso, Off-line signature verification based on grey level information using texture features, Pattern Recognit. 44 (2) (2011) 375–385.
[24] D. Yambay, L. Ghiani, P. Denti, G.L. Marcialis, F. Roli, S. Schuckers, Livdet 2011 - fingerprint liveness detection competition 2011, in: Proceedings of the 5th IAPR International Conference on Biometrics (ICB), 2012, pp. 208–215.
[25] M.B. Yılmaz, B. Yanıkoğlu, Score level fusion of classifiers in off-line signature verification, Inf. Fusion (2016), doi:10.1016/j.inffus.2016.02.003.