Université de Genève

Chosen Chapter

14x060

TP 2 : Steganalysis based on fusing classifiers built on random subspaces

Author: Deniz Sungurtekin

E-mail: Deniz.Sungurtekin@etu.unige.ch

Mars 2020

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

Département d'informatique

1 Introduction

In this small work, we will quickly investigate steganalysis based on fusing classifiers built on random subspaces. First, we will form two high-dimensional prefeatures where each individual in rows is a samples (Stego or cover image) and each column are their features. Both matrix have the same dimension and are ordered by names, so the i^{th} image in the cover matrix have his corresponding stego in the same row in the other matrix. The idea is to implement a model capable to classify an image telling if it's a stego or not. Our model will create L random subspace where the number of chosen features are random. Finally each L subspace will classify (using Fisher Linear Discriminants) an image and the final prediction is obtained by majority voting (Fusion) of individual base learners.

2 Impact of the size of training set

We know that generally the more the training set is big the best are the predictions:

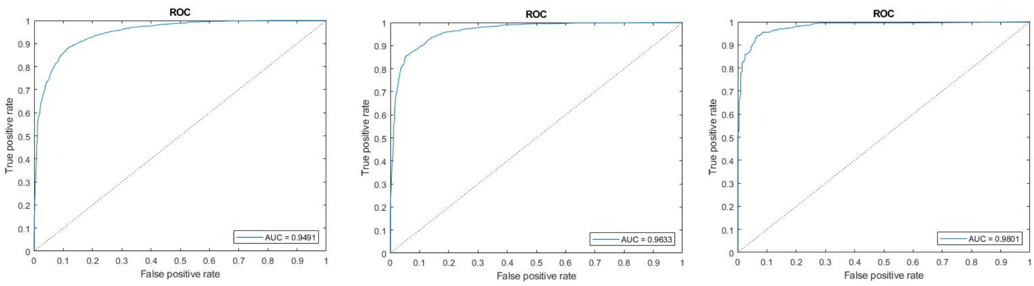

Figure 1: 20% / 50% / 90% of the samples

Clearly this assumption is true for this case but we can see that even with 20% of the samples we have good results. The accuracy of our model improve very quickly with the first training samples, the more we add samples the less we have amelioration. (Logarithmic evolution)

3 Impact of cover data distortions

In this section, we will analyse the impact of noise on our training cover dataset. We will observe the accuracy of our model after adding a white Gaussian noise with different values of standard deviation:

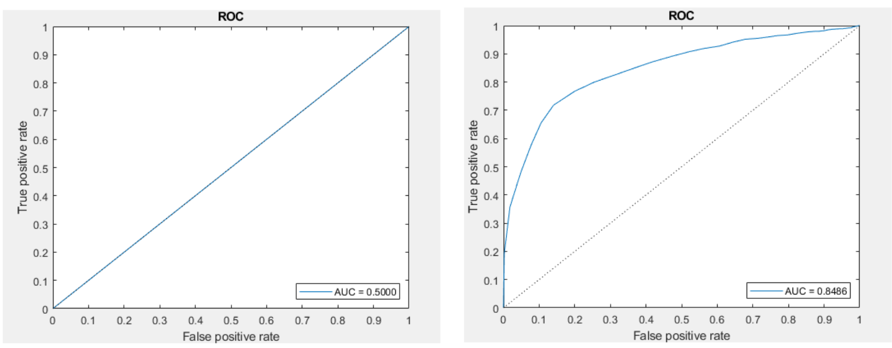

Figure 2: 0.1 and 10^{-5} std noise

Because of the very small value of our features even a little noise on our cover images have an huge impact on our prediction. We can see with these two plots that the model isn't robust to noise. With std = 0.1 we clearly see that the prediction are random because we have 50% of accuracy which is very bad because there is only two labels value: 1 or -1. For the second case we have a very little noise and 50% of the training dataset size. Consequently, we can compare it to the previous figure where we plot the accuracy without noise with 50% of the samples, we can observe that we lost more than 10% of accuracy which is huge compared to the power of the noise.