



Machine Learning and Pattern Recognition

Fingerprint Spoofing Detection

Matteo Borzi
s280104

ACADEMIC YEAR 2022-2023

Contents

1	Introduction	3
1.1	Abstract	3
1.2	Dataset overview	3
1.3	Feature Analysis	3
1.3.1	Feature distributions	3
1.3.2	Feature correlations and dimensionality reduction	5
2	Model selection and validation	8
2.1	Validation methodology	8
2.2	Classifiers	9
2.2.1	Multivariate Gaussian classifiers	9
2.2.2	Logistic regression classifiers	10
2.2.3	Support Vector Machines	14
2.2.4	Gaussian Mixture Model classifiers	18
2.3	Most promising models	20
2.4	Score calibration	21
2.4.1	Uncalibrated models	21
2.4.2	Calibrated models	22
2.5	Model fusion	23
3	Evaluation	24
3.1	Multivariate Gaussian classifiers	25
3.2	Logistic regression classifiers	25
3.3	Support Vector Machines	27
3.4	Gaussian Mixture Model classifiers	28
3.5	Model fusion	29
3.6	Actual best model	30
4	Conclusions	31

Chapter 1

Introduction

1.1 Abstract

The goal of the Fingerprint Spoofing Detection task is to tell authentic fingerprints apart from spoofed ones with Machine Learning tools. In order to achieve this goal, many binary classification models are built and compared on a sample dataset.

1.2 Dataset overview

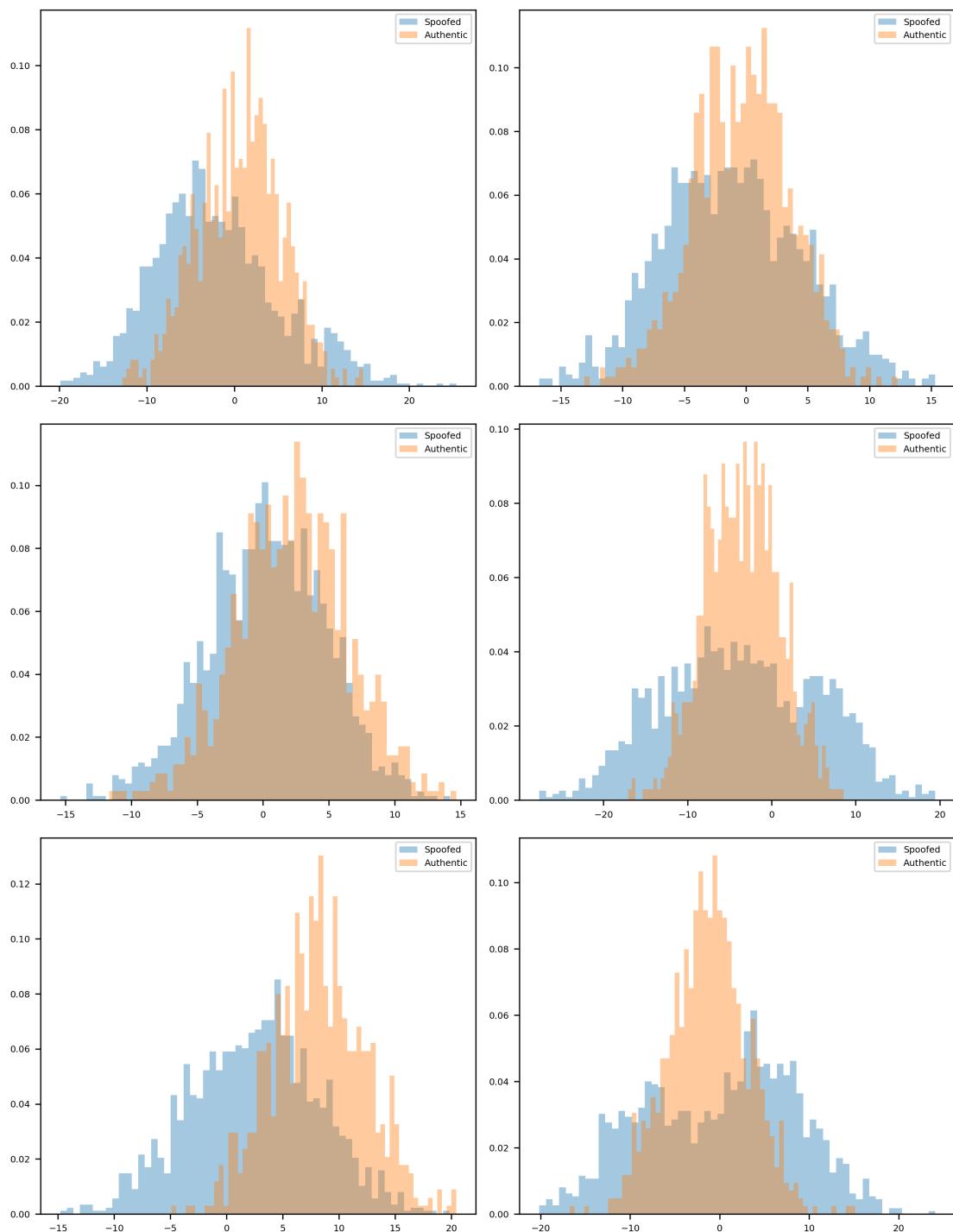
The dataset is made of fingerprint images, represented by low-dimensional embeddings for the sake of tractability of the problem. Each embedding is a continuous-valued vector that represent a point in the 10-dimensional feature space and has no physical interpretation. The spoofed fingerprint samples can be divided into 6 groups depending on the spoofing method, but sub-class labels are not available.

The training set is composed of 2325 samples and it is imbalanced, since there are 800 authentic fingerprint images (label 1) and 1525 spoofed ones (label 0). The test set contains 7704 samples.

1.3 Feature Analysis

1.3.1 Feature distributions

The following histograms show the distribution of the raw features.



1.3 – Feature Analysis

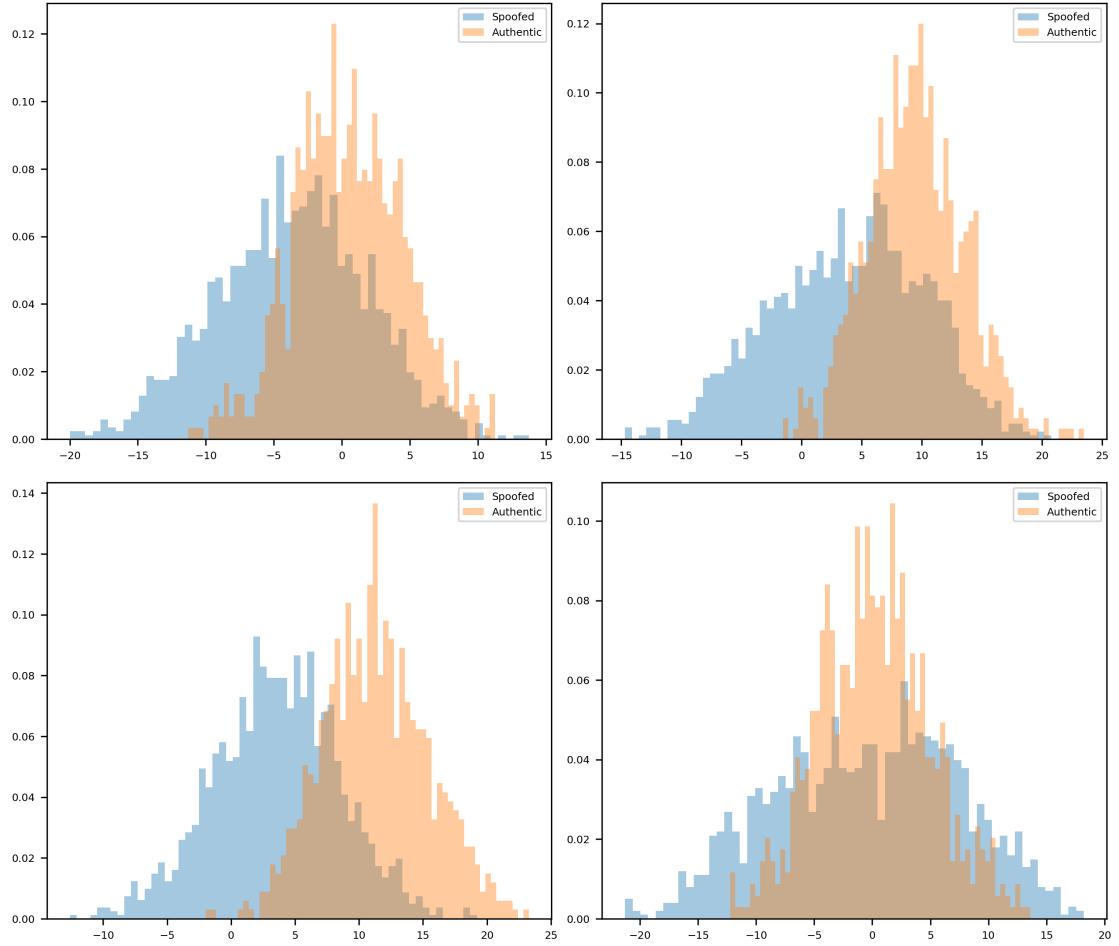


Figure 1.1: Histograms for per-class feature distributions

Feature distributions of samples belonging to the target class generally resemble Gaussian distributions (especially features 6 and 9), while some of the non-target class distributions (features 4 and 6) resemble a mixture of Gaussian distributions, coherently with the presence of sub-classes. Histograms of features 5 and 9 are quite well separated, making their contribution relevant. All the range scales are similar and there are very few outliers, thus we expect Z-Normalization not to have a significant contribution to models improvement.

1.3.2 Feature correlations and dimensionality reduction

General and per-class feature correlations are shown in Figure 1.2.

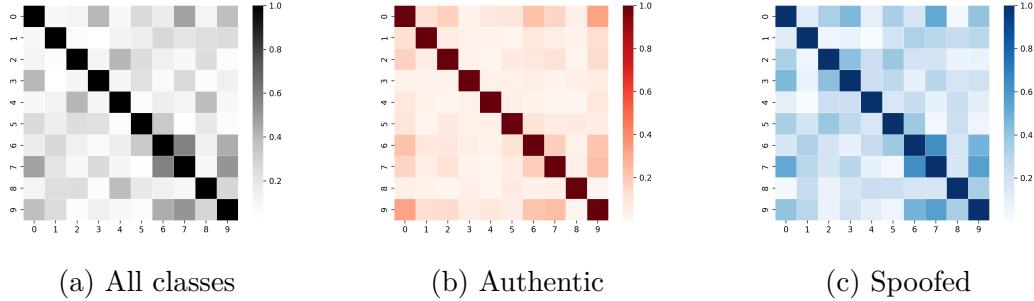


Figure 1.2: Pearson Correlation heatmaps for dataset features

Pearson correlation heatmaps generally show weak correlations between features. Whereas both target and non-target classes are weakly correlated, there is still an increase in non-target class correlations, especially between feature 7 and features 6 and 9, thus we expect Full Covariance Multivariate Gaussian classifier (*MVG*) and the Naïve Bayes MVG to outperform the Tied Covariance ones.

Let's now analyse the variance with respect to the number of PCA dimensions. As features are already weakly correlated, we expect PCA not to be much effective.

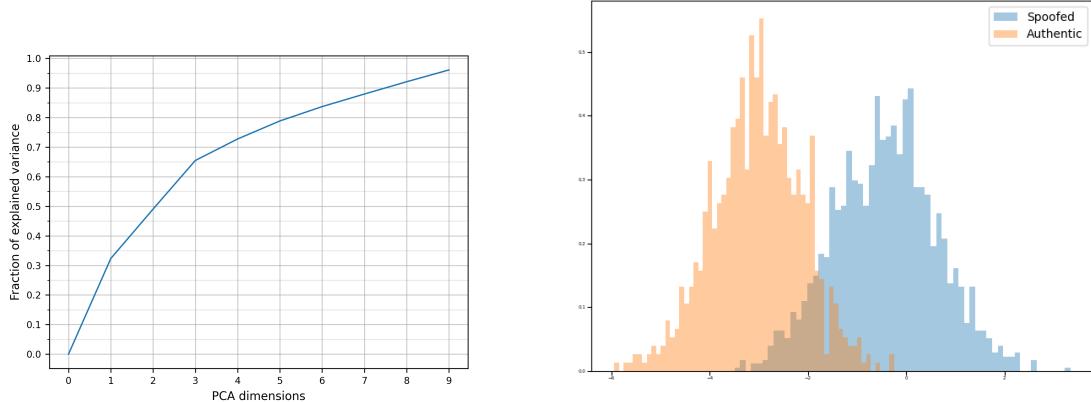


Figure 1.3: PCA explained variance

Figure 1.4: LDA direction for raw dataset

Applying PCA with 9 dimensions should allow the dataset to retain roughly 97% of its variance, while with fewer dimensions we might start losing information. As expected, PCA is not likely to impact considerably the models performance. We still consider the option with $m = 9$.

With Linear Discriminant Analysis (LDA) we can determine that a linear approach can be used to build a decent classifier (Figure 1.4), but we expect non-linear models to perform much better by looking at the features distributions in the scatter plots in Figure 1.5.

1.3 – Feature Analysis

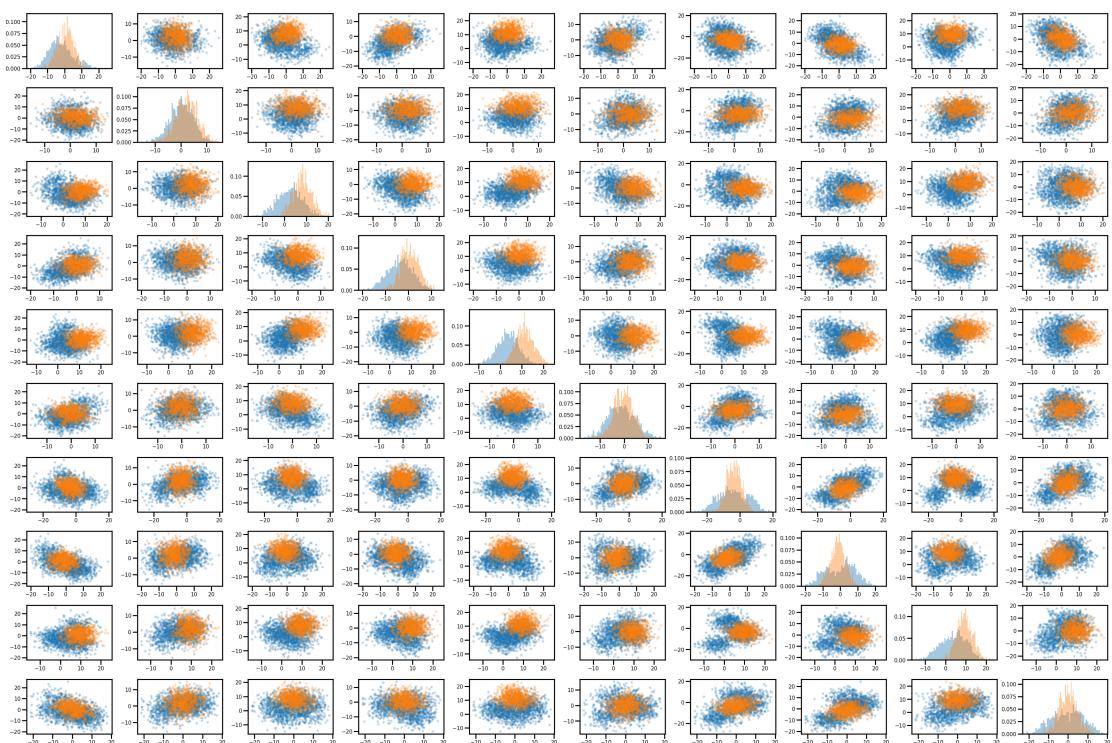


Figure 1.5: Histograms and scatter plots for all the features for the raw dataset

Chapter 2

Model selection and validation

2.1 Validation methodology

A K-Fold cross validation approach with $K = 5$ has been chosen both to increase the robustness of the models and to mitigate the imbalance between classes in the training set. The application we are building has a balanced prior π and a much higher cost for false positives C_{fp} , since an incorrect classification of a spoofed fingerprint would have more serious security implications than an authentic fingerprint classified as spoofed:

$$(\pi, C_{fn}, C_{fp}) = (0.5, 1, 10)$$

which corresponds to the working point with effective prior $\tilde{\pi}$:

$$(\tilde{\pi}, \tilde{C}_{fn}, \tilde{C}_{fp}) = (0.09, 1, 1)$$

However, validating the models with different priors may be notable, thus we consider two other applications with different (lower) priors and same misclassification costs:

$$(\pi, C_{fn}, C_{fp}) = (0.2, 1, 10), (0.4, 1, 10)$$

The chosen metric for evaluation is *minDCF* (normalized minimum Detection Cost Function): we will compute the actual normalized DCF values only after this preliminary phase.

2.2 Classifiers

Here is a list of binary classifiers and their validation result in order to select the best performing models.

2.2.1 Multivariate Gaussian classifiers

Binary Multivariate Gaussian Classifiers (MVG) are generative models that assume data samples are independent and identically distributed with a per-class multivariate Gaussian probability density function (*pdf*):

$$(X_i \mid C_i = c, \theta) \sim (X \mid C = c, \theta) \sim \mathcal{N}(\mu_c, \Sigma_c)$$

- **Full Covariance MVG** Each class *pdf* is described by parameters (μ_c, Σ_c) . The separation rule based on the log-likelihood ratio *llr* is quadratic in \mathbf{x} :

$$llr(\mathbf{x}) = \log \frac{\mathcal{N}(\mathbf{x} \mid C = h_1)}{\mathcal{N}(\mathbf{x} \mid C = h_0)} = \log \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1^{-1})}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0^{-1})} = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{b} + c$$

where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ is the precision matrix and $\mathbf{A} = -\frac{1}{2}(\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_0)$.

- **MVG with Tied Covariance matrix** Per-class *pdfs* are assumed to have the same covariance matrix:

$$(X_i \mid C_i = c, \theta) \sim (X \mid C = c, \theta) \sim \mathcal{N}(\mu_c, \Sigma)$$

Tied covariance models have a linear separation surface $llr(\mathbf{x}) = \mathbf{x}^T \mathbf{b} + c$.

- **MVG with Naïve Bayes assumption** Assuming that the M features are uncorrelated, each feature can be described by a univariate Gaussian *pdf*:

$$f_{X|C}(\mathbf{x} \mid c) \approx \prod_{j=1}^M f_{X_{[j]}|C}(x_{[j]} \mid c) = \prod_{j=1}^M \mathcal{N}(x_{[j]} \mid \mu_{c,[j]}, \sigma_{c,[j]}^2)$$

The resulting covariance matrix $\boldsymbol{\Sigma}_c$ is thus diagonal.

- **MVG with Tied Covariance under Naïve Bayes assumption** Both Naïve Bayes and tied covariance assumption are considered, resulting in a diagonal covariance matrix that is the same for all the classes.

For each Gaussian classifier both raw and preprocessed data with PCA have been reported in Table 2.1. Since features are weakly correlated, we also tried PCA preprocessing with $m = 10$ in order to diagonalize the covariance matrix.

As expected, PCA does not improve the models, with the exception of the Naïve Bayes one, since the diagonalized covariance matrix preserves the directions with

the highest variance. The best performing models are the quadratic ones, with the Full Covariance MVG outperforming the Diagonal Covariance matrix one.

Model	PCA	minDCF		
		$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
Full Covariance MVG	-	0.330	0.522	0.392
Full Covariance MVG	9	0.331	0.550	0.393
Full Covariance MVG	8	0.334	0.537	0.388
Full Covariance MVG	7	0.339	0.539	0.384
Tied Covariance MVG	-	0.474	0.641	0.510
Tied Covariance MVG	9	0.483	0.629	0.527
Tied Covariance MVG	8	0.475	0.667	0.518
Tied Covariance MVG	7	0.478	0.671	0.520
Naïve Bayes MVG	-	0.471	0.686	0.528
Naïve Bayes MVG	10	0.366	0.591	0.432
Naïve Bayes MVG	9	0.367	0.606	0.438
Naïve Bayes MVG	8	0.359	0.581	0.419
Naïve Bayes MVG	7	0.359	0.589	0.427
Tied Naïve Bayes MVG	-	0.545	0.746	0.636
Tied Naïve Bayes MVG	10	0.525	0.735	0.609
Tied Naïve Bayes MVG	9	0.542	0.739	0.629
Tied Naïve Bayes MVG	8	0.539	0.739	0.628
Tied Naïve Bayes MVG	7	0.534	0.741	0.625

Table 2.1: Validation results for Multivariate Gaussian Classifiers both with and without PCA

2.2.2 Logistic regression classifiers

Logistic Regression (*LogReg* or *LR*) binary classifiers are discriminative models. The score the optimal Bayes decision is performed on is the log-ratio between class-posterior probabilities:

$$s(\mathbf{x}; \mathbf{w}, b) = \log \frac{P(C = 1 \mid \mathbf{x}, \mathbf{w}, b)}{P(C = 0 \mid \mathbf{x}, \mathbf{w}, b)} = \mathbf{w}^T \mathbf{x} + b \leq t$$

where (\mathbf{w}, b) are the model parameters. Class posterior probability for the authentic fingerprints can be computed with the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$:

$$P(C = 1 \mid \mathbf{x}, \mathbf{w}, b) = \sigma(s(\mathbf{x}; \mathbf{w}, b)) = \sigma(\mathbf{w}^T \mathbf{x} + b)$$

Since Logistic Regression score depends on the empirical prior of the training data, we are minimizing a prior-weighted version of the objective function in order to take into account the effective prior $\tilde{\pi}$:

$$J(\mathbf{w}, b) = \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{\tilde{\pi}}{n_T} \sum_{i|z_i=1} \log(1 + e^{-z_i(\mathbf{w}^T \mathbf{x} + b)}) + \frac{1 - \tilde{\pi}}{n_F} \sum_{i|z_i=-1} \log(1 + e^{-z_i(\mathbf{w}^T \mathbf{x} + b)})$$

The regularization term $\frac{\lambda}{2} \|\mathbf{w}\|^2$ depends on the hyper-parameter λ for which we are considering a 10-log scale between 10^{-4} and 10^4 .

LogReg separation surfaces are linear, but a quadratic version of the model can be trained by considering an expanded feature space $\phi(\mathbf{x})$ in which the separation surface is linear:

$$\phi(\mathbf{x}) = \begin{bmatrix} \text{vec}(\mathbf{x} \mathbf{x}^T) \\ \mathbf{x} \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \text{vec}(\mathbf{A}) \\ \mathbf{b} \end{bmatrix}, \quad s(\mathbf{x}; \mathbf{w}, c) = \mathbf{w}^T \phi(\mathbf{x}) + c$$

Values of minDCF for prior-weighted linear LogReg models are reported in Figure 2.1, Figure 2.2 and Table 2.2.

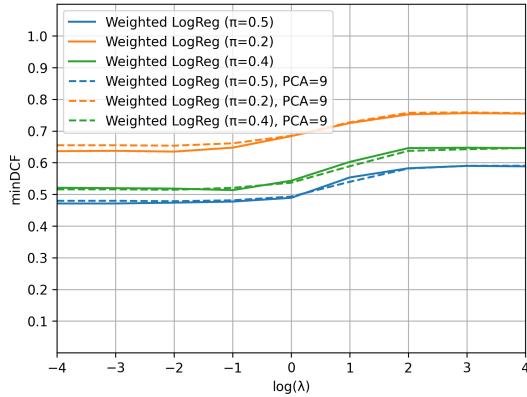


Figure 2.1: Hyper-parameter selection for Linear LR (raw data)

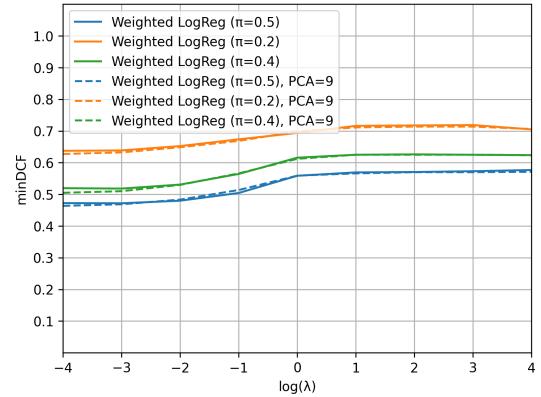


Figure 2.2: Hyper-parameter selection for Linear LR (Z-Norm)

Validation for Prior-weighted Linear Logistic Regression models

λ	minDCF (RAW)			minDCF (Z-Norm)		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
10^{-4}	0.471	0.636	0.521	0.472	0.637	0.519
10^{-3}	0.471	0.637	0.519	0.472	0.639	0.518
10^{-2}	0.474	0.635	0.518	0.480	0.652	0.531
10^{-1}	0.477	0.647	0.513	0.504	0.674	0.564
1	0.489	0.684	0.543	0.559	0.694	0.616
10	0.553	0.725	0.603	0.569	0.716	0.625
10^2	0.582	0.752	0.646	0.571	0.717	0.626
10^3	0.590	0.756	0.647	0.573	0.719	0.625
10^4	0.588	0.755	0.646	0.577	0.705	0.624
PCA $m = 9$						
10^{-4}	0.480	0.655	0.516	0.464	0.627	0.505
10^{-3}	0.480	0.655	0.516	0.469	0.632	0.510
10^{-2}	0.478	0.654	0.514	0.483	0.649	0.530
10^{-1}	0.481	0.661	0.521	0.514	0.669	0.566
1	0.493	0.685	0.537	0.559	0.699	0.612
10	0.540	0.728	0.589	0.566	0.711	0.625
10^2	0.581	0.757	0.637	0.570	0.714	0.625
10^3	0.590	0.759	0.642	0.570	0.714	0.625
10^4	0.590	0.756	0.646	0.571	0.706	0.624

Table 2.2: Validation results for prior-weighted Logistic Regression classifiers

Validation results are generally worse than generative ones: LogReg models used until now are linear, while we still expect better results with the LogReg ones (*Q-LogReg*). Z-score normalization has practically no effect on performance, being better than the model without preprocessing only combined with PCA with $m = 9$.

Q-LogReg classifiers have been trained with the same values of λ as the linear LogReg models. Validation results are reported in Figure 2.3, Figure 2.4 and Table 2.3.

Both linear and quadratic LogReg classifiers do not perform as good as Full Covariance MVG model. Again, Z-Score normalization has no impact on models performance. Moreover, these scores most likely need to be calibrated when dealing with the actual DCF costs: this option will be considered later.

We thus select the Q-LogReg model with $\lambda = 0.01$ and $m = 9$ for PCA on raw data, since results are slightly better.

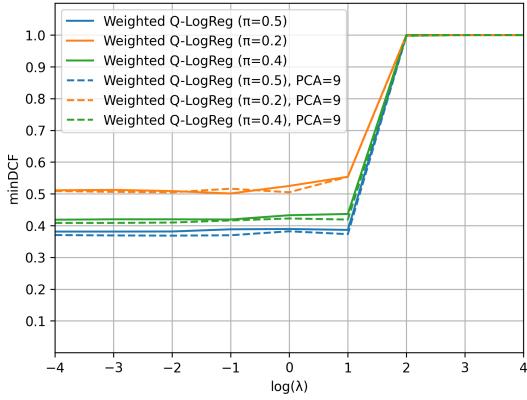


Figure 2.3: Hyper-parameter selection for Q-LogReg (raw data)

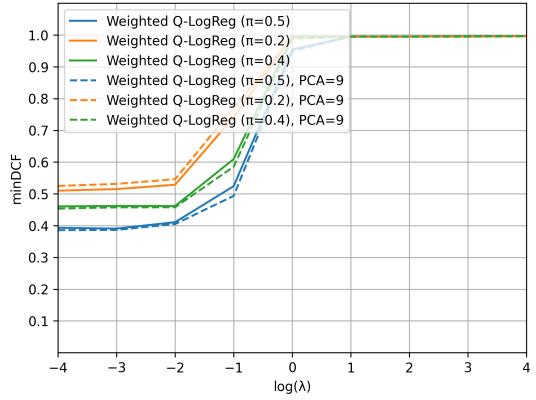


Figure 2.4: Hyper-parameter selection for Q-LogReg (Z-Norm)

Validation for Prior-weighted Quadratic Logistic Regression models

λ	minDCF (RAW)			minDCF (Z-Norm)		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
10^{-4}	0.381	0.511	0.419	0.393	0.510	0.460
10^{-3}	0.381	0.512	0.420	0.391	0.515	0.462
10^{-2}	0.381	0.509	0.420	0.411	0.529	0.462
10^{-1}	0.388	0.501	0.419	0.524	0.741	0.609
1	0.389	0.525	0.433	0.955	0.995	0.995
10	0.386	0.554	0.437	0.995	0.995	0.995
10^2	1.000	1.000	1.000	0.995	0.995	0.995
PCA $m = 9$						
10^{-4}	0.370	0.509	0.408	0.385	0.525	0.453
10^{-3}	0.369	0.506	0.408	0.386	0.531	0.457
10^{-2}	0.368	0.505	0.410	0.405	0.546	0.458
10^{-1}	0.370	0.516	0.416	0.494	0.771	0.585
1	0.382	0.505	0.422	0.950	0.991	0.988
10	0.373	0.554	0.419	0.995	0.995	0.995
10^2	0.998	0.998	0.998	0.995	0.995	0.995

Table 2.3: Validation results for prior-weighted Quadratic LogReg classifiers

2.2.3 Support Vector Machines

Support Vector Machines (*SVMs*) are binary discriminative classifiers. The models are trained through the maximization of the objective function with respect to the model parameters α in the dual formulation:

$$L_D(\alpha) = \alpha^T \mathbf{1} - \frac{1}{2} \alpha^T \mathbf{H} \alpha, \quad \mathbf{H}_{ij} = z_i z_j k(\mathbf{x}_i, \mathbf{x}_j), \quad 0 \leq \alpha_i \leq C_i, i = 1, \dots, n$$

where $k(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function and the costs are rebalanced over the application prior π_T and the empirical prior π_T^{emp} :

$$C_i = \begin{cases} C \frac{\pi_T}{\pi_T^{emp}} & \text{if } C_i = \mathcal{H}_T \\ C \frac{\pi_F}{\pi_F^{emp}} & \text{if } C_i = \mathcal{H}_F \end{cases}$$

SVMs are trained using three kernel functions:

- **Dot-product (linear SVMs)** $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- **Polynomial kernel** $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d + \xi$
- **Radial Basis Function kernel (RBF)** $k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2} + \xi$

Both polynomial of degree d ($Poly(d)$) and RBF kernels are considered with the addition of a bias regularization term $\xi = K^2$, while in linear models the bias regularization term K is incorporated in the extended feature space:

$$\hat{\mathbf{D}} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ K & K & \dots & K \end{bmatrix}$$

For linear SVMs, tuning of hyper-parameters C and K is performed with a grid search approach both for raw data and data preprocessed with PCA $m = 9$. Values for C are searched in a log scale between 10^{-2} and 10^6 , while for K values 1 and 10 are considered. We expect linear SVMs to perform comparably to other linear models.

The same values of C and K are considered for Poly(2) SVMs, which are expected to perform better than the linear ones, just like LogReg models. Poly SVMs of degree $d = 3$ may overfit due to it being too complex, thus we try only the most promising values from previous results.

We omit results for $C = 0.01$ in Table 2.4 and we also do not report tabular results for linear SVMs with PCA, since they do not bring useful information.

2.2 – Classifiers

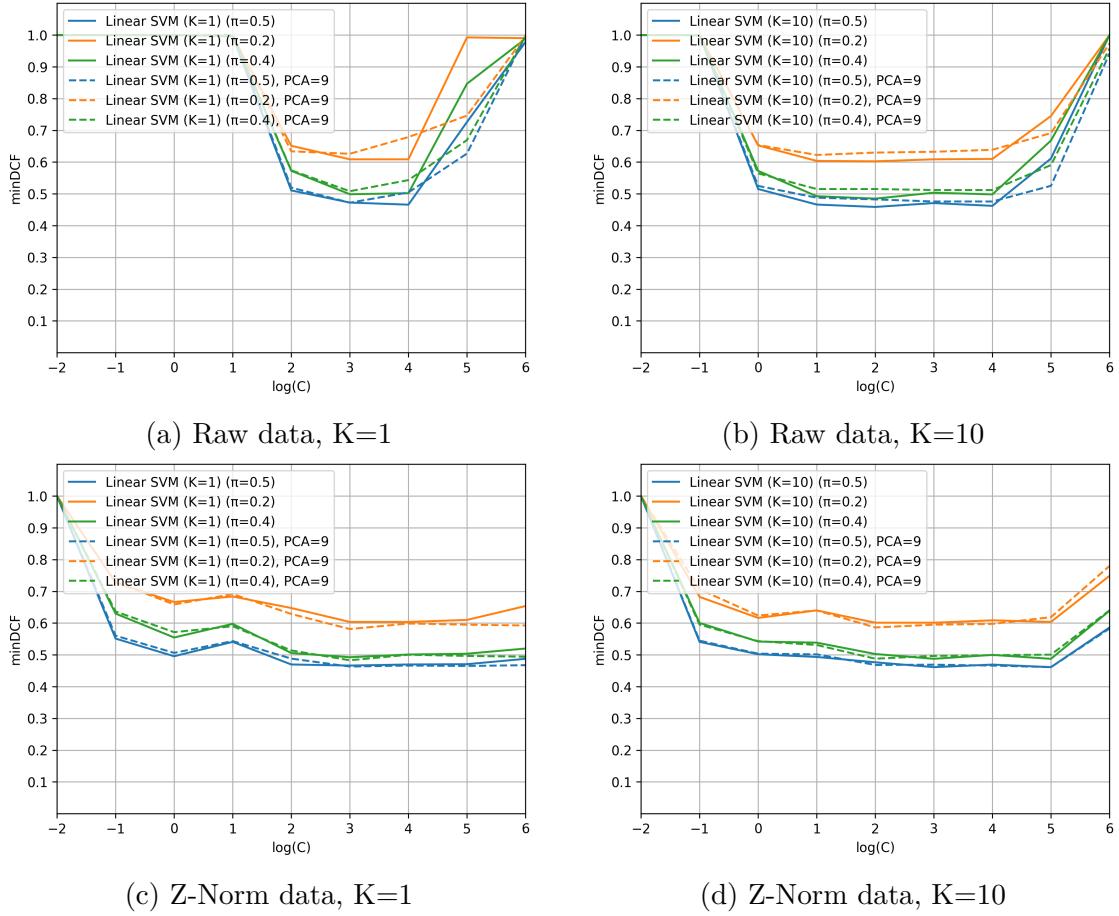


Figure 2.5: Hyper-parameter selection for Linear SVMs

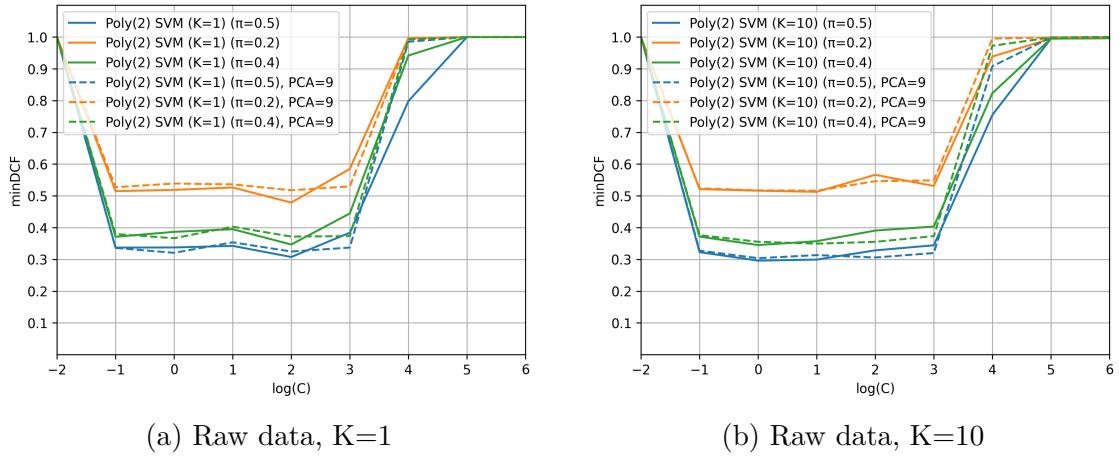


Figure 2.6: Hyper-parameter selection for Poly(2) SVMs

K	C	minDCF (RAW)			minDCF (Z-Norm)		
		$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
1	10^{-1}	1.000	1.000	1.000	0.551	0.727	0.630
1	1	1.000	1.000	1.000	0.496	0.666	0.555
1	10	0.996	0.996	0.996	0.541	0.684	0.597
1	10^2	0.511	0.651	0.573	0.470	0.647	0.506
1	10^3	0.472	0.609	0.499	0.466	0.604	0.492
1	10^4	0.466	0.609	0.502	0.470	0.604	0.501
1	10^5	0.726	0.993	0.847	0.471	0.610	0.504
1	10^6	0.978	0.990	0.989	0.487	0.654	0.520
10	10^{-1}	1.000	1.000	1.000	0.541	0.682	0.600
10	1	0.515	0.652	0.572	0.502	0.616	0.542
10	10	0.466	0.604	0.492	0.493	0.640	0.538
10	10^2	0.459	0.602	0.485	0.476	0.601	0.503
10	10^3	0.471	0.609	0.504	0.461	0.601	0.487
10	10^4	0.462	0.610	0.498	0.469	0.609	0.500
10	10^5	0.611	0.745	0.668	0.461	0.604	0.487
10	10^6	1.000	1.000	1.000	0.586	0.749	0.639

Table 2.4: Validation results for rebalanced linear SVMs (no PCA)

K	C	minDCF (RAW)			minDCF (PCA, $m = 9$)		
		$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
1	10^{-1}	0.337	0.515	0.371	0.336	0.527	0.380
1	1	0.337	0.519	0.387	0.321	0.539	0.366
1	10	0.342	0.526	0.395	0.354	0.536	0.403
1	10^2	0.307	0.479	0.347	0.325	0.517	0.372
1	10^3	0.384	0.585	0.444	0.337	0.530	0.373
1	10^4	0.798	0.996	0.942	0.985	0.993	0.993
10	10^{-1}	0.323	0.521	0.371	0.327	0.524	0.376
10	1	0.296	0.516	0.345	0.304	0.517	0.355
10	10	0.299	0.512	0.357	0.313	0.516	0.349
10	10^2	0.328	0.566	0.391	0.306	0.546	0.355
10	10^3	0.344	0.531	0.403	0.320	0.549	0.373
10	10^4	0.755	0.939	0.823	0.908	0.995	0.972

Table 2.5: Validation results for rebalanced Poly(2) SVMs

Taking into account results from Table 2.5, we train Poly(3) SVM models for values of C in range $10^{-1} – 10$ without PCA. As expected, by observing Figure 2.7, Poly(3) SVMs are not suitable for the task.

For RBF kernel SVMs we perform a grid search on γ and C , keeping $K = 10$. Results for $\pi = 0.5$ are reported in Table 2.6 and Figure 2.8, omitting the least promising results.

γ	C	minDCF (RAW)		
		$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
0.001	10^{-1}	0.943	0.946	0.946
0.001	1	0.555	0.687	0.609
0.001	10	0.363	0.505	0.423
0.001	10^2	0.516	0.609	0.568
0.001	10^3	0.299	0.456	0.356
0.001	10^4	0.305	0.466	0.359
0.001	10^5	0.301	0.470	0.370
0.002	10^{-1}	0.657	0.769	0.682
0.002	1	0.335	0.494	0.402
0.002	10	0.312	0.466	0.385
0.002	10^2	0.305	0.481	0.374
0.002	10^3	0.299	0.451	0.342
0.002	10^4	0.271	0.436	0.324
0.002	10^5	0.342	0.487	0.393
0.005	10^{-1}	0.340	0.531	0.386
0.005	1	0.310	0.496	0.385
0.005	10	0.313	0.504	0.394
0.005	10^2	0.663	0.722	0.673
0.005	10^3	0.293	0.464	0.351
0.005	10^4	0.289	0.476	0.358
0.005	10^5	0.385	0.611	0.454
0.01	1	0.330	0.532	0.401
0.01	10^3	0.316	0.555	0.391
0.02	10^3	0.432	0.659	0.491
0.05	10^3	0.538	0.814	0.669
0.1	1	0.729	0.945	0.788

Table 2.6: Validation results for rebalanced RBF SVMs

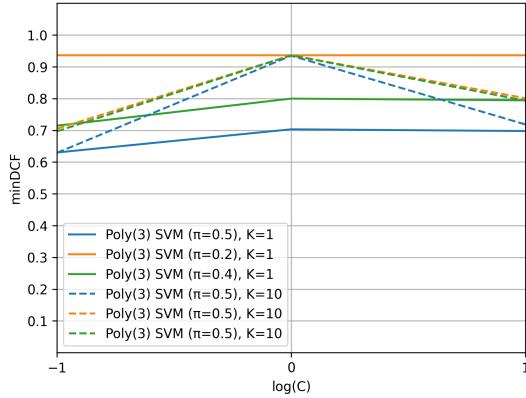


Figure 2.7: Hyper-parameter selection for Poly(3) SVMs

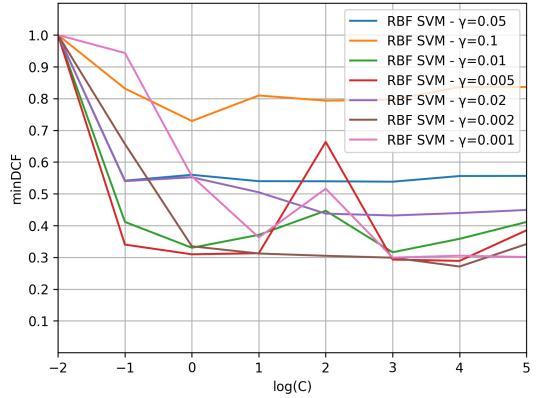


Figure 2.8: Hyper-parameter selection for RBF SVMs with $\pi = 0.5$, no PCA

Overall, RBF SVMs provide good results, having the best performing model up to now with $\gamma = 0.002$, $C = 10^4$ and $K = 10$. PCA does not improve results significantly. Here are the selected SVM models:

Model	minDCF (RAW)		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
Poly(2) SVM ($C = 1$, $K = 10$)	0.296	0.516	0.345
RBF SVM ($\gamma = 0.002$, $C = 10^4$, $K = 10$)	0.271	0.436	0.324

2.2.4 Gaussian Mixture Model classifiers

Gaussian Mixture Models (*GMMs*) are generative models which assume that samples are modeled by a weighted combination of K Gaussian densities (*components* or *clusters*):

$$f_X(\mathbf{x}) = \sum_{c=1}^K w_c \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

GMM binary classifiers estimate, for each class c , a K_c components GMM with parameters $(\mathbf{M}_c, \mathbf{S}_c, \mathbf{w}_c)$ in order to compute the log-likelihood ratios. We choose to separately model the number of components for target class and non-target class, allowing the GMMs to account the diversity of per-class distributions. GMMs are also expected to be the best choice to describe the sub-groups of the spoofed class. The hyper-parameters to be tuned are the number of components for target class K_T and the number of components for non-target class K_{NT} , with fewer components for target class.

2.2 – Classifiers

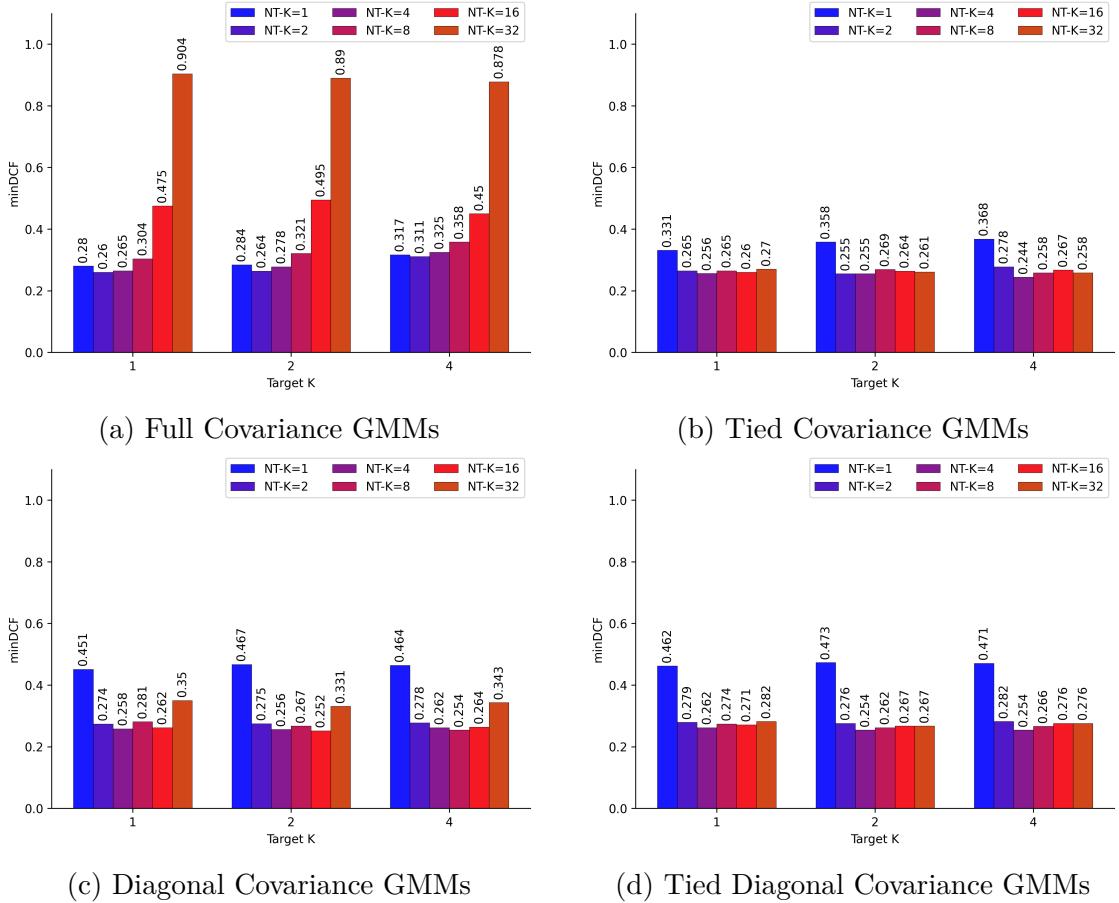


Figure 2.9: Hyper-parameter selection for GMMs ($\pi = 0.5$)

In Figure 2.9 validation performances on Full Covariance, Tied Covariance, Diagonal Covariance and Tied Diagonal Covariance GMMs are reported. The best models for each GMM can be found in Table 2.7. Validation results for PCA with $m = 9$ are not reported since it does not contribute significantly.

As expected, all the GMMs provide the best results so far. Diagonal GMMs show an improvement in performance with respect to the Full Covariance ones due to the low feature correlation between classes, while the promising results of both the Tied models may depend on the similar distributions among clusters.

Although the lowest $minDCF$ for $\pi = 0.5$ is obtained with the Diagonal Covariance with $K_T = 2$ and $K_{NT} = 16$, other working points get better results with $K_{NT} = 4$ or $K_{NT} = 8$, which is closer to the number of subclasses in the non-target class samples. We thus choose the Tied Diagonal Covariance GMM with $K_T = 4$ and $K_{NT} = 4$.

Model	K_T	K_{NT}	minDCF		
			$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
Full Covariance	1	2	0.260	0.432	0.321
Tied Covariance	2	4	0.255	0.426	0.297
Tied Covariance	4	4	0.244	0.431	0.300
Tied Covariance	4	8	0.258	0.437	0.295
Diagonal Covariance	2	4	0.256	0.450	0.296
Diagonal Covariance	2	16	0.252	0.484	0.301
Diagonal Covariance	4	8	0.254	0.407	0.300
Tied Diagonal Covariance	2	4	0.254	0.454	0.306
Tied Diagonal Covariance	4	4	0.254	0.430	0.293

Table 2.7: Validation results for GMM models (raw data)

2.3 Most promising models

Here are the most promising models for each type of classifier.

Model	minDCF		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
Full Covariance MVG	0.330	0.522	0.392
Q-LogReg ($\lambda = 10^{-2}$, PCA $m = 9$)	0.368	0.505	0.410
Poly(2) SVM ($C = 1$, $K = 10$)	0.296	0.516	0.345
RBF SVM ($\gamma = 0.002$, $C = 10^4$, $K = 10$)	0.271	0.436	0.324
Tied Diagonal GMM ($K_T = 4$, $K_{NT} = 4$)	0.254	0.430	0.293

Table 2.8: Validation in terms of $minDCF$ for the five most promising models

2.4 Score calibration

Until now model validation has been performed on the minimum DCF, whereas the metric used for the evaluation phase will be the actual DCF, which is the cost paid with the effective threshold $t = -\log \frac{\tilde{\pi}}{1-\tilde{\pi}}$. The effective threshold and the theoretical one are the same with well calibrated scores, while they differ when model assumptions are not accurate or the train and test set differ significantly or scores have no probabilistic interpretation (like SVMs).

In order to recover calibrated scores we use a prior-weighted Logistic Regression, assuming the calibration function $s_{cal} = f(s)$ is linear. Since Logistic Regression is a discriminative model, in order to recover the log-likelihood ratio from the log-posterior ratio we subtract the prior-dependent term $\log \frac{\tilde{\pi}}{1-\tilde{\pi}}$.

$$f(s) = \log \frac{f_{S|C}(s | \mathcal{H}_T)}{f_{S|C}(s | \mathcal{H}_F)} = \alpha s + \gamma = \alpha s + \beta - \log \frac{\tilde{\pi}}{1-\tilde{\pi}}$$

The prior-weighted LogReg will learn the parameters α and β . Since the feature space is one-dimensional there is no need to add the regularization term ($\lambda = 0$). Results are validated using a K-fold approach with $K = 5$ to improve robustness.

2.4.1 Uncalibrated models

Here are Bayes error plots and Receiver Operating Characteristic (*ROC*) curves for each of the chosen classifiers.

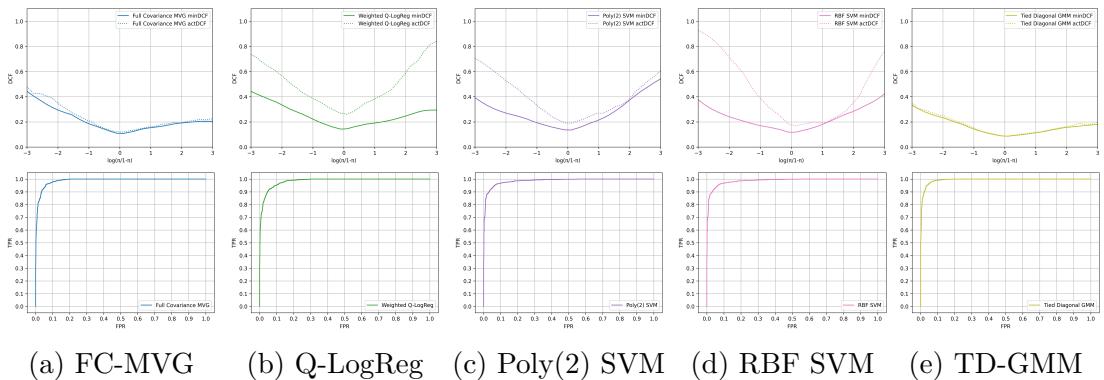


Figure 2.10: Bayes Error plot and ROC curves for each of the chosen models

As expected, generative models are well calibrated, while Quadratic Logistic Regression and SVMs are not: the mis-calibration of the Q-LogReg may depend on a variation of the empirical prior of the validation set, while SVMs scores are subject to mis-calibration due to their non-probabilistic interpretation.

2.4.2 Calibrated models

Results of score calibration for the three mis-calibrated models is reported in Figure 2.11, where they are compared to the uncalibrated results.

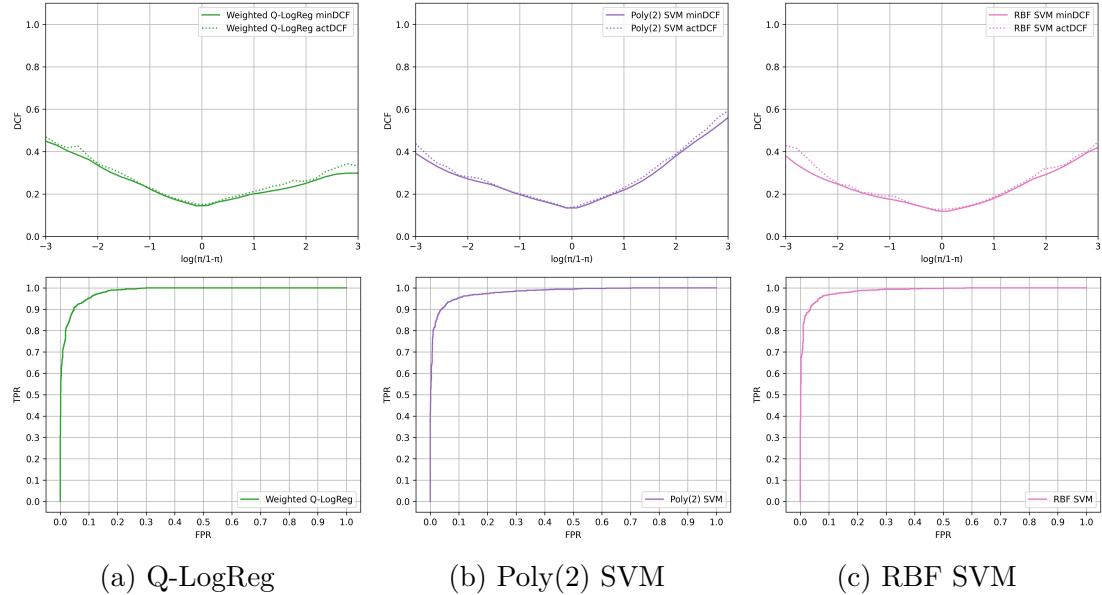


Figure 2.11: Bayes Error plot and ROC curves for calibrated versions of the selected mis-calibrated models

Model	actDCF (Uncalibrated)			actDCF (Calibrated)		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
FC-MVG	0.418	0.625	0.421	-	-	-
Q-LogReg	0.617	0.855	0.695	0.418	0.526	0.426
Poly(2) SVM	0.583	0.786	0.662	0.311	0.546	0.373
RBF SVM	0.800	0.993	0.879	0.302	0.482	0.392
TD-GMM	0.273	0.434	0.296	-	-	-

When comparing actual normalized DCFs, Tied Diagonal GMM is still the best model, whereas RBF and Poly(2) SVMs have been improved considerably. Multivariate Gaussian Classifier and calibrated Poly(2) SVM have the same performance for $\pi = 0.5$ and similar performances for $\pi = 0.4$, with the latter being much better for applications with $\pi = 0.2$.

We now try models fusion with the best three models: Tied Diagonal GMM, RBF SVM and Full Covariance MVG.

2.5 Model fusion

The same process for score calibration can be used for combining multiple models: the transformation becomes a linear function of the scores obtained by different classifiers. Fusion is performed by training the same prior-weighted Logistic Regression model in which each feature corresponds to a different model. Since the feature space is still low-dimensional, we use the same LogReg used for calibration. Results compared with base models are reported in Table 2.9 and in the Detection Error Trade-off curve in Figure 2.12.

Model	minDCF			actDCF		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
FC-MVG	0.330	0.522	0.392	0.418	0.625	0.421
RBF SVM	0.271	0.436	0.324	0.302	0.482	0.392
TD-GMM	0.254	0.430	0.293	0.273	0.434	0.296
MVG + RBF	0.299	0.515	0.358	0.317	0.567	0.390
MVG + GMM	0.260	0.465	0.314	0.281	0.504	0.318
RBF + GMM	0.264	0.461	0.309	0.272	0.475	0.311
MVG + RBF + GMM	0.263	0.470	0.314	0.276	0.481	0.318

Table 2.9: Minimum and actual DCF results for all the most promising models

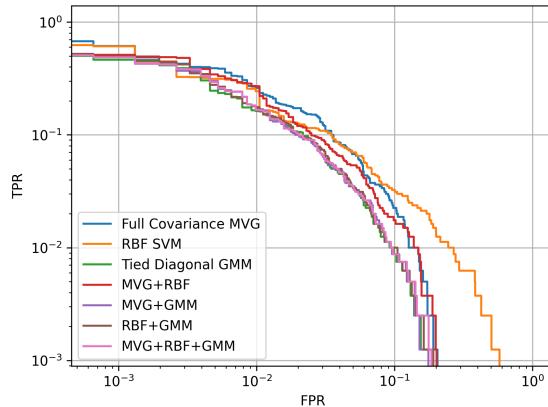


Figure 2.12: Comparison among DET curves for all the considered models

Despite all model fusions provide good results, with the best one being the combination RBF+GMM according to actDCF, the DET curves allow us determine that the overall best performing model is the Tied Diagonal GMM, which will be our final choice.

Chapter 3

Evaluation

We now evaluate the performance of the selected models on the evaluation set in terms of minimum DCF for the three applications we examined until now. As mentioned in Section 1.2, the evaluation set consists in 7704 samples. The empirical prior is different from the training set to a small extent (there are 2400 authentic samples and 5304 spoofed ones, $\pi_{eval} = 0.69$), so we expect models to slightly underperform validation results. We only calibrate the models we selected beforehand.

3.1 Multivariate Gaussian classifiers

We now consider results for Multivariate Gaussian classifiers.

Model	PCA	minDCF		
		$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
Full Covariance MVG	-	0.276	0.445	0.325
Full Covariance MVG	9	0.272	0.454	0.323
Full Covariance MVG	8	0.271	0.452	0.319
Full Covariance MVG	7	0.273	0.448	0.324
Tied Covariance MVG	-	0.455	0.652	0.521
Tied Covariance MVG	9	0.461	0.646	0.517
Tied Covariance MVG	8	0.456	0.643	0.519
Tied Covariance MVG	7	0.459	0.648	0.518
Naïve Bayes MVG	-	0.352	0.587	0.410
Naïve Bayes MVG	10	0.320	0.518	0.361
Naïve Bayes MVG	9	0.310	0.524	0.365
Naïve Bayes MVG	8	0.315	0.518	0.368
Naïve Bayes MVG	7	0.313	0.518	0.363
Tied Naïve Bayes MVG	-	0.480	0.721	0.545
Tied Naïve Bayes MVG	10	0.488	0.738	0.555
Tied Naïve Bayes MVG	9	0.483	0.739	0.556
Tied Naïve Bayes MVG	8	0.484	0.742	0.558
Tied Naïve Bayes MVG	7	0.482	0.742	0.556

Table 3.1: Evaluation results for Multivariate Gaussian Classifiers both with and without PCA

Results are consistent with validation. Moreover, evaluation performance is even better: this might mean either that the validation set does not reflect well the training set population, or the evaluation set distribution is much more favorable for the models we trained, making them robust.

3.2 Logistic regression classifiers

Here are evaluation and validation minDCF plots for different values of λ of prior-weighted Linear Logistic Regression models. We evaluate only the model trained on unprocessed raw data and the model with PCA on Z-score normalized data, which was the best performing one among the linear LogReg classifiers.

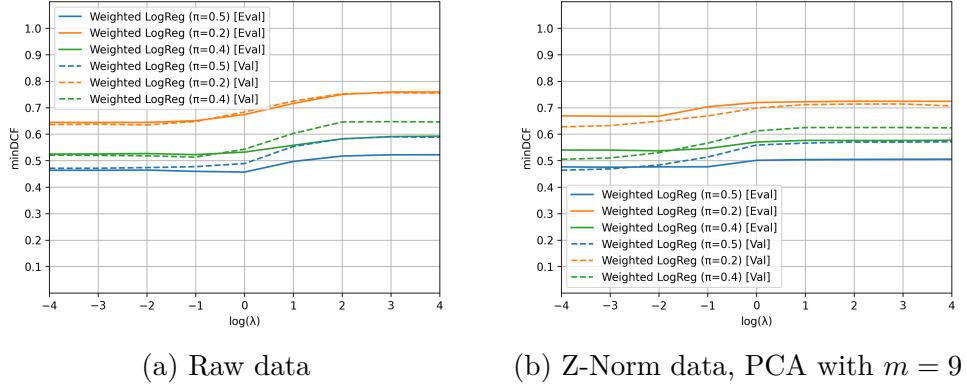


Figure 3.1: Evaluation results for prior-weighted linear LogReg models

Evaluation results of Linear LogReg models are consistent as well, with preprocessing leading to a little improvement as in validation phase. The chosen value of $\lambda = 10^{-4}$ during validation is confirmed suitable for both the raw features and the Z-normalized models.

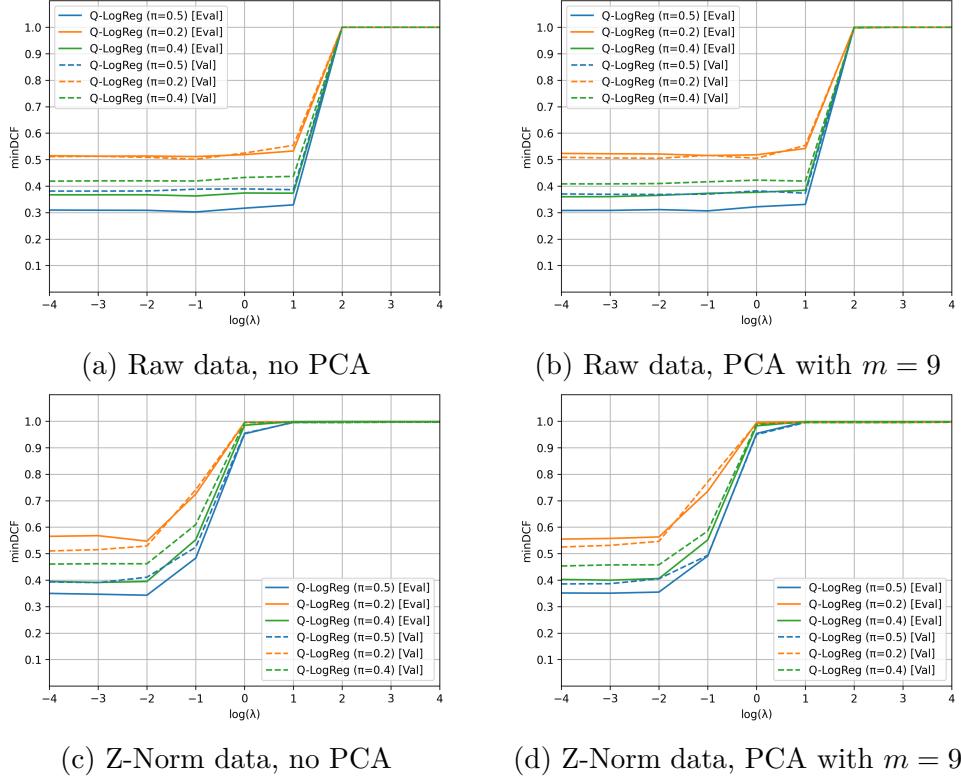


Figure 3.2: Evaluation for Q-LogRegs (without score calibration)

Quadratic Logistic regression had better validation results than its linear counterpart. In Figure 3.2 are the minDCF plots for all the four combination that were discussed earlier. Again, both PCA and Z-Score normalization have little effect on performance for smaller values λ , in accordance to previous results. The selected value $\lambda = 10^{-2}$ is confirmed to be a good choice.

3.3 Support Vector Machines

Evaluation on Support Vector Machines will be performed limiting the number of variable hyper-parameters. For tractability reasons, we consider only models trained on the raw dataset with $K = 10$ and we fix $\gamma = 0.002$ for RBF SVMs. In Table 3.2 are the evaluation results for the most performing models in evaluation phase for each SVM kernel, followed by plots of evaluation minDCF on different values of C .

Model	minDCF		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
Linear SVM (RAW, $K = 10, C = 10^2$)	0.472	0.678	0.522
Poly(2) SVM (RAW, $K = 10, C = 1$)	0.297	0.439	0.334
RBF SVM (RAW, $\gamma = 0.002, C = 10^4, K = 10$)	0.259	0.401	0.290

Table 3.2: Evaluation results for the best SVM model from validation for each kernel

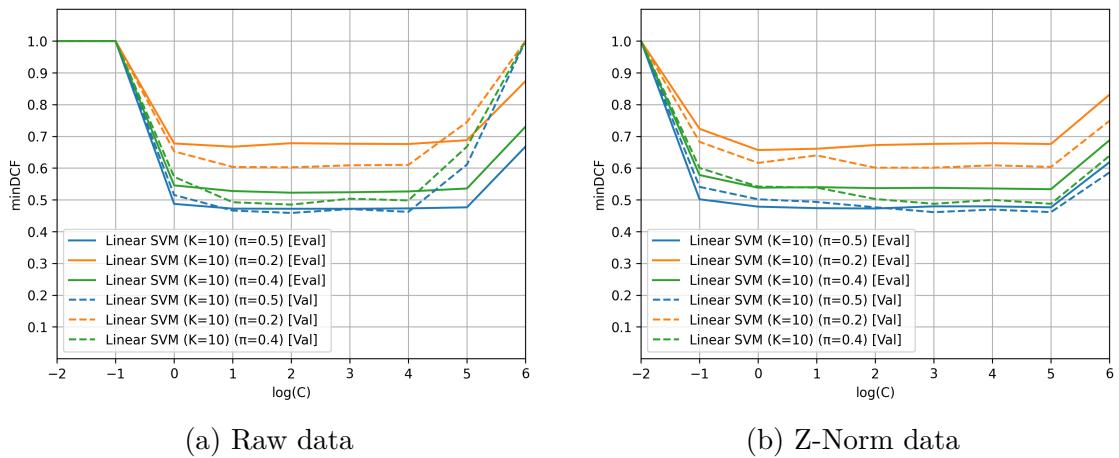


Figure 3.3: Evaluation for rebalanced linear SVMs

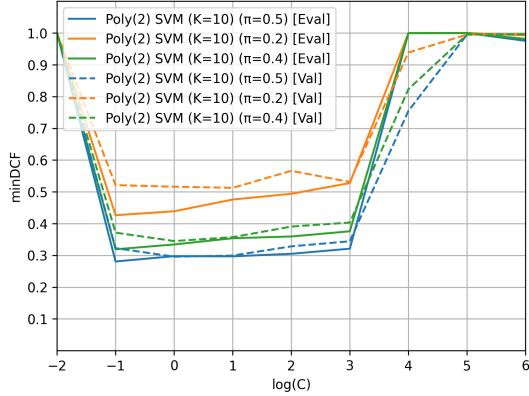


Figure 3.4: Evaluation for rebalanced Poly(2) SVMs

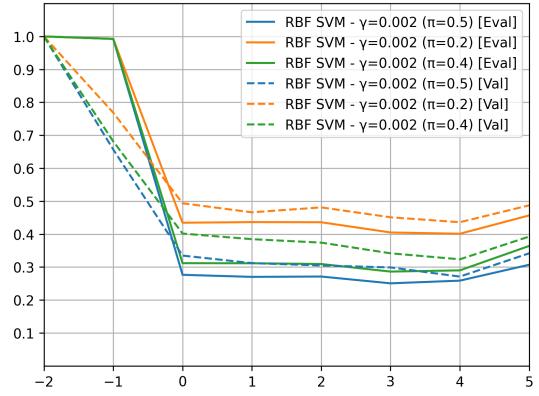


Figure 3.5: Evaluation for rebalanced RBF kernel SVMs

The best SVM models are the RBF kernel ones again, but the choice of $C = 10^4$ is not optimal, since the best result is obtained with $C = 10^3$. For Polynomial kernel SVMs with degree 2 the choice of $C = 1$ is good for all the considered applications, but the optimal performance is obtained with $C = 0.1$.

3.4 Gaussian Mixture Model classifiers

We report all the evaluation results for GMMs in Figure 3.6, while in Table 3.3 we report only the evaluation results of the most promising model during validation for each GMM type.

Model	minDCF		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
FC-GMM ($K_T = 1, K_{NT} = 2$)	0.229	0.387	0.278
T-GMM ($K_T = 4, K_{NT} = 4$)	0.217	0.378	0.256
D-GMM ($K_T = 2, K_{NT} = 16$)	0.231	0.399	0.281
TD-GMM ($K_T = 4, K_{NT} = 4$)	0.223	0.406	0.267

Table 3.3: Evaluation results for GMMs

3.5 – Model fusion

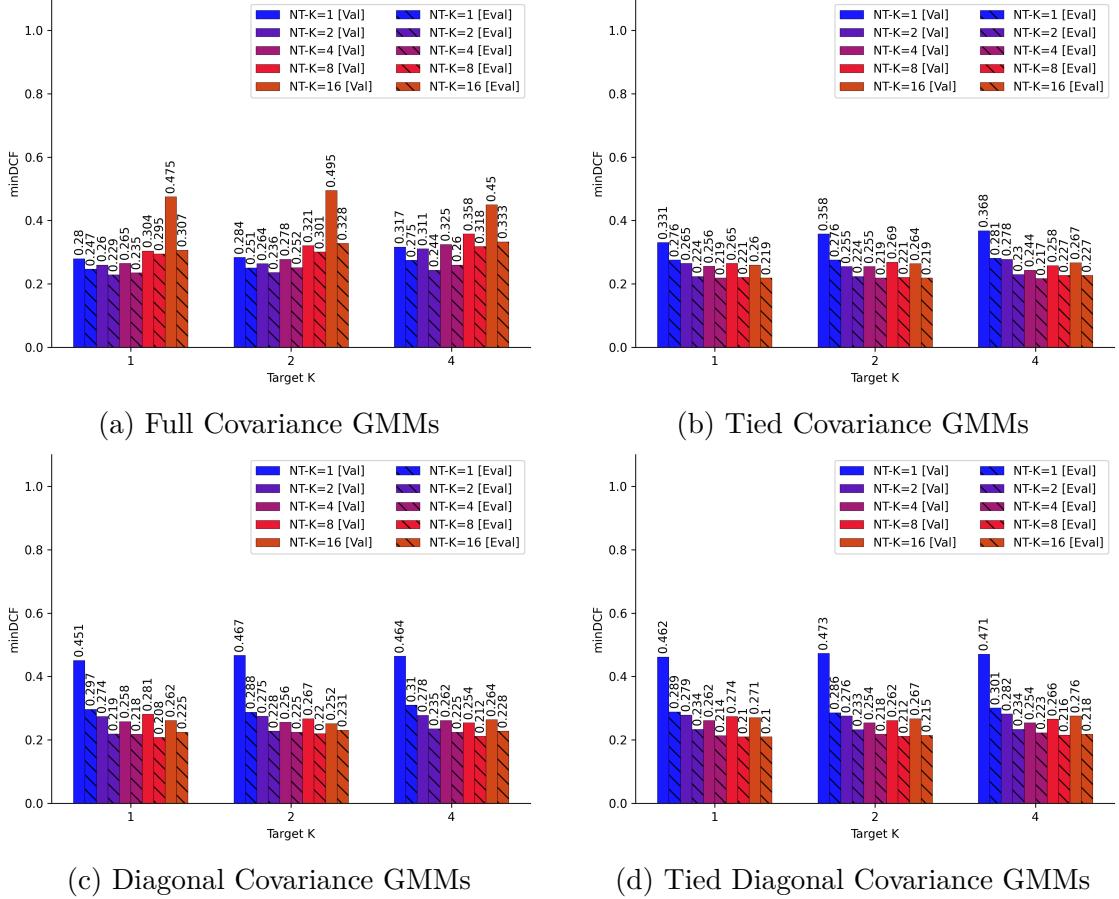


Figure 3.6: Evaluation for GMMs ($\pi = 0.5$)

GMMs are confirmed to be the best performing models. However, our choice was suboptimal, since Diagonal Covariance GMMs outperformed the Tied Diagonal models also in the evaluation: the highest $\text{minDCF} = 0.208$ is obtained with a Diagonal Covariance GMM with $K_T = 1$ and $K_{NT} = 8$.

3.5 Model fusion

Here are reported the evaluation results for the best three performing models during validation with score calibration and model fusion (Table 2.9) both in terms of minDCF .

Model	minDCF		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
MVG + RBF	0.305	0.485	0.357
MVG + GMM	0.269	0.454	0.325
RBF + GMM	0.240	0.427	0.284
MVG + RBF + GMM	0.277	0.445	0.332

Table 3.4: Minimum DCF results for model fusions (evaluation)

Model fusions are confirmed not to improve results. Evaluation confirms that the optimal fusion is *RBF + GMM*.

3.6 Actual best model

We report minimum and actual DCF costs and plots for calibration with respect to the optimal model (Tied Covariance GMM with $K_T = K_{NT} = 4$).

From Figure 3.7 we can state that calibration is not required for the model, since its score already have a probabilistic interpretation. Here in Table 3.5 are the costs for the optimal solution.

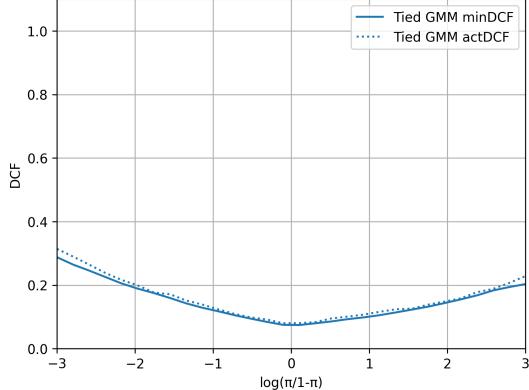


Figure 3.7: Bayes error plot for uncalibrated optimal model

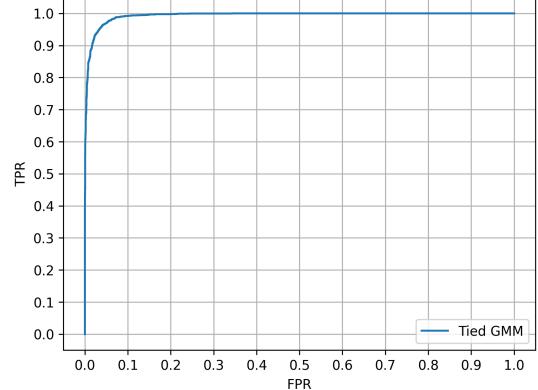


Figure 3.8: Bayes error plot for calibrated optimal model

Model	minDCF			actDCF		
	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$	$\pi = 0.5$	$\pi = 0.2$	$\pi = 0.4$
TC-GMM	0.225	0.393	0.280	0.251	0.420	0.271

Table 3.5: Minimum and actual DCF results for the optimal solution

Chapter 4

Conclusions

The performed analysis provided good results, especially with Gaussian Mixture Models classifiers. Our choice to select a Tied Diagonal Covariance GMM classifier over a better performing Diagonal Covariance GMM one in order to avoid overfitting was not effective, however it still provided good results for the chosen application $(\pi_T, C_{fn}, C_{fp}) = (0.5, 1, 10)$.

Two points must be addressed:

- The DCF costs obtained on the evaluation set are lower than the ones obtained on the validation set: this can be most likely ascribable to an imbalance in the validation set (the empirical prior of the validation set does not reflect the empirical prior of the training set).
- Both the other two working points were chosen with lower empirical prior than the main one in order to address two tasks with higher chances of fingerprints being spoofed. Selecting a task with an higher C_{fp} could have been a valuable alternative, in order to address a task in which an incorrect classification of a spoofed fingerprint would have a larger impact (e.g. a bank vault would require a much higher cost for false positives than a smartphone).