Fingerprint spoofing - Project Report

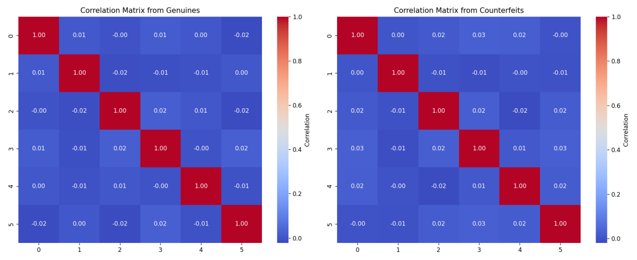
Fabio Barbieri
Politecnico di Torino
s317783@studenti.polito.it

Abstract

This report describes the process of finding a model able to identify genuine vs counterfeit fingerprint images. The data set consists of labeled samples corresponding to the genuine (true, label 1) class and the fake (false, label 0) class. The samples are computed by a feature extractor that summarizes the high-level characteristics of a fingerprint image in a 6-dimensional feature vector. The goal is to try different shallow models and evaluate them using Bayes evaluation. The best model will then be calibrated using score calibration and delivered as a final system after it's evaluation on an held-out set. A preliminary dataset analysis that aims to understand how features are distributed can be found in Section 1. In Section 2 an analysis of running dimensionality reduction on the dataset in the form of PCA and LDA is reported; alongside a first result applying LDA classification. A further analysis on learning MVG models on the dataset is reported in Section 3; while Sections 4, 5, 6, report the process of learning a Logistic Regression Model, a Support Vector Machine and Gaussian Mixtures, respectively. In Section 7 we compare all the learned models on our main application, which consist in assuming that most of the users will be impostors (or equivalently, that the cost of granting access to an impostor has a higher cost than labeling as impostor a legit user - we aim for strong security). The implementation can be found at: https://github.com/Gardusio/Fingerprint-Spoofing.

1. Dataset Analysis

We begin by examining the distributions of the features in our dataset, as depicted in Figure 2. This preliminary analysis provides insight into the characteristics of genuine and counterfeit fingerprint samples. The dataset is balanced (2090 Counterfeits and 3010 Genuines) and Pearson correlation for the two classes shows that there's no redundancy and each feature contributes almost independently to the classification task. For feature 1 and 2, we can see how both approximate Gaussian distributions, each displaying a single mode near their respective means. Feature 1 ex-

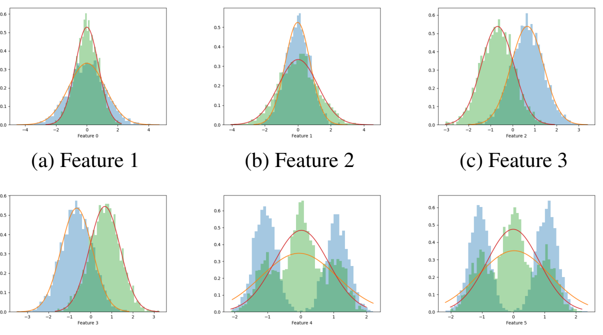


	0	1	2	3	4	5
0	1.00	0.01	-0.00	0.01	0.00	-0.02
1	0.01	1.00	-0.02	-0.01	-0.01	0.00
2	-0.00	-0.02	1.00	0.02	0.01	-0.02
3	0.01	-0.01	0.02	1.00	-0.00	0.02
4	0.00	-0.01	0.01	-0.00	1.00	-0.01
5	-0.02	0.00	-0.02	0.02	-0.01	1.00

	0	1	2	3	4	5
0	1.00	0.00	0.02	0.03	0.02	-0.00
1	0.00	1.00	-0.01	-0.01	-0.00	-0.01
2	0.02	-0.01	1.00	0.02	-0.02	0.02
3	0.03	-0.01	0.02	1.00	0.01	0.03
4	0.02	-0.00	-0.02	0.01	1.00	0.02
5	-0.00	-0.01	0.02	0.03	0.02	1.00

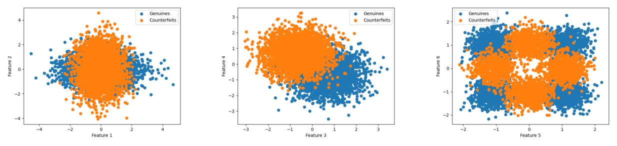
Figure 1. Features correlation matrices

hibits a significant overlap around zero, with genuine samples (mean: 0.0005, variance: 1.43) showing greater variance compared to counterfeit samples (mean: 0.003, variance: 0.57). This overlap indicates that Feature 1 alone is insufficient for clear class separation. Feature 2 mirrors the pattern of Feature 1, with a large overlap around zero. However, the variances are inverted: genuine samples (mean: -0.0085, variance: 0.58) exhibit less variance compared to counterfeit samples (mean: 0.018, variance: 1.42). Like Feature 1, Feature 2 alone does not provide a strong decision boundary. Feature 3 and 4 also show Gaussian-like distributions, but with significantly different statistics between the two classes. Feature 3 shows some overlap around zero; however, the distributions are more separated than features 1 and 2. Genuine samples span [-2, 3] with a mean of 0.66 and variance of 0.54, while counterfeit samples span [-3, 2) with a mean of -0.68 and variance of 0.55. Feature 4 demonstrates a similar pattern in reverse, with genuine samples spanning [-3, 2) (mean: -0.66, variance: 0.55) and counterfeit samples spanning (-2, 3] (mean: 0.67, variance: 0.53). The distinct means in these features suggest they can better facilitate class separation compared to Features 1 and 2. Feature 5 reveals a bimodal distribution for genuine samples, with modes around -1 and 1. Counterfeit samples display a trimodal distribution, peaking around -1, 0, and 1. Overlaps occur primarily in the ranges [-2, 0] and [0, 2], with few genuine samples around zero. Feature 6 shows a similar pattern to Feature 5, indicating that genuine samples are rare around zero for both features. The scatter plots in Figure 3c exhibit clear clustering, which suggests potential for effective classification when these features are considered jointly. Samples near zero for both features indicate counterfeits, while those around -1 and 1 can be distinguished by the value of the other feature.


(a) Feature 1 (b) Feature 2 (c) Feature 3

(d) Feature 4 (e) Feature 5 (f) Feature 6

Figure 2. Features histograms with Gaussian fits



(a) Features 1-2 (b) Features 3-4 (c) Feature 5-6

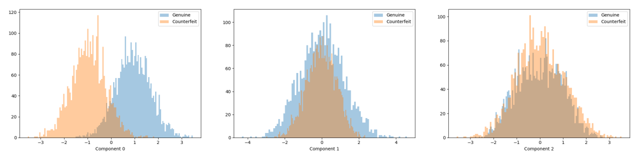
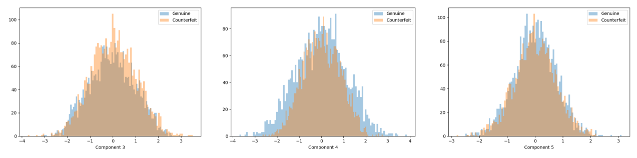
Figure 3. Features scatter plots

2. Dimensionality reduction

In this Section we'll apply dimensionality reduction to the dataset. First, we analyze the effect of applying PCA, then we will resort to LDA and show a first result using LDA classification. Applying PCA (Figure 4) clearly shows that only the first component provides good separation, thus reducing to only the Main component (Figure 4a) might retain most of the discriminatory power. The fact that the classes overlap significantly considering the other components suggests that they capture less of the discriminatory variance between the classes. They may still contribute to the overall model by capturing more nuanced variations, since we are in a setting with zero to few correlation.

2.1. Linear Discriminant Analysis

The result of applying LDA to the unmodified dataset is shown in Figure 5, we can see that the two classes are fairly linearly separable. The separability is similar as in the first PCA component. Applying LDA classification we get the results shown in Table 1, in which each entry represent an experiment of running PCA and retaining a specific number of component before applying LDA. To perform LDA classification, we've splitted the dataset in training and test

(a) Main component (b) Second component (c) Third component

(d) Fourth component (e) Fifth component (f) Sixth component

Figure 4. Principal components

with a ratio of 2 to 1. Two type of thresholds have been used: distance between projected means and an empirical threshold that minimizes the error rate for the given experiment. The best performing setting is with retaining the first 2 principal components when applying the mean distance threshold and just the first when minimizing the error rate selecting the threshold. Overall, a simple mean distance threshold provides a good separation boundary and applying PCA can improve accuracy but not with appreciable results, as we have seen that the dataset is already linearly separable in it's original feature space.

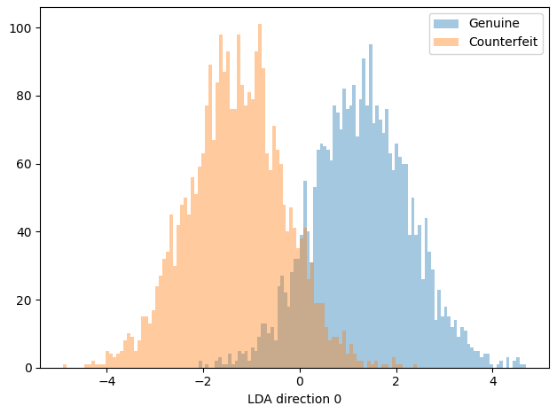


Figure 5. LDA separability

3. Multivariate Gaussian Classification

In this section we analyze the result of learning Multivariate Gaussian models (MVG) for the spoofing dataset. We'll use the same train and test splits used for LDA classification in Section 2.1. As shown in Figure 2, features 1 to 4 show Gaussian-like distributions, with different degrees of overlap. These characteristics suggest that MVG models may perform well on these features, while for features 5 and 6 fitting a Gaussian does not capture the nature of the distri-

PCA Dimensions	Threshold	Error Rate (%)
Classification with Best Threshold Selection		
0	-0.106	9.05
1	0.0140	**8.85**
2	0.0040	8.95
3	-0.0560	9.15
4	-0.0620	9.15
5	-0.0950	9.05
Classification with Mean Dist. Threshold		
0	-0.0185	9.3
1	-0.0176	9.35
2	0.0183	**8.95**
3	-0.0184	9.25
4	-0.0183	9.25
5	-0.0185	9.3

Table 1. LDA Classification results

PCA	MVG (%)	NB (%)	TIED (%)
Original	**7.00**	7.20	9.30
1	9.25	9.25	9.35
2	8.80	8.85	9.25
3	8.80	9.00	9.25
4	8.05	8.85	9.25
5	7.10	8.75	9.30
6	7.00	8.90	9.30

Table 2. Error Rates of MVG Models and Variants

Features	MVG (%)	NB(%)	TIED(%)
1,2,3,4	7.95	**7.65**	9.50
1,2	36.50	36.30	49.45
3,4	9.45	9.45	9.40

Table 3. MVGs error Rates after Feature Selection

bution. Despite the inaccuracy of the Gaussian distribution assumption for these two features, the models are still able to extract discriminant information from those, and no feature selection strategy managed to improve classification. Table 2 shows the performances of a standard MVG, a Tied MVG (TIED) and a Naive Bayes MVG (NB) with and without PCA, while Table 3 shows the result after applying feature selection. The standard MVG without PCA performs best overall, followed by the NB. The performances of the NB classifier are similar to the standard MVG in almost all settings. This is supported by the fact that there's little to no correlation between features as shown in Figure 1, and hence the Naive assumption holds fairly well. Dropping the last two features do not improve overall performances; but as expected, in this setting the NB classifier perform best since it's assumptions holds for all the (selected) features (de-correlation and Gaussian distribution). Applying MVGs using only features 1-2 (jointly) proved detrimental. As the scatters in Figure 3a shows, there is a significant overlap in the space composed by only features 1 and 2, and moreover, the class covariances are clearly different ("orthogonal"). Thus, the Tied model perform worst due to it's strict covariance assumptions. When dropping all features but the 3 and 4 instead, we obtain better results, as the samples overlap less in this space and class covariances are more similar; but still performances are not surpassing the standard models on all the features jointly.

3.1. MVGs Evaluation

We now analyze the performance of MVGs using our main application with effective prior $\tilde{\pi} = 0.1$, that encodes an higher cost of misclassification of spoofed fingerprints, aiming at more secure recognition. As shown in fig 4, since the empirical prior is different, we obtain a decrease of accuracy and calibration, but we earn a lower amount of false positives (i.e: a more secure system).

PCA	MVG		MVG Tied		MVG Naive	
	actDCF	minDCF	actDCF	minDCF	actDCF	minDCF
$m = 6$	0.305	0.263	0.406	0.363	0.302	0.257
$m = 5$	0.304	0.274	0.405	0.365	0.393	0.354
$m = 4$	0.353	0.301	0.403	0.361	0.397	0.361
$m = 3$	0.388	0.356	0.408	0.368	0.395	0.365
$m = 2$	0.388	0.353	0.396	0.363	0.387	0.356
$m = 1$	0.397	0.369	0.402	0.369	0.397	0.369

Table 4. Model evaluation with effective prior $\tilde{\pi} = 0.1$.

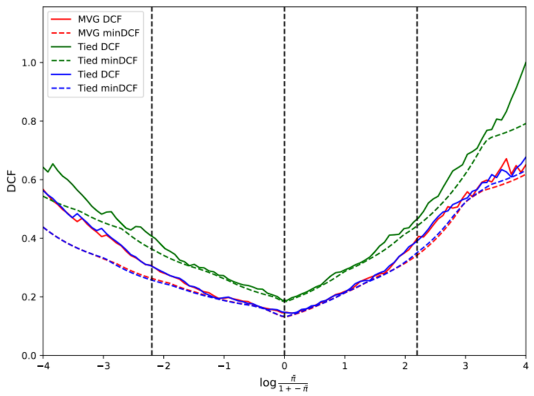


Figure 6. Bayes error plot of MVGs (6 principal components)

To gain better insights on the relative performances of the different flavors of MVG over different applications, we can refer to 6, which shows the Bayes error plot of mvg over all possible applications considering a prior log odds in the range of [-4, 4]. Our main application lies around -2, we see again that models tends to get miscalibrated and less precise for this application, but overall, models are consistently well calibrated for a large space of applications.

4. Logistic Regression

In this section we analyze the result of applying different flavors of logistic regression on the dataset.

4.1. Standard LogReg

λ	**accuracy**	**actDCF**	**minDCF**
10^{-4}	**90.70%**	**0.402**	0.364
10^{-3}	**90.65%**	**0.413**	0.365
10^{-2}	**90.75%**	**0.457**	**0.361**
10^{-1}	**90.75%**	0.852	0.364
10^{0}	**90.75%**	1.000	0.364
10^{1}	90.50%	1.000	**0.363**
10^{2}	87.15%	1.000	0.362
10^{3}	50.40%	1.000	0.362

Table 5. LogReg performances over λ for application $\tilde{\pi} = 0.1$

Results in Table 5 reflect a tension between the dataset's underlying linear separability and the strong effect of regularization under our stringent prior. As seen from Features 3, 4, and the combined behavior of 5 and 6, the dataset is largely linearly separable. With small λ (e.g., 10^{-4} or 10^{-2}), logistic regression can exploit those strongly discriminative features. Because these features already provide good class separation, the model does not need much shrinkage to avoid overfitting. Once λ becomes large (10^{-1} and beyond), the model underfits. Under heavy regularization, logistic regression coefficients shrink too much to capture the separation between genuine and counterfeit fingerprints despite the fact that the data itself has good linear discriminative structure. This causes *actDCF* to spike toward 1.0, indicating that the actual operating point of the model is effectively ignoring the prior and misclassifying more often. Because the application treats counterfeit detection with a strong prior on the spoofed class, the chosen threshold in actual DCF heavily penalizes certain misclassifications. Once the model's scores collapse toward a less discriminative solution (large λ), it fails to place the decision boundary to respect that prior, driving *actDCF* to 1.0. This model can indeed leverage the dataset's near-linear separability if we keep regularization mild. Too much shrinkage forces the model away from the strongly discriminative dimensions (especially 3, 4, 5, 6), resulting in a trivial or near-trivial predictor under the security-driven prior. That is why we see stable or improving minimum DCF but worsening actual DCF for larger values of λ.

π	**best actDCF**	**best minDCF**
0.20	0.406	**0.358**
0.40	0.403	0.360
0.60	0.399	0.361
0.80	**0.393**	0.362
π_{EMP}	0.402	0.360

Table 6. Weighted Logistic Regression best actDCF and minDCF.

4.2. Prior weighted model

Table **??** shows the results of prior weighted logistic regression models trained with different effective priors. We employ a small set of them to understand the effects of weighting. Regardless of how we weight the classes during training, features retain their discriminative power. This is confirmed by the stability of minDCF. Even though we see an improvement growing the effective prior towards a more positive-heavy distribution, thte overall spread is not that big. This can be explained by the dataset being pretty balanced, having the empirical prior close enough to the tested ones. If features were weaker, these prior weightings would yield larger fluctuations in DCF, but has summarized in Sec. 1, dataset is fairly separable. In the end, weighting helps align the default threshold to the given prior, but since dataset is already separable, any trained model can still reach similarly low minimum DCF by adjusting its threshold afterward.

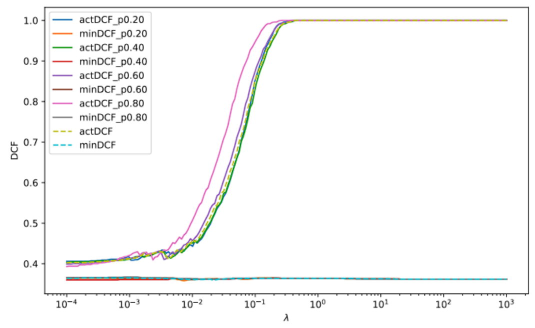


Figure 7. Prior weighted logistic regression DCF over priors and lambda

4.3. Quadratic LogReg

Assuming a quadratic boundary improves the accuracy on our main application, as Table **??** shows. We used the non-weighted model, since there is no significant performance difference. Accuracy peaks above 94 percent (versus 90–91 classic logreg), and minDCF drops to 0.25. This improvement reflects the fact that a set of features exhibit multimodal distributions and strong pairwise interactions (e.g., between 3 and 4 and among 5 and 6). Quadratic expansions let the model combine those features better, reducing overlap in regions that are otherwise inseparable under a linear boundary. We still observe actDCF diverging under heavy regularization, the extra degree of freedom added gets shrunk causing underfitting.

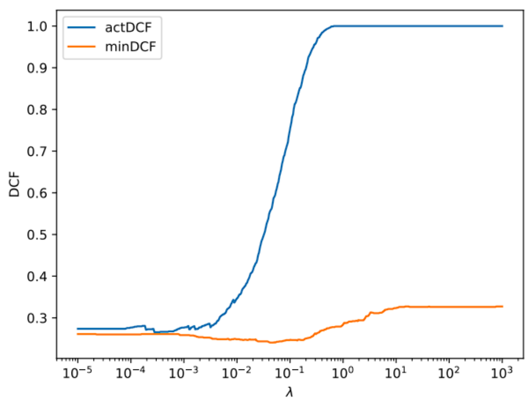


Figure 8. Quadratic logistic regression DCF over lambda

5. Support vector machines

A linear SVM tries to find a single linear boundary that maximizes the margin between the two classes, subject to a penalty term C. The penalty term acts as a regularization term, and given the dataset characteristic, similar to what observed with logistic regression, we obtain poor result when the model is highly regularized.

K	C	accuracy	actDCF	minDCF
1.0	1.00×10^{-5}	49.60	1.000	1.000
1.0	3.16×10^{-5}	90.85	1.000	0.362
1.0	1.00×10^{-4}	90.80	1.000	0.364
1.0	3.16×10^{-4}	90.75	0.999	0.362
1.0	1.00×10^{-3}	90.70	0.959	0.362
1.0	3.16×10^{-3}	90.65	0.845	0.359
1.0	1.00×10^{-2}	90.75	0.672	0.362
1.0	3.16×10^{-2}	90.85	0.579	0.359
1.0	1.00×10^{-1}	90.85	0.516	**0.358**
1.0	3.16×10^{-1}	90.90	0.497	0.358
1.0	1.00×10^{0}	**90.95**	**0.489**	**0.358**

Table 7. Linear SVM model over different C and $K = 1.0$ for effective prior $\tilde{\pi} = 0.1$

Table **??** shows the accuracy and dcf results for this model. When the model prioritizes a wide margin (C is very small) many data points are ignored and this produce near-random predictions. As the term grows, the SVM places more emphasis on fitting misclassified examples, so both accuracy and actual DCF improve. With C near 1, the SVM learns a reasonable boundary on the dataset's features, reaching 91 percent accuracy. Even at the best settings, there's a significant miscalibration: SVM decisions do not inherently calibrate scores to reflect the prior of our applicatin, so without post-hoc adjustments, the default "distance from hyperplane" trhreshold can't capture the cost structure under our effective prior. Results of linear classification are similar to those seen with logistic regression, confirming we can achieve a strong linear boundary, but uncalibrated decision scores lead to higher actual DCF under our effective prior and no calibration.

5.1. Polynomial kernels

We tested the SVM with polynomial kernels of degrees 2,3 and 4. Obtaining a boost in performances, matching what we already saw with logistic regression and in light of the dataset characteristics highlited in Sec. 1.

Degree	K	C	Accuracy (%)	actDCF	minDCF
2	1.0	1.00×10^{-5}	49.60	1.000	1.000
2	1.0	3.16×10^{-5}	92.20	1.000	**0.245**
2	1.0	1.00×10^{-4}	92.80	1.000	0.250
2	1.0	3.16×10^{-4}	92.95	0.994	0.252
2	1.0	1.00×10^{-3}	93.55	0.903	0.250
2	1.0	3.16×10^{-3}	93.70	0.739	0.264
2	1.0	1.00×10^{-2}	93.80	0.571	0.259
2	1.0	3.16×10^{-2}	**94.10**	0.461	**0.239**
2	1.0	1.00×10^{-1}	**94.10**	0.414	0.253
2	1.0	3.16×10^{-1}	**94.05**	**0.389**	0.260
2	1.0	1.00×10^{0}	**94.05**	**0.382**	0.265
3	1.0	1.00×10^{-5}	49.60	1.000	1.000
3	1.0	3.16×10^{-5}	92.35	0.941	0.259
3	1.0	1.00×10^{-4}	92.75	0.842	**0.249**
3	1.0	3.16×10^{-4}	93.05	0.723	0.260
3	1.0	1.00×10^{-3}	93.65	0.609	0.263
3	1.0	3.16×10^{-3}	93.70	0.508	0.257
3	1.0	1.00×10^{-2}	**94.10**	0.446	0.262
3	1.0	3.16×10^{-2}	**94.00**	0.405	0.289
3	1.0	1.00×10^{-1}	**94.00**	**0.383**	0.306
3	1.0	3.16×10^{-1}	**94.00**	**0.380**	0.311
3	1.0	1.00×10^{0}	93.95	**0.379**	0.303

Table 8. Polynomial kernel SVM over different C and polynomial degrees, with $K = 1.0$; the effective prior is $\tilde{\pi} = 0.1$

As shown in Table 8 polynomial kernels of degree 2 and 3 reach accuracies around 94 percent. This gain over the linear SVM's 90–91 percent suggests once again that the added nonlinear terms capture the more complex separation boundaries. Is again confirmed the role of the regularization terms, that shows the same sensitivity of the linear model. Again, minimum and actual DCF gap remains, since the margin decision rule is not calibrated to the prior. Both polynomial degrees 2 and 4 yield similar performance, once the kernel can capture the main interactions of features 3 and 4 and 5-6, going from second to third degree polynomials yields only small boosts.

When we move to degree 4 (table **??**), we see that it moves beyond the quadratic and cubic expansions. The

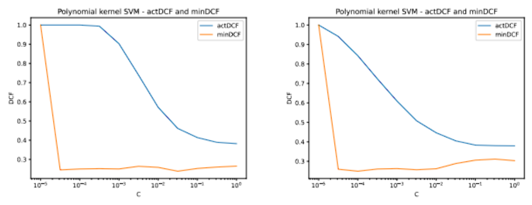


Figure 9. SVM kernels degrees 2 and 3 DCFs

K	**C**	**Accuracy (%)**	**actDCF**	**minDCF**
1.0	1.00×10^{-5}	49.60	1.000	1.000
1.0	3.16×10^{-5}	93.55	0.806	0.215
1.0	1.00×10^{-4}	94.50	0.652	0.211
1.0	3.16×10^{-4}	95.10	0.495	0.192
1.0	1.00×10^{-3}	**95.45**	0.395	**0.179**
1.0	3.16×10^{-3}	**95.65**	0.319	**0.174**
1.0	1.00×10^{-2}	**95.85**	**0.274**	0.190
1.0	3.16×10^{-2}	95.65	**0.259**	0.210
1.0	1.00×10^{-1}	95.55	0.281	0.235
1.0	3.16×10^{-1}	**95.70**	0.272	0.257
1.0	1.00×10^{0}	**95.45**	0.302	0.262

Table 9. Polynomial kernel ($d = 4$) SVM model over different C and $K = 1.0$; the effective prior is $\tilde{\pi} = 0.1$

main reason it performs so well is the presence of multi-peaked distributions in features5 and 5, combined with the strong but not purely linear separability in 3-4. A higher degree polynomial can effectively segment each mode in those multimodal features and capture interactions among all six dimensions. Accuracy reaches 96 percent and minDCF is brought under 0.18 at its best point so far. We maintain the miscalibration gap with our main prior.

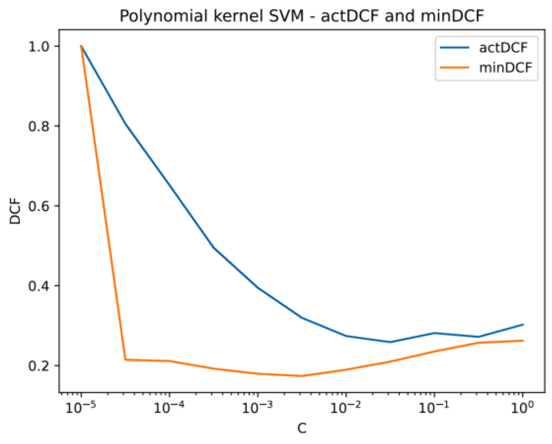


Figure 10. SVM with kernel degree 4 DCFs

5.2. RBF Kernel

Results in table 10 show that we can achieve the highest levels of accuracy and some of the lowest detection costs observed so far. In particular, we can see accuracy values

γ	**C**	**Accuracy (%)**	**actDCF**	**minDCF**
e^{-2}	1.00×10^{-3}	91.20	1.000	0.327
e^{-2}	3.16×10^{-3}	92.15	1.000	0.324
e^{-2}	1.00×10^{-2}	92.70	1.000	0.298
e^{-2}	3.16×10^{-2}	93.15	1.000	0.276
e^{-2}	1.00×10^{-1}	93.35	0.984	0.257
e^{-2}	3.16×10^{-1}	94.25	0.798	0.249
e^{-2}	1.00×10^{0}	94.70	0.675	0.236
e^{-2}	3.16×10^{0}	95.05	0.600	0.237
e^{-2}	1.00×10^{1}	95.30	0.501	0.196
e^{-2}	3.16×10^{1}	**95.50**	0.422	**0.175**
e^{-2}	1.00×10^{2}	**95.85**	0.321	0.199
e^{-1}	1.00×10^{-3}	93.25	1.000	0.329
e^{-1}	3.16×10^{-3}	93.20	1.000	0.331
e^{-1}	1.00×10^{-2}	92.75	1.000	0.290
e^{-1}	3.16×10^{-2}	93.70	1.000	0.272
e^{-1}	1.00×10^{-1}	94.75	1.000	0.236
e^{-1}	3.16×10^{-1}	95.55	0.991	0.202
e^{-1}	1.00×10^{0}	**95.75**	0.901	**0.183**
e^{-1}	3.16×10^{0}	**95.80**	0.715	**0.188**
e^{-1}	1.00×10^{1}	**95.95**	0.537	0.229
e^{-1}	3.16×10^{1}	95.50	0.411	0.268
e^{-1}	1.00×10^{2}	94.90	0.375	0.329

Table 10. SVM with RBF kernel on γ and C; the effective prior is $\tilde{\pi} = 0.1$.

reaching 96 percent, and minDCF going below 0.19. This strong performance aligns with the idea that RBF kernels can adapt locally to the multi-lobed structure in Features 5 and 6 and can also capture subtle relationships among the other dimensions. When the regularization term is too small, model underfits the data because the ”bubbles” in the feature space are large and coarse, producing near-linear or only partially curved boundaries. As we increase C, maintaining a moderate regularization, the model gains flexibility and gain in accuracy without overfitting. RBF shows better ability to form localized boundaries outperforming the linear SVM and matching higher degree polynomial kernels with fewer parameters.

6. Gaussian Mixture Models

We conclude our analysis with the GMM family. The GMM is a probabilistic model that assumesdata is generated by a mixture of several Gaussian distributions: it can be considered as an enhanced version of the MVG, in which data are fit in multiple weighted Gaussian distributions.

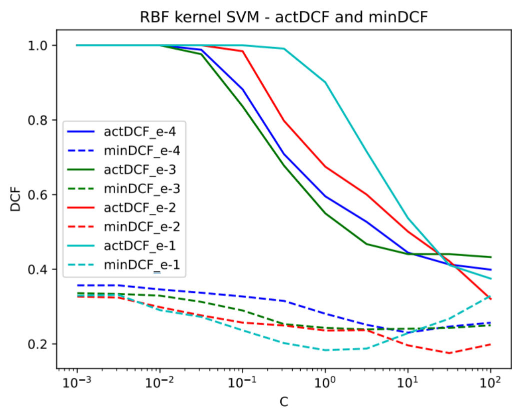


Figure 11. SVM with RBF kernels DCF over lambda

6.1. Full covariance matrix

Let's start with the standard model considering full covariance matrices.

	1	2	4	8	16	32
1	86.05%	86.80%	92.35%	94.30%	94.35%	—
	0.664	0.668	0.483	0.312	**0.251**	—
	0.263	0.265	0.214	0.185	**0.159**	—
2	88.20%	88.55%	93.10%	95.05%	94.80%	—
	0.654	0.615	0.391	0.273	0.252	—
	0.218	0.216	0.223	0.186	0.170	—
4	87.95%	88.20%	92.95%	94.90%	94.55%	—
	0.653	0.653	0.426	0.293	0.271	—
	0.233	0.232	0.216	0.190	0.168	—
8	89.30%	89.75%	93.20%	**95.35%**	95.05%	—
	0.601	0.592	0.358	0.297	0.273	—
	0.183	0.180	0.194	0.184	**0.154**	—
16	88.45%	88.70%	92.50%	94.85%	94.60%	—
	0.672	0.654	0.422	0.312	0.298	—
	0.180	0.185	0.188	0.179	**0.158**	—
32	88.45%	88.75%	92.70%	94.80%	94.60%	—
	0.637	0.620	0.430	0.313	0.296	—
	0.189	0.190	0.190	0.185	0.163	—

Table 11. Results for different clustering components for class 0 and class 1. Covariance type: *Full*; effective prior $\tilde{\pi} = 0.1$.

Table 11 confirms the expectations on GMM performance. Being able to capture multi-peaked distributions by carving out separate Gaussians for each mode, this model is able to capture the different clusters inside each class better than the others. Since feature are largely uncorrelated as shown in Sec. 1, we can confidently say that the full covariance is not contributing that much to these performances. Even though GMMs produce posterior probabilities by comparing class-conditional likelihoods, which in principle should align better with our custom effective prior, we still see gaps between actual and min DCFs, meaning the mixture scores are somehow still miscalibrated.

6.2. Diagonal covariance matrix

Applying a Gaussian Mixture with diagonal covariance matrices results in Table **??**.

	1	2	4	8	16
1	85.80%	85.80%	92.05%	92.10%	94.45%
	0.684	0.696	0.463	0.481	0.284
	0.257	0.261	0.210	0.204	**0.141**
2	86.35%	86.25%	91.90%	91.95%	94.60%
	0.660	0.659	0.466	0.482	0.266
	0.245	0.249	0.199	0.203	**0.154**
4	87.90%	87.75%	93.25%	93.50%	95.05%
	0.660	0.660	0.418	0.416	0.328
	0.145	0.154	**0.148**	**0.140**	**0.136**
8	87.50%	87.45%	93.30%	93.35%	95.05%
	0.622	0.631	0.382	0.390	0.327
	0.174	0.181	0.167	0.151	**0.139**
16	87.60%	87.45%	93.15%	93.30%	94.75%
	0.651	0.659	0.436	0.442	0.377
	0.222	0.222	0.202	0.198	0.157

Table 12. GMM with Diagonal covariance matrix. The effective prior is $\tilde{\pi} = 0.1$.

We already highlighted features uncorrelation in Sec.1, here we once again notice that the complexity of a full covariance matrix is not needed to capture the differences between our classes. The simpler diagonal-covariance GMM ends up also better calibrated for the cost-sensitive application we are using, as shown by the smallest minDCF achieved so far. Ignoring off-diagonal terms provides better robustness and avoids overly confident misclassifications that hurts the DCF when the errors fall in the costly side of them. Having only 6 variance terms per mixture greatly improves the overfitting of the full covariance (instead of 36).

6.3. Tied covariance matrix

Finally, applying the tied assumptions led to results in Table 13. The tied covariance compromise proves to be better than other models, but not superior to the extremes of full and diagonal covariances. While with tied covariances we achieve slightly better accuracy than diagonal, we lose on minDCF.

7. Models comparison

In this section we compare the best performing models of the different families and draw our conclusion in light of the characteristic of the dataset and the main application.

We end up having the Tied GMM at 95.60 percent as best overall accuracy. Its shared covariance is flexible enough to

	1	**2**	**4**	**8**	**16**	**32**
1	86.05%	86.05%	86.05%	94.30%	94.50%	94.50%
	0.664	0.664	0.664	0.321	0.320	0.334
	0.263	0.263	0.263	**0.155**	**0.158**	**0.166**
2	86.05%	86.05%	86.05%	94.30%	94.50%	94.50%
	0.664	0.664	0.664	0.321	0.320	0.334
	0.263	0.263	0.263	**0.155**	**0.158**	**0.166**
4	86.05%	86.05%	86.05%	94.30%	94.50%	94.50%
	0.664	0.664	0.664	0.321	0.320	0.334
	0.263	0.263	0.263	**0.155**	**0.158**	**0.166**
8	89.65%	89.65%	89.65%	**95.60%**	95.55%	95.50%
	0.539	0.539	0.539	0.293	0.275	0.285
	0.156	**0.156**	**0.156**	**0.164**	**0.164**	0.171
16	89.15%	89.15%	89.15%	95.25%	95.30%	**95.50%**
	0.547	0.547	0.547	0.285	0.277	0.287
	0.151	**0.151**	**0.151**	0.167	**0.165**	**0.167**
32	88.90%	88.90%	88.90%	95.20%	94.95%	95.30%
	0.580	0.580	0.580	0.303	0.285	0.285
	0.164	0.164	0.164	0.176	0.182	0.186

Table 13. GMM with Tied covariance matrix. The effective prior is $\tilde{\pi} = 0.1$.

Model	**Accuracy**	**Min DCF**	**Parameters**
LDA	91.00%	—	$m = 1, t_{\text{off}} = +0.05$
MVG	93.00%	0.253	No PCA, Naive
Linear LR	90.75%	0.361	$\lambda = 10^{-2}$
Quadratic LR	94.25%	0.247	$\lambda = 2 \cdot 10^{-2}$, no PCA
Linear SVM	90.95%	0.358	$K = 1.0, C = 1$
Polynomial SVM	95.85%	**0.174**	$deg = 4, K = 1.0, C = 3.16 \times 10^{-3}$
RBF SVM	**95.95%**	0.175	$\gamma = e^{-2}, C = 3.16 \times 10^{1}$
GMM Full	95.35%	**0.154**	No PCA, $c_0 = 8, c_1 = 16$
GMM Diagonal	95.05%	**0.136**	No PCA, $c_0 = 4, c_1 = 16$
GMM Tied	95.60%	**0.155**	No PCA, $c_0 = 1, c_1 = 8$

Table 14. Comparison of different models with their best accuracies, minimum DCF, and parameters. Effective prior is $\tilde{\pi} = 0.1$

capture correlations, and multiple mixture components fit the multi-modal data well. Diagonal GMM has instead the lowest min DCF at 0.136. While ignoring cross-feature covariance might seem suboptimal, it prevents overfitting or incorrect correlation assumptions, yielding more stable posterior estimates under cost-sensitive conditions. Non linear discriminative SVM models also reach nearly 96 percent accuracy, but their min DCF is higher, because SVMs optimize margin rather than the cost function, resulting in class probability miscalibration. Linear models like LDA, single MVG and LogReg cannot fully capture the multimodal aspect of the last two features, resulting in lower accuracy and higher DCF. To sum up we have that, from a purely accuracy standpoint, Tied GMM is the best, but from a cost-sensitive perspective, Diagonal GMM is the best.

8. Calibration

Miscalibration is present in basically in all scenarios, as fig 12 shows. In this final section, we calibrate the scores with an held out dataset using K-fold and drawn the conclusions on the best model. Table 15 shows the results of the calibrated models. As we expected, the best model results in the diagonal GMM. This proved to be the model that best adapts to our main application and to the dataset uncorrelated features.

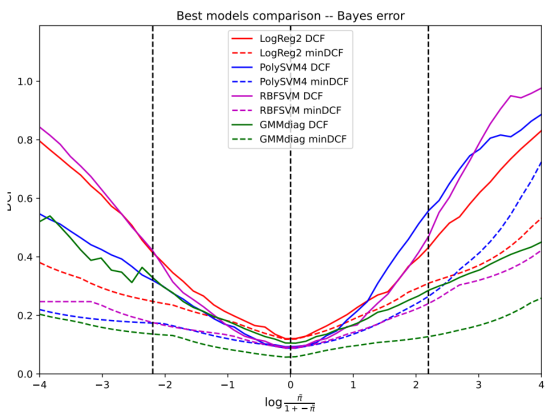


Figure 12. SVM with RBF kernels DCF over lambda

Model	**Accuracy**	**actDCF**	**minDCF**
Validation set			
Quad LR	94.25%	0.270	0.254
Poly4 SVM	95.75%	0.187	0.177
RBF SVM	95.50%	0.189	0.178
GMM diag	**96.85%**	**0.166**	**0.138**

Table 15. K-fold calibration results on Validation set.