ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed
Computing

Signature Verification on Finger Operated Touchscreen Devices

Semester Thesis

Pascal Bissig
bissigp@ee.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Tobias Langner, Samuel Welten
Prof. Dr. Roger Wattenhofer

October 20, 2011

Abstract

The increasing number of smartphone users and the involved increased amount of privacy critical information that these users carry around call for improved methods to protect this data. Biometric authentication methods have one important advantage when compared to a password. A biometric trait contains more information, which is harder to copy, steal or reproduce. Modern smartphones are generally touchscreen operated and therefore suitable to acquire hand-written signatures from a user. The goal of this thesis is to implement a signature verification system that is suitable for the finger operated touchscreen devices of today. In the process different signature verification approaches are discussed and finally two separate verification systems are implemented. These are a Dynamic Time Warping based system and a global feature based system which uses a Support Vector Machine for classification. These two systems are finally combined to further increase classification performance.

Keywords: Online Signature Verification, Dynamic Time Warping, One-class Support Vector Machine

CR Categories: Verification, Experimentation

Contents

Chapter 1

Introduction

1.1 The Need for Mobile Authentication

Smartphones and notebooks are becoming increasingly popular. These devices hold privacy critical information such as contact lists, email account access, webbrowser passwords, communication history with contacts and so on. Additionally these devices are mobile and hence exposed to a higher risk of being accessed by unauthorized users. Therefore it seems necessary to explore new ways to protect this data using existing hardware. In the notebook industry, fingerprint readers and file encryption are becoming increasingly popular. Modern mobile phones provide lockscreens, which require a user to enter a code or draw a simple pattern on the touchscreen. These authorization mechanisms help to protect critical data more or less effectively. Fingerprint readers can be considered to be fairly secure, whereas current lockscreens are rather easily bypassed. A code or pattern entered on a touchscreen device can be observed either directly or by looking at the smears left by the finger on the screen. The simplicity of the entered token allows an adversary to easily observe and reproduce it. Compared to the fingerprint reader, a lockscreen code or pattern contains very little information which can easily be stolen. Because acquiring a fingerprint requires special hardware, this thesis is focused on exploiting a users signature as the basic authentication token.

1.2 Biometrics

1.2.1 The Goal of Biometrics

The goal of biometrics is to validate the identity of a user by analyzing physical or behavioural characteristics of said user. A good introduction to the known biometrical traits was given by Jain et al. [1]. A password can be used for the same purpose, but there are great differences between these two options. Usually a password is relatively short, meaning it does not contain a lot of information.

Even a password with twenty characters contains by far less information than an iris or fingerprint scan. On one hand this leads to the previously discussed advantage that more information tends to be harder to steal. On the other hand a password has the advantage of being either valid or invalid, whereas a biometric trait will never be perfectly valid or invalid but always somewhere in between.

1.2.2 Performance of a Biometric System

Because a biometric system most likely will not be able to perfectly distinguish valid traits from forged ones, a measure for the verification performance is necessary. Generally the task of a biometric system can be described as a classification task with two possible results, namely valid or invalid. Classification performance is characterized by three basic measures:

- False accept rate: The percentage of invalid test samples that are classified as valid.
- False reject rate: The percentage of valid test samples that are classified as invalid.
- Equal error rate: The error rate (false accept and false reject) which is reached if the classifier is configured such that it produces a false accept rate which is equal to the false reject rate.

To visualize the performance of a classification, Detection Error Trade-off (DET) curves are used. Depending on the intended use of the biometric system, it might be interesting to minimize either false accept-, false reject- or equal error rate.

1.3 Online Signature Verification

The task of verifying a user identity based on the user's signature is called signature verification. An offline signature verification method can classify a static image of a signature whereas an online signature verification method also considers the dynamics of the signing process. Signatures can be recorded using different methods and hardware. Several online signature datasets are available. For example the MCYT baseline corpus [2] or the dataset from SVC2004 [3]. Most commonly, digitizing tablets in combination with a pen are used to capture online signature data. Online signature verification methods are generally categorized to either use global or local features to classify a given signature.

1.3.1 Input Methods

Digitizing the dynamics of the signing process can be achieved using different recording devices such as digitizing tablets, or touchscreen devices. Several dynamic signature databases are available for researchers to test and compare their verification algorithms. These datasets are captured using some sort of pens (be it on a resisitive touchscreen or an inductive digitizing tablet) [4]. This method of signature aquisition is not applicable in todays environment of smartphones with capacitive touchscreens which are operated using a finger instead of a pen.

In this work, a small database of signature is captured. Each signature is entered using a finger instead of a pen. The problem of verifying signatures captured using pens on either digitizing tablets or resistive touchscreens has been studied before. However, no literature describing finger input signature verification was found during this thesis. Using the finger instead of a pen to input a signature on a touchscreen is a challenge since moving the finger in different directions on the touchscreen results in different friction characteristics. Also, the screen size of our recording device (3.7 inches) is very limited compared to a digitizing tablet.

1.3.2 Feature-based Methods

As the name suggests, feature-based methods use features extracted from the signatures to perform the verification task. In general two different types of features are considered: global and local. Global features are related to the signature as a whole, for instance the signature duration or the mean pressure applied. Local features are based on single sample points. Examples for local features are the maximum velocity or the highest curvature. A signature verification system using only such features was proposed by Lee et al. [5]. A large number of features useful for signature verification was listed by Fierrez-Aguilar et al. [6].

The dimensionality of the feature vector can get arbitrarily big whereas the training set usually is limited to a small number of signatures. Estimating too many parameters from a small set of training samples deteriorates verification performance as shown by Fierrez-Aguilar et al. [6]. This is why in general a smaller set of more discriminative features is preferred. The issue of feature selection in the signature verification task was extensively studied by Richiardi et al. [7].

After having found a suitable feature vector, an arbitrary one-class classification scheme can be used to perform the classification into valid and invalid signatures. Since there is no negative training data available, the task of classification is very similar to the task of outlier- or novelty detection. Several methods to perform such a classification are known [8].

1.3.3 Function-based Methods

Function-based signature verification methods are focused on building a signature model which accurately reflects the temporal behaviour of the signature. Instead of extracting features that are used for verification, the emphasis of the signature model lies in the (timed) sequence of strokes drawn during the signing process. Comparing any signature to the model built during training will lead to a match score which is based on the level of similarity between (timed) sequence of strokes in the model and the signature under test. Function-based methods are reported to deliver better performance than global feature based methods [6]. Two function-based methods are predominant in the literature: Hidden Markov Model based methods and Dynamic Time Warping (DTW) based methods.

Chapter 2

Signature Verification System

2.1 Signature Acquisition

We used a Google Android smartphone (HTC Desire) with a capacitive 3.7 inch touchscreen to record signatures. All the signatures were entered with a finger. Technical properties of the touchscreen include a sampling rate which is not constant but has a mean of 87 Hz and a standard deviation of 7 Hz as shown in figure 2.1. In some very rare cases the sampling rate reaches up to 500 Hz. From the device the following information is available:

- $x(t)$: X-coordinate of touch location.
- $y(t)$: Y-coordinate of touch location.
- $p(t)$: Pressure value.
- $a(t)$: Touch area estimation of the touching object

In order to reflect different usage scenarios, signatures were gathered from users without restricting their body posture, some were standing and holding the device, others were sitting. Some people signed by holding the device in one hand, and signing with the other while others signed by putting the device on a surface and holding it down to that surface with one hand while writing with the other. Each user entered about 20 signatures. In addition to these genuine signatures, users were asked to try to forge signatures of other users. Different levels of knowledge about the signature to be forged were considered:

1. random forgeries: the signatures from other users
2. forgeries based on the knowledge of the sequence of letters
3. forgeries based on the knowledge of the image of the signature

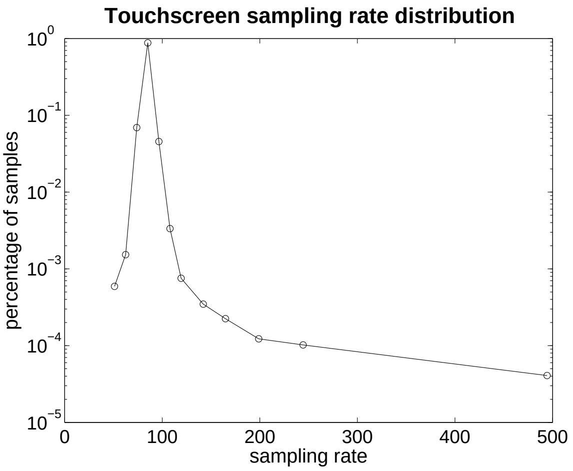


Figure 2.1: Sampling rate distribution on HTC Desire

4. forgeries based on the observation of the signing process

Forgery types 2-4 were only recorded for a subset of the recorded signatures. Instead of explicitly acquiring random forgeries, we used signatures from different users as random forgeries.

2.2 Feature-based System

The feature-based system extracts global and local features from the signature samples. Compared to the work of Fierrez-Aguilar et al. [6] our feature vector also contains features extracted from the pressure signal. See table 2.1 for the complete list of features used in this work during the experiments.

Table 2.1: List of signature features

No.	Rank	Feature	s	Description
0	9	T		Total signature duration
1	2	#Strokes		Number of finger strokes
2	6	T_{touch}		Amount of time the finger touched the display
3	18	$\frac{T_{\text{touch}}}{T}$		Percentage of time with finger on display
4	36	$\frac{\text{mean}(\lvert v\rvert)}{\max(v_x)}$		Average velocity / maximal velocity in x direction

No.	Rank	Feature	s	Description
5	10	#sign changes	$\dot{x}$	The number of sign changes
6	19		$\ddot{x}$	
7	13		$\dot{y}$	
8	16		$\ddot{y}$	
9	14		$\dot{p}$	
10	15		$\ddot{p}$	
11	37	$\text{mean}(s) = \frac{1}{T}\int_0^T s$	$\|j\|$	$\|j\| = \sqrt{\dddot{x}^2 + \dddot{y}^2}$
12	1		$\dot{x}$	Mean of given signals
13	41		$\ddot{x}$	
14	11		$\dot{y}$	
15	34		$\ddot{y}$	
16	8		p	
17	52		$\dot{p}$	
18	50		$\ddot{p}$	
19	35	$\sqrt{\text{mean}(s - \text{mean}(s))^2}$	x	Standard deviation
20	20		$\dot{x}$	
21	24		$\ddot{x}$	
22	22		y	
23	23		$\dot{y}$	
24	26		$\ddot{y}$	
25	12		p	
26	17		$\dot{p}$	
27	25		$\ddot{p}$	
28	27	$\max(s)$	$\dot{x}$	Maximum value
29	45		$\ddot{x}$	
30	43		$\dot{y}$	
31	42		$\ddot{y}$	
32	4		p	
33	40		$\dot{p}$	
34	38		$\ddot{p}$	
35	31	$\text{rms}(s) = \sqrt{\frac{1}{T}\int_0^T s}$	$\dot{x}$	Root mean square value
36	33		$\ddot{x}$	
37	28		$\dot{y}$	
38	30		$\ddot{y}$	
39	3		p	
40	21		$\dot{p}$	
41	56		$\ddot{p}$	
42	39	$\text{meanmax}(s) = \frac{\text{mean}(s)}{\max(s)}$	p	
43	54		$\frac{dp}{dt}$	
44	48		$\frac{d^2p}{dt^2}$	

No.	Rank	Feature	s	Description
45	7	$t_{\max}(s) = \frac{1}{T}(t \vert s(t) = \max(s(t)))$	$\dot{x}$	Normalized point in time when the maximal value of signal s is reached
46	51		$\ddot{x}$	
47	44		$\dot{y}$	
48	32		$\ddot{y}$	
49	47		p	
50	53		$\dot{p}$	
51	55		$\ddot{p}$	
52	29	$t_{\min}(s) = \frac{1}{T}(t \vert s(t) = \min(s(t)))$	$\dot{x}$	Normalized point in time when the minimal value of signal s is reached
53	46		$\ddot{x}$	
54	5		$\dot{y}$	
55	49		$\ddot{y}$	

The features in this table were selected according to the following criteria:

1. scale invariance
2. rotation invariance
3. high discriminative ration according to the work of Fierrez-Aguilar et al. [6]

The list of features is composed of top ranked features from Fierrez-Aguilar et al. [6] as well as applying proposed functions (rms(s),std(s) etc.) to the pressure signal. Because the recorded signals are not preprocessed, special attention was paid to the scale and rotation invariance of the added features. There are two reasons for this: Firstly, a rotated and/or scaled version of the signature will be similarly hard to produce for an intruder, the false accept rate is not expected to be influenced dramatically by rotation/scale invariance. Secondly, if genuine users enter their signatures with another rotation or scale compared to the training set, this should not affect the false reject rate.

2.2.1 Feature Selection

The features where ranked using the fisher discriminant ratio as previously described in [6] and [9]:

$$\mathrm{FDR}_i(u) = \frac{(\mu_{i,G} - \mu_{i,F})^2}{\sigma_{i,G}^2 + \sigma_{i,F}^2} \tag{2.1}$$

In this formula u signifies the user, i is the index of the feature, G represents the set of genuine user signatures and F represents the set of forgeries. μ and σ indicate the mean and standard deviation respectively. From the formula we can see that the FDR of a feature generally will be higher if the difference in the mean scores is bigger. Also the FDR will be bigger if the two standard deviations of the genuine-and forgery score distribution are lower. The higher the mean FDR of a given feature i is, the higher it is ranked in table table 2.1. No feature selection algorithm was utilized, but instead a set of feature vectors was chosen to investigate their respective performance: (see table 2.1 for the features corresponding to the indices). Note that 36% of all features are derived from the pressure signal. In the 10 top ranked features we find 30% pressure features, in the top 20 35% and in the top 30 30%. The relative importance of the pressure related features therefore can be expected to be roughly the same as for the x and y features. Pressure features are indicated by bold font.

1. all features: [0..55]
2. pressure signal related features: **[9, 10, 16, 17, 18, 25, 26, 27, 32, 33, 34, 39, 40, 41, 42, 43, 44, 49, 50, 51]**
3. non pressure signal related features: [0, 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 14, 15, 16, 19, 20, 21, 22, 23, 24, 28, 29, 30, 31, 35, 36, 37, 38, 45, 46, 47, 48, 52, 53, 54, 55]
4. 30 highest ranked features: [12, 1, **39**, **32**, 54, 2, 45, **16**, 0, 5, 14, **25**, 7, **9**, **10**, 8, **26**, 3, 6, 20, **40**, 22, 23, 21, **27**, 24, 28, 37, 52, 38]
5. 20 highest ranked features: [12, 1, **39**, **32**, 54, 2, 45, **16**, 0, 5, 14, **25**, 7, **9**, **10**, 8, **26**, 3, 6, 20]
6. 10 highest ranked features: [12, 1, **39**, **32**, 54, 2, 45, **16**, 0, 5]

2.2.2 Feature Score Normalization

Since the different features presented before lead to feature values in very different co-domains, it is neccesary to apply a normalization scheme if we want to relate them. This normalization will assure that different features are not weighted differently because of their co-domain. As proposed by Hsu et al. [10] for Support Vector Machines, we use the min-max normalization scheme to get feature values in the range $[-1, 1]$.

$$s'_i = 2\left(\frac{s_i - \min(T_{s_i})}{\max(T_{s_i}) - \min(T_{s_i})}\right) - 1 \tag{2.2}$$

Here, s'_i is the normalized feature value, s_i is the non normalized feature score and T_{s_i} are the scores for feature i in the training set. Values outside $[-1, 1]$ therefore are possible after normalization, because only values inside $[min(T_{s_i}), max(T_{s_i})]$ are mapped to $[-1, 1]$. However the probability of a value lying outside $[-1, 1]$ is small and decreases as the distance to the interval grows. The normalization should therefore serve the purpose of normalizing the effect of each feature value in the detection process.

2.2.3 Single Class Classification – Anomaly Detection

In our case using negative training samples is not practical because finding a signature set representative for all possible signatures is extremely hard if not impossible. The performance would strongly depend on the chosen set of negative training samples. Therefore we want to use a classification scheme that does not rely on negative training samples. Several different methods suitable for this task were proposed in the literature, Chandola et al. [8] summarized many different approaches. We decided to use the following two methods for anomaly detection.

1. One-class Support Vector Machines: SVMs and their application to one class problems are described in Schölkopf et al. [11, 12].

2. Mahalanobis distance: For each feature of the training set, μ and σ are estimated.The Mahalanobis distance of a test signature can then be calculated via

$$d = \sum_{n=0}^{\#\text{Features}} \left(\frac{s_n - \mu_n}{\sigma_n} \right)^2 \tag{2.3}$$

   Classification is performed by applying a threshold to the distance value d.

The Mahalanobis distance was chosen because of its implementation simplicity. First experiments using the Mahalanobis distance were used to get a working system in a short period of time. The one-class SVM was chosen because of the good results reported from SVMs in many classification tasks. In addition to these two classifiers, the Weka OneClassClassifier [13] was also considered (and implemented) but due to time constraints, it could not be added to the fusion classifier.

2.3 Function-based System: Dynamic Time Warping

Function-based methods to perform signature verification compare the temporal behaviour of the recorded signals $x(t)$, $y(t)$, $p(t)$ and $a(t)$. In this work, a dynamic time warping based method is used. The performance of a Hidden Markov Model system might be higher than the DTW system performance but the implementation complexity is much lower for the DTW system. Once again, time was the crucial element in this decision. Dynamic time warping is a classic dynamic programming algorithm that is also used in speech recognition.

A signature verification system using DTW is reported by Martens and Claesen [14]. DTW is a method to measure the similarity between two sequences which may vary in time or speed. This similarity measure can be used to perform signature verification. Comparing two sequences can be done in many different ways such as correlation, integration of the difference of two signals etc. However all these similarity measures cannot cope with non-linear time distortions which we will have in the signature signals. If a user takes a little longer for the first letter of the signature, the similarity measure should not deteriorate more than if the user was taking a little longer on the last letter. The DTW method allows us to compare two signatures while a small time distortion in the beginning of the signature does not accumulatively deteriorate the similarity measure throughout the whole signature.

In our DTW system no smoothing is carried out, also the signals are not resampled. The preprocessing consists of center of mass normalzation and normalization of x- and y coordinates according to their standard deviation.The algorithm to obtain an optimal warp path between two given signatures is described in algorithm 1.

Algorithm 1 Dynamic Time Warping

```
$n$ = Length of Signature A
$m$ = Length of Signature B
dtw[$n$][$m$] = DTW
for $i = 1 \to m$ do
  dtw[0][$i$] = $\infty$
end for
for $i = 1 \to n$ do
  dtw[$i$][0] = $\infty$
end for
dtw[0][0] = 0;
cost = 0;
for $i = 1 \to n$ do
  for $j = 1 \to m$ do
    cost = dist(Signature A, Signature B, $i$, $j$)
    dtw[$i$][$j$] = cost + min(dtw[$i-1$][$j$], dtw[$i$][$j-1$], dtw[$i-1$][$j-1$])
  end for
end for
```

The distance function dist() used in algorithm 1 computes a distance measure between sample i in signature A and sample j in signature B. The distance measure used in this work is:

$$\sqrt{\sum_{n=0}^{\#\text{Features}} (A_n[i] - B_n[j])^2} \tag{2.4}$$

A_n stands for the n'th feature of signature A, B_n analogously indicates the n'th feature of signature B and i and j are the specific samples which are compared.

Finding the best warping path through dtw[][] is done by iteratively backtracking from dtw[n][m] to the neighboring field with minimal cost value .The value of dtw[n][m] gives the accumulated distances of the two signatures while not accounting any cost to warping time (i.e. there is no cost associated with high time shifts between the signatures A and B). As previously stated, DTW allows us to compare two sequences by allowing time distortions between the two sequences without lowering the similarity measure. This would mean, that a signature which is similar in shape, but very different in writing speed would still achieve a high similarity score. This certainly is not desirable. A difference in writing speed should be penalized, but the penalty should be limited to the section where the time behaviour actually is different. This is achieved by using the derivatives of the signals x, y and p in the feature vector of the DTW.

2.3.1 Feature Selection

Similar to the feature-based method discussed previously, different feature vectors are used to compare their performance.

1. All features: $[p, x, y, \dot{p}, \dot{x}, \dot{y}]$
2. Pressure features: $[p, \dot{p}]$
3. X-coordinate features: $[x, \dot{x}]$
4. Y-coordinate features: $[y, \dot{y}]$

2.3.2 DTW Signature Verification

Training the DTW signature verification algorithm is performed by calculating the distance measure ($dtw[n][m]$) for all pairs of signatures in the training set. The signature with the smallest mean distance measure to all the other signatures is selected as the prototypical signature of the given user.

The DTW matching score of a signature to be verified is equal to the distance measure dtw[n][m] between the prototypical signature found during training and the signature under test.

2.4 System Fusion and Decision Function

The previously presented signature verification methods yield non-normalized scores which can be used independently for verification. However, in order to maximize verification performance, the combination of both should be considered. Jain et al. [15] discussed Score fusion in biometric systems.

2.4.1 Positive Class Score Estimation

Because of the high inter-user variability of the means and standard deviations in the score outputs, it is impossible to define global decision thresholds or probability mappings. Therefore we use the estimated score distribution of the training set to find good decision functions and probability estimators for each individual user.

The score distribution of the positive class is estimated by performing cross-validation on the training set. In each cross-validation step, exactly one score is estimated, using all the other training samples for training. This leads to a sparse score distribution estimate which will later be referred to as the training set score distribution S_T.

2.4.2 Decision Function

From a score, the decision about the class label is performed applying a threshold. The threshold is calculated using the training set score distribution and a global parameter T. This parameter is specific for each classifier and has to be estimated from a dataset which is as large as possible. The parameter T which leads to the smallest equal error rate on the entire dataset of signatures of all users is used later on. The size of this parameter can also be used as a measure for the quality of the classifier. A large T means the decision threshold can be set further from the mean of S_T which generally will lead to smaller false reject rates. The decision threshold is then calculated as:

$$\text{DecisionThreshold} = \text{mean}(S_T) - T \cdot \text{std}(S_T) \tag{2.5}$$

2.4.3 Probability Estimation

When combining classifier score outputs to obtain a combined classifier with higher performance, it is crucial to limit the influence of each separate classifier in a meaningful way. The tanh normalization scheme presented by Jain et al. [15] proved to be insensitive to outliers in the training data as well as being robust and efficient. The parameters μ and σ refer to the mean and standard deviation of S_T respectively. This tanh estimation scheme is used to normalize the raw score from each score output which is used in the fusion step.

$$P[Score] = 1 - \frac{1}{2}\left(\tanh\left(\frac{1}{2}\frac{\text{Score} - \mu}{\sigma}\right) + 1\right) \tag{2.6}$$

2.4.4 Score Fusion

After normalization of each score that should be utilized in the fusion step, the final score is calculated as the weighted sum:

$$\text{score}_{\text{fusion}} = \frac{\sum\limits_{i=0}^{\#\text{Classifiers}} T_i \cdot \text{score}_{\text{classifier}_i}}{\sum\limits_{i=0}^{\#\text{Classifiers}} T_i} \tag{2.7}$$

The T_i are the decision function parameters discussed in section 2.4.2. To understand why the thresholds T_i are used to weight the scores of the different classifiers, note that the scores are normalized using eq. (2.6) before the thresholds T_i are estimated. In the following we assume tanh() to be approximately linear in the range of the inputs we reach with $\frac{1}{2}\frac{\text{Score}-\mu}{\sigma}$. Furthermore we assume that the training set score distribution allows us to accurately estimate μ and σ used in tanh estimation. With these two assumptions we can deduce that the mean and standard deviation of the tanh estimated scores are the same for any users score distributions. The threshold values T_i then can be interpreted as measure for the spread between the positive and negative signature sample scores. A small threshold value T_i means the EER is reached close to the positive score distribution whereas a large threshold indicates that the false accept rate starts rising further from the positive score distribution. The proposed fusion scheme rewards classifiers with high threshold values T_i which means classifiers which deliver a high spread between negative and positive signature scores.

Chapter 3

Experiments

In this chapter, the performance of the feature- and function-based systems is analyzed. Because the two systems have free parameters (e.g. which feature vector to use, the one-class SVM parameters) we try to maximize the performance of the systems before using score fusion to create the final verification system. From the score fusion we expect to obtain a system which performs at least as good as the best performing system used in the fusion step but ideally the fusion system would deliver superior performance.

3.1 Experimental Protocol

The cross-validation scheme used in all the experiments described below is different from the one that is usually utilized. Instead of defining the number of cross-validation steps, we define the training set size. Similar to a classical cross-validation scheme, the set of positive samples is split into two parts: the training- and validation set. The training sets of all cross-validation runs are disjoint from each other. This method was chosen because we expect the training set size to have a major influence on the performance of the system. For the following experiments we use a training set size of 5.

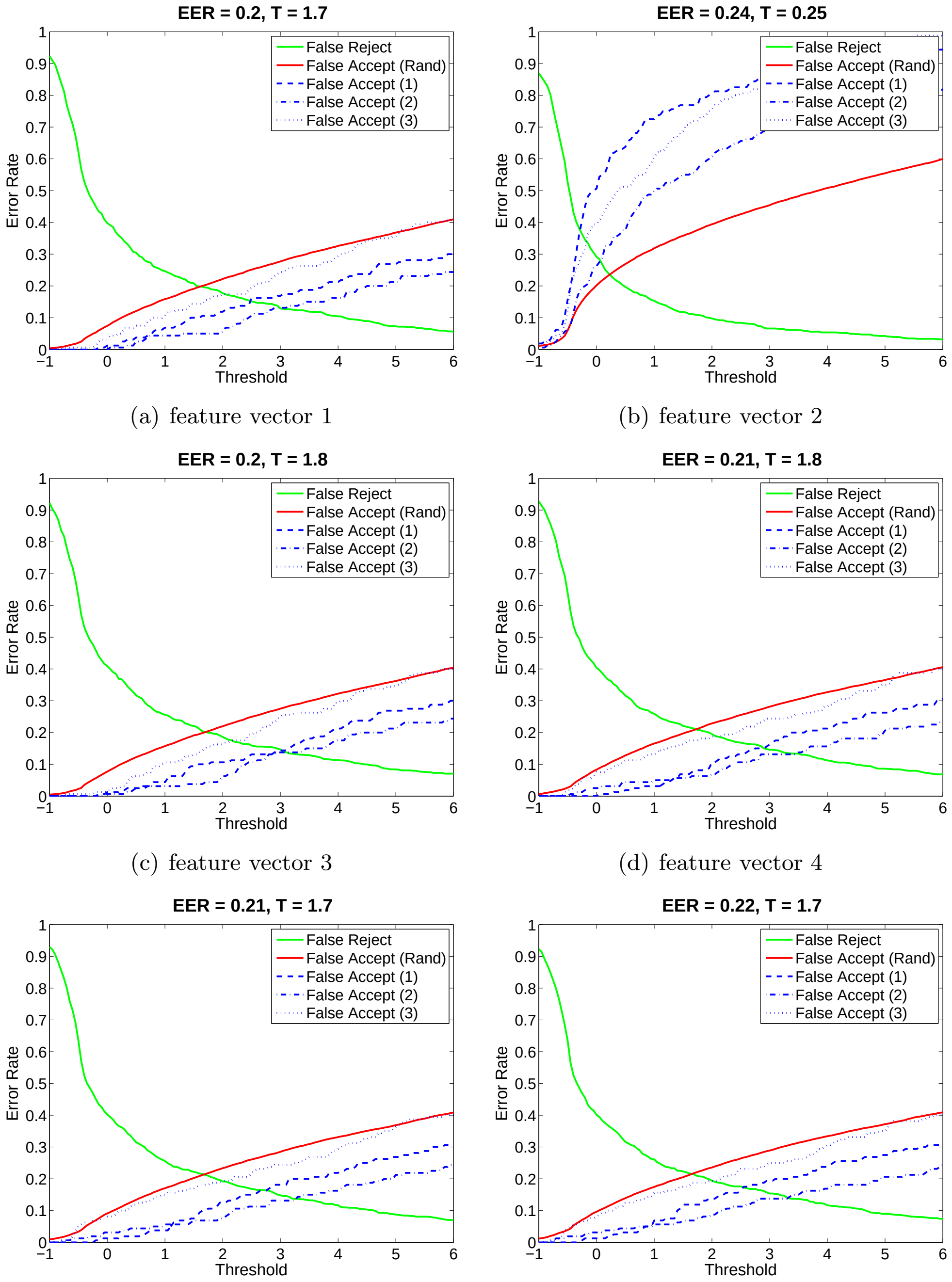
3.2 Feature-based Verification System

For the complete list of features extracted from the signature traces see table 2.1. In order to maximize the performance of the feature-based system, the feature subsets described in section 2.2.1 were used.

DET curves and equal error rates were produced and calculated using the decision threshold discussed in section 2.4.2. All DET curves show the false reject rate along with the false accept rates for the four forgery types (random, 1, 2, and 3).

3.2.1 Mahalanobis Distance

The Mahalanobis distance measure can be computed without any parameter tuning. The resulting DET curves for all the feature vectors can be found in fig. 3.1. The equal error rate is very similar throughout the experiments. Still the performance of feature vector 2 has to be considered to be significantly lower than the performance of all the other feature vectors. The equal error rate for the random forgeries is not much higher compared to the other feature vectors, but the range of thresholds that leads to a well performing system is very narrow compared to the others. We expected to have generally lower performance on the forged signatures. Surprisingly this is only true for feature vector 2. For all other feature vectors, the forgery EER is consistently lower than the EER of the random forgeries. Interestingly the forgery types 1, 2 and 3 behave as expected. The more knowledge the forger has, the better the forgeries become. The fact that skilled forgeries reach higher false accept rates could mean that many features are similar amongst users when they sign with their own signature. As soon as a user tries to forge a different signature, these features start to deviate because the user is in an unnatural situation. As an example, consider the total signature time. Many users have a total signature time which is smaller than 3 seconds. When trying to forge a complex signature, the total signature time tends to increase and therefore lowering the false accept probability. In feature vector 2 (pressure features) the fraction of features behaving like this seems to be smaller compared to all the other feature vectors. Therefore the more knowledge the forger has, the higher the false accept rate gets. The much lower performance delivered by the pressure based feature vector 2 could be explained by a higher variance of feature values in the training set or a higher feature value correlation between users.



(a) feature vector 1

(b) feature vector 2

(c) feature vector 3

(d) feature vector 4

(e) feature vector 5

(f) feature vector 6

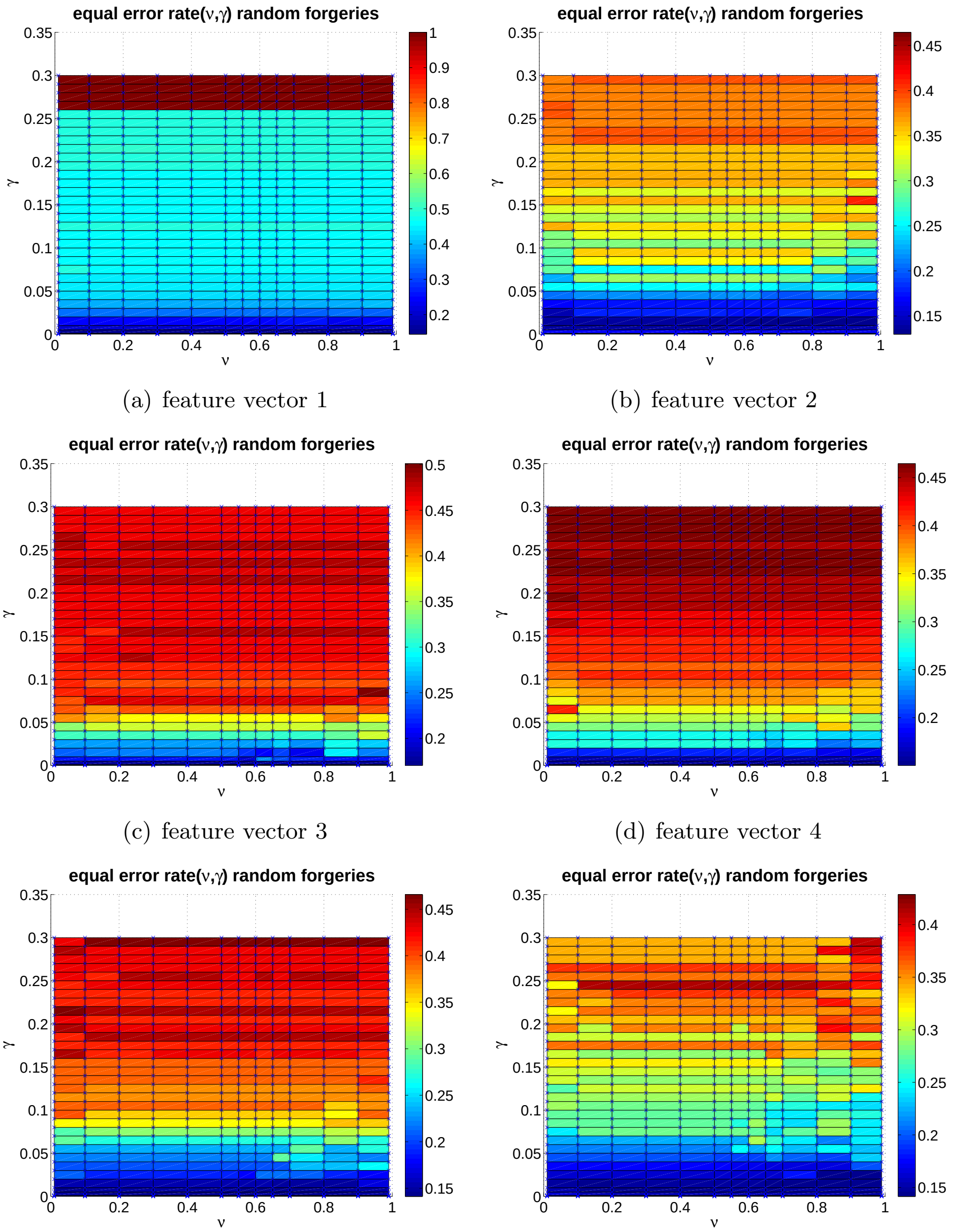
Figure 3.1: Mahalanobis distance measure DET curves

3.2.2 Support Vector Machine

The SVM implementation is based on LibSVM [16]. This library is publicly available and has interfaces to a large number of programming languages. LibSVM was modified such that no threshold is applied to the hypersphere distance output of the classification. Instead of getting a binary decision, we would like to get a distance measure (distance from hypersphere) which will be much more helpful when performing score fusion. The one-class SVM has several different parameters which influence its performance. Following Hsu et al. [10], we used an Radial Basis Function Kernel. This leaves two independent parameters which we have to tune for our problem: ν, γ. Note that getting a recall rate of 1 was impossible using the decision threshold given by LibSVM, the precision however was extremely high (≈ 1) for any set of parameters ν and γ and any forgery set. A likely reason for this behaviour of the one-class SVM is model overfitting to the training data.

Parameters ν and γ for each feature vector were found using a parameter grid search. For the results of this grid search see fig. 3.2. For each parameter set ν and γ under consideration, the equal error rate is shown. Generally values for $\gamma > 0.1$ deliver lower performance than values $\gamma < 0.1$. Especially if the feature vector has more elements (feature vectors 1, 3 and 4) the regions with good equal error rates are smaller compared to the shorter feature vectors (2, 5 and 6).

Based on figure 3.2 the optimal parameters ν and γ are estimated for each feature vector. The detection performance for each feature vector along with the optimal set of parameters for the svm ν and γ is shown in figure fig. 3.3. The equal error rates are very similar for all the feature vectors. However from the DET curves we can see that fig. 3.3(a), fig. 3.3(b) and fig. 3.3(c) reach the equal error rate at lower thresholds compared to the other parameter settings. Also a small variation in the threshold results in a bigger variation in the error rates. A classifier with a higher threshold and a lower sensitivity to the choice of the threshold is very desirable because we can only estimate the threshold based on the sparse training set. A higher threshold generally leads to higher performance. To understand this, note that the scores are estimated using the tanh estimation scheme presented in eq. (2.6). We expect scores with mean 0.5 and fixed standard deviation. The higher the threshold is, the larger the gap between the positive and negative score values is.



(a) feature vector 1

(b) feature vector 2

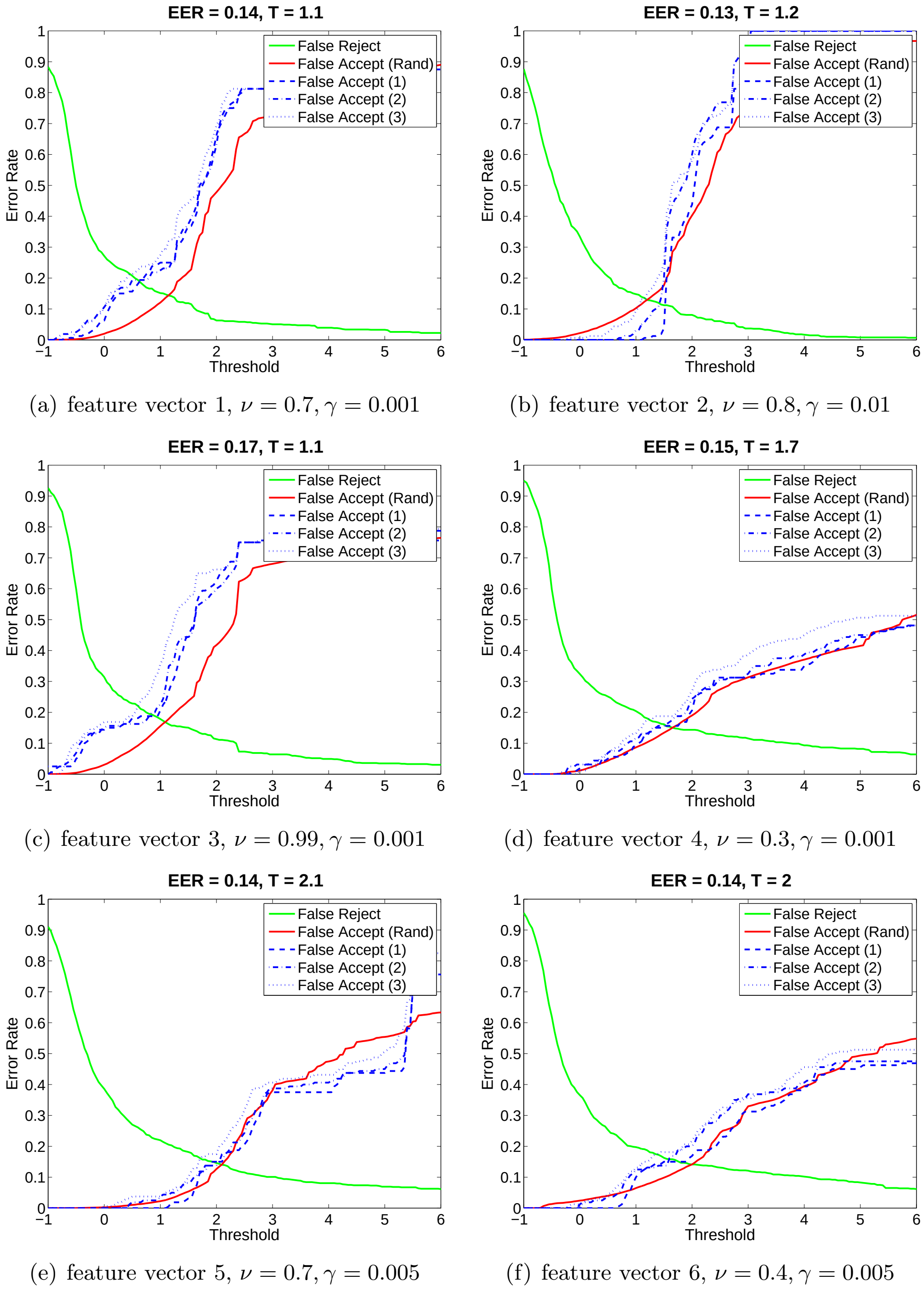
(c) feature vector 3

(d) feature vector 4

(e) feature vector 5

(f) feature vector 6

Figure 3.2: SVM grid search



(a) feature vector 1, $\nu = 0.7, \gamma = 0.001$

(b) feature vector 2, $\nu = 0.8, \gamma = 0.01$

(c) feature vector 3, $\nu = 0.99, \gamma = 0.001$

(d) feature vector 4, $\nu = 0.3, \gamma = 0.001$

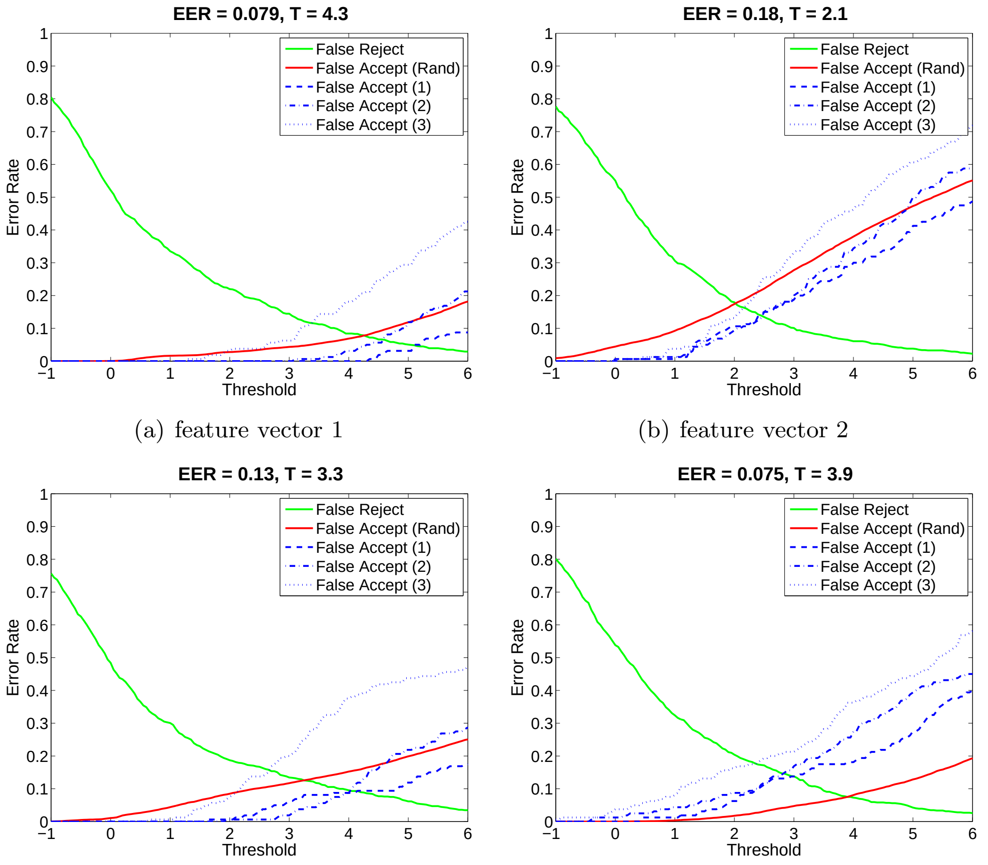
(e) feature vector 5, $\nu = 0.7, \gamma = 0.005$

(f) feature vector 6, $\nu = 0.4, \gamma = 0.005$

Figure 3.3: SVM DET courves

3.3 Function-based Expert – Dynamic Time Warping

The dynamic time warping algorithm does not have free parameters except the choice of the feature vector which it utilizes. The performance was evaluated using the feature vectors presented in section 2.3.1. The resulting DET curves are shown in fig. 3.4. We can see that unlike the results from the SVM experiments, a longer feature vector seems to have a positive impact on the performance. Comparing the equal error rates of feature vectors 2, 3 and 4, which are composed of the signal value and the signal derivative value, we can see that the y coordinate seems to deliver the best performance. The EER of feature vector 4 is even lower than the EER of feature vector 1. When it comes to the forgeries however, feature vector 1 clearly delivers better performance.



(a) feature vector 1

(b) feature vector 2

(c) feature vector 3

(d) feature vector 4

Figure 3.4: DTW distance measure DET courves

3.4 Fusion of Feature and Function Expert

As we can see from the previous sections, the one-class SVM performs slightly better than the Mahalanobis distance measure. A reason to use the Mahalanobis distance could be the limited processing power of the target hardware. Since the SVM based classifier works reasonably fast on the HTC Desire, only the SVM score was utilized for score fusion.

In this section, fusion of the function-based DTW and feature-based SVM signature verification method is evaluated. The fusion process is visualized in fig. 3.6 which shows a single cross-validation run for a single user. The figures fig. 3.6(a), fig. 3.6(b) and fig. 3.6(c) are the same except the forgery type shown. As expected, the more information is revealed to the forger (forgery type 1 trough 3) the higher the scores of the forgeries get. The decision threshold line visualizes where the weighted sum (2.7) of the DTW and SVM scores are equal to the decision threshold (2.5). All samples to the upper right of the decision threshold line are considered to be genuine, all samples to the lower left are considered to be forgeries. Using only five training samples, the parameters T used to weight the sum (2.7) are $T_{\text{SVM}} = 2$ and $T_{\text{DTW}} = 4.3$. When we project the score distributions to the x- and y-axis respectively, we can see why the two thresholds are that different. When projected to the x-axis (DTW) the scores of genuine and forged samples can be separated whereas the projection to the y-axis (SVM) leads to an overlap between genuine- and forged signatures. The different threshold values lead to a decision threshold which prioritizes the DTW score over the SVM score. The equal error rate of the combined DTW and SVM classifier for random forgeries is only slightly lower than compared to the individual systems.

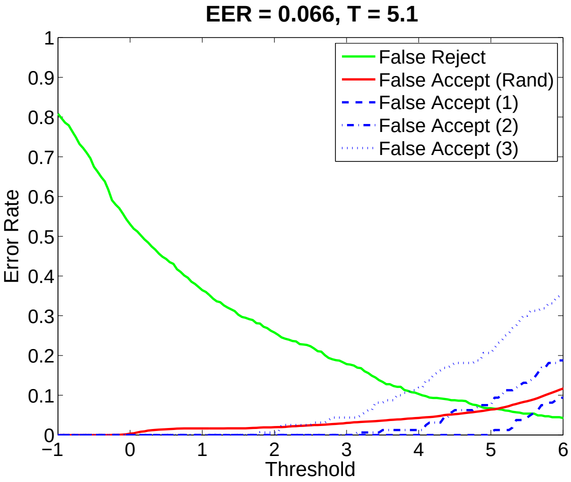
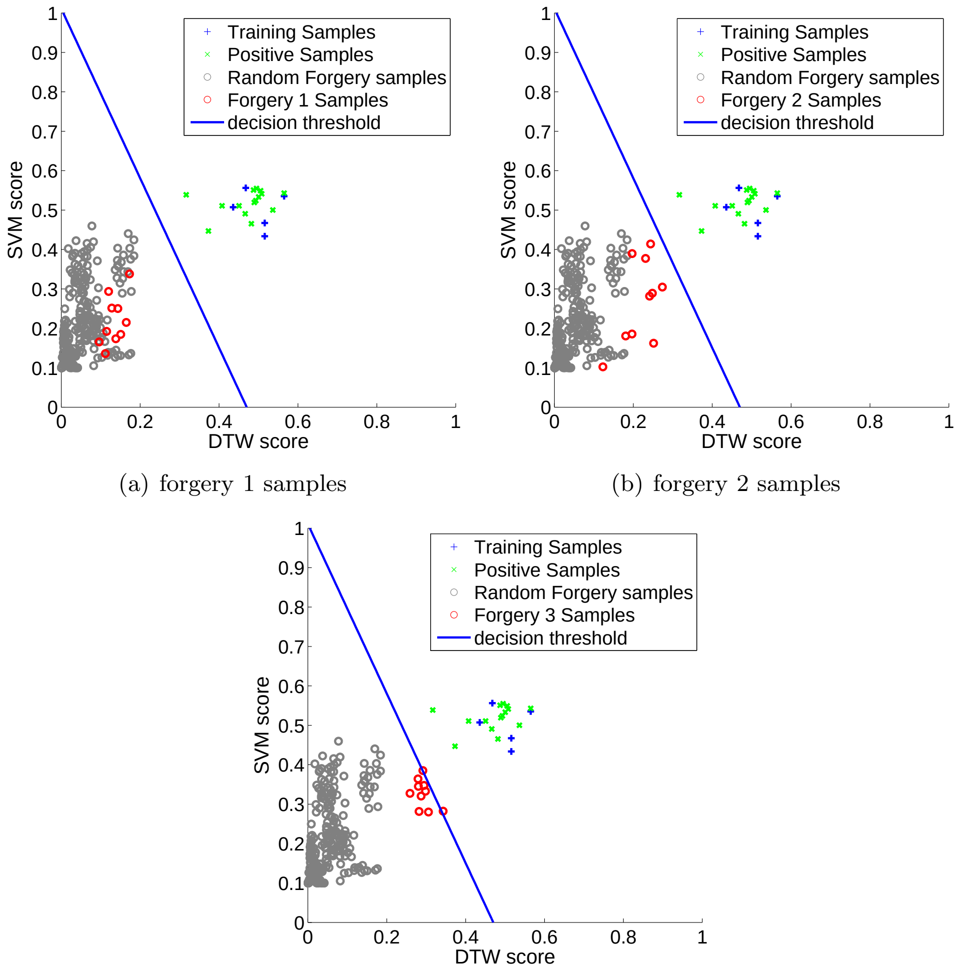


Figure 3.5: DET curve of combined SVM and DTW classifiers



(a) forgery 1 samples

(b) forgery 2 samples

(c) forgery 3 samples

Figure 3.6: Score fusion visualization

Chapter 4

Conclusions

4.1 Summary

Signature verification is a well documented topic in the literature ([17], [18], [9], [4], [14], [5], [19], [20], [6], [21]). However, previous work was focused on signature acquisition using a pen either on a graphical tablet or a resistive touchscreen. The application of signature verification methods on capacitive touchscreens has been discussed theoretically by Mendaza-Ormaza et al. [22]. The presented methods to perform signature verification are not new, but the results show that even a signature entered with the finger instead of a pen can be utilized to perform signature verification with reasonable accuracy. In addition to this, the introduction of pressure signal related features in the global feature-based classifier was analyzed. On one hand, using pressure features only did not improve detection performance. On the other hand, pressure related features had similarly high Fisher Discriminant Ratios as other features and therefore played a crucial part in the experiments with the feature vectors containing high Fisher Discriminant Ratio features only.

Even when extracting features from the pressure signal only, the performance of the system delivers fair classification performance (see fig. 3.1(b), fig. 3.3(b) and fig. 3.4(b)).

In addition to the experiments presented in chapter 3, the set of experiments was performed using a training set size of 10 instead of 5. The results for both training set sizes are summarized in section 4.1 and section 4.1. The equal error rate of the SVM classifier is improved drastically. Also note that the feature vector 5 and not feature vector 6 delivers the best performance when increasing the training set size from 5 to 10. The increasing number of training samples seems to allow the estimation of more features without overfitting the SVM. The dynamic time warping score does not seem to gain EER performance. The threshold at which the best EER is reached however differs a lot (2.7 instead of 4.3). During training, the DTW method finds a prototypical signature. If we have more training samples, this prototypical signature will become better which means thatthe distribution of validation scores will become lower.

Classifier	Parameters	$T_{\text{classifier}}$	Equal Error Rate
DTW	feature vector 1	4.3	7.9 %
SVM	feature vector 6, $\nu = 0.4$, $\gamma = 0.005$	2	14.2 %
Fusion	see above	5.1	6.6 %

Table 4.1: Result overview with 5 training samples

Classifier	Parameters	$T_{\text{classifier}}$	Equal Error Rate
DTW	feature vector 1	2.7	7.9 %
SVM	feature vector 5, $\nu = 0.8$, $\gamma = 0.03$	2.7	6.0 %
Fusion	see above	4.3	5.5 %

Table 4.2: Result overview with 10 training samples

4.2 Limitations

The size of the gathered dataset is very limited. The results obtained from this dataset have to be considered carefully. A single outlier in the dataset caused a very big part of the equal error rate. A dataset this small is very sensitive to outliers as well as the the signing behaviour of the sparse sample of signers. In order to get harder results, a larger dataset is required.

The performance measures used to quantify the results presented in this work are calculated using random forgeries only. Lower performance has to be expected when considering forgeries also. The impact of more skilled forgeries on the respective error rates can be seen in fig. 3.6. If good forgeries are available, the equal error rate should be optimized with respect to the more qualified forgery samples. A system optimized in such a way will most probably lead to good performance against random forgeries also.

The general system performance highly depends on the combination of two user properties. Firstly the variance in a users signature highly influences system performance. Higher variance in the training set means that the system will allow lower scores to be accepted as valid signature, whereas lower variance will lead to more restrictive decision thresholds. Secondly the signature complexity is very important as well. If a user has a very complex signature, it will be harder to forge than a signature which is very simple. The overall system performance depends on the combination of both, signature complexity and variance. Generally, low variance and high complexity is the most desirable combination. In our dataset either the signature complexity was high or the variance between signatures was low such that forgeries could be detected with reasonable performance. However the performance can be nullified by training the system with signatures of low complexity with very high variance. If trained in such a way, the system will accept virtually any signature as valid because the training set score distribution will require extremely permissive decision thresholds.

Practical limitations of the presented system lie in the fact that the training set is acquired without using any quality control. An outlier in the training set will lead to decision thresholds which allows an attacker to forge the signature easier than necessary. Also a persons signature usually is not constant but evolves over time. Since training only initially performed, the model against which signatures are tested does not adapt. An increasing false reject rate therefore has to be expected.

4.3 Future Work

The choices about classifiers and fusion methods during this work were drastically limited by the time frame for this work. Additional classifiers could use the Weka OneClassClassifier in conjunction with any two-class classification method. Also other anomaly detection methods such as nearest neighbor, clustering or spectral techniques could be applied to the problem of signature verification. Also the score fusion approach was not tuned in any way. Instead of using a weighted sum, one could try to combine the scores by simple multiplication.

The local algorithm could be further improved by using more than just the cost of the best warping path, but also the mean, max, min time warp as a score output. Also the feature vector used to calculate the cost function between two samples could be expanded with additional features like the curvature, the acceleration etc. Additionally a Hidden Markov Model verification system would most likely deliver superior performance compared to the presented DTW method.

Another issue in the current system is the sensitivity to outliers during the training. If we have a big outlier in the training set, the threshold estimate will be influenced because the standard deviation of the training set distribution estimate will increase. Methods to calculate outlier-insensitive mean and standard deviations on the training set distribution could help in overcoming this problem.

In addition to the traditional approach of learning a model once and then using this model to test all upcoming test samples, the system could be extended to allow continuous learning. In a first step, the score distribution of the positive class could be estimated more precisely with every testing sample. In a second step, the model could be retrained after having collected additional positive signature samples. Because a signature is not static but changes over time, the ability of the algorithm to adapt to these changes is crucial and should be considered.

Bibliography

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

[2] Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., Satue, A., Hernaez, I., Igarza, J.J., Vivaracho, C., Escudero, D., Moro, Q.I.: MCYT baseline corpus: a bimodal biometric database. Vision, Image and Signal Processing, IEE Proceedings- **150** (2003) 395–401

[3] yan Yeung, D., Chang, H., Xiong, Y., George, S., Kashi, R., Matsumoto, T., Rigoll, G.: Svc2004: First international signature verification competition. In: In Proceedings of the International Conference on Biometric Authentication (ICBA), Hong Kong, Springer (2004) 16–22

[4] Martinez-Diaz, M.: Dynamic signature verification for portable devices. Master's thesis, Universidad Autonoma de Madrid (2008)

[5] Lee, L.L., Berger, T., Aviczer, E.: Reliable on-line human signature verification systems. IEEE Trans. Pattern Anal. Mach. Intell. **18** (1996) 643–647

[6] Fierrez-Aguilar, J., Nanni, L., Lopez-Peñalba, J., Ortega-Garcia, J., Maltoni, D.: An on-line signature verification system based on fusion of local and global information. In Kanade, T., Jain, A., Ratha, N., eds.: Audio- and Video-Based Biometric Person Authentication. Volume 3546 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2005) 627–656

[7] Richiardi, J., Ketabdar, H., Drygajlo, A.: Local and global feature selection for on-line signature verification. In: Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on. (2005) 625 – 629 Vol. 2

[8] Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. **41** (2009)

[9] Mendaza-Ormaza, A., Miguel-Hurtado, O., Rubio-Polo, I., Alonso-Moreno, R.: On-line signature biometrics using support vector machine. In Brömme, A., Busch, C., Hühnlein, D., eds.: BIOSIG. Volume 155 of LNI., GI (2009) 185–188

[10] Hsu, C.W., Chang, C.C., Lin, C.J.: A Practical Guide to Support Vector Classification (2000)

[11] Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Comput. **12** (2000) 1207–1245

[12] Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13** (2001) 1443–1471

[13] Hempstalk, K., Frank, E., Witten, I.H.: One-class classification by combining density and class probability estimation. In: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I. ECML PKDD '08, Berlin / Heidelberg, Springer-Verlag (2008) 505–519

[14] Martens, R., Claesen, L.: On-line signature verification by dynamic time-warping. Pattern Recognition, International Conference on **3** (1996) 38

[15] Jain, A.K., Nandakumar, K., Ross, A.A.: Score normalization in multimodal biometric systems. Pattern Recognition **38** (2005) 2270–2285

[16] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27:1–27:27 Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[17] Yang, L., Widjaja, B.K., Prasad, R.: Application of hidden markov models for signature verification. Pattern Recognition **28** (1995) 161 – 170

[18] Piyush Shanker, A., Rajagopalan, A.N.: Off-line signature verification using dtw. Pattern Recogn. Lett. **28** (2007) 1407–1414

[19] Kashi, R., Hu, J., Nelson, W., Turin, W.: A hidden markov model approach to online handwritten signature verification. International Journal on Document Analysis and Recognition **1** (1998) 102–109 10.1007/s100320050010.

[20] Ji, H.W., Quan, Z.H.: Signature verification using wavelet transform and support vector machine. In Huang, D.S., Zhang, X.P., Huang, G.B., eds.: Advances in Intelligent Computing. Volume 3644 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2005) 671–678

[21] Fierrez, J., Ortega-Garcia, J., Ramos, D., Gongalez-Rodriguez, J.: Hmm-based on-line signature verification: Feature extraction and signature modeling. Pattern Recognition Letters **28** (2007) 2325–2334

[22] Mendaza-Ormaza, A., Miguel-Hurtado, O., Sánchez-Reillo, R., Uriarte-Antonio, J.: Analysis on the resolution of the different signals in an on-line

handwritten signature verification system applied to portable devices. In: Security Technology (ICCST), 2010 IEEE International Carnahan Conference on. (2010) 341 –350