AI in Cryptography – project 1

Adam Korba – index: 151962

1. Plaintext dataset:

   Webpages:
   - Historia_Politechnika Poznańska.html
   - Artificial intelligence - Wikipedia.html
   - Command Line Interface Guidelines.html

   Binaries:
   - vmlinuz-6.1.0-37-amd64
   - grep.elf

   Source code:
   - README.md
   - stdtypes.rst
   - ast.c
   - turtle.py
   - main.c

   Datasets:
   - titanic.csv
   - pg100.txt
   - california_housing.csv

   Images (bitmaps):
   - windows.bmp
   - tux.bmp
   -

2. High-entropy dataset

Window sizes of 32KB are converted to high-entry blocks by:

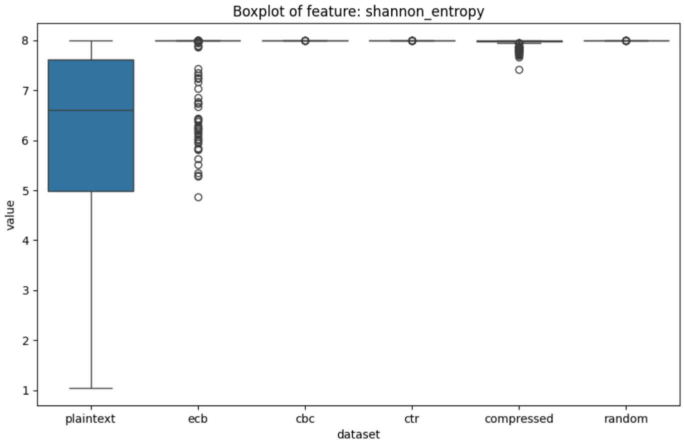
- ECB AES mode of operation
- CBC AES mode of operation
- CTR AES mode of operation
- zlib.compress
- os.urandom

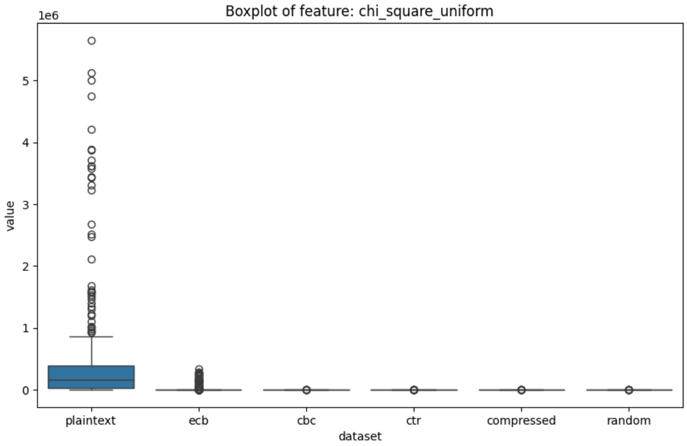
3. Feature visualization

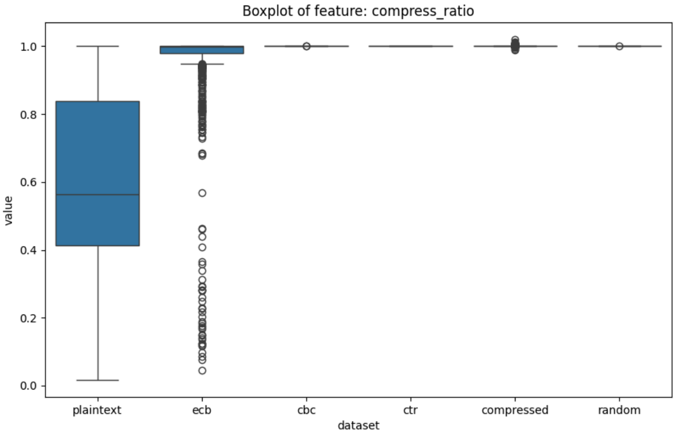
For each window the following features were obtained:

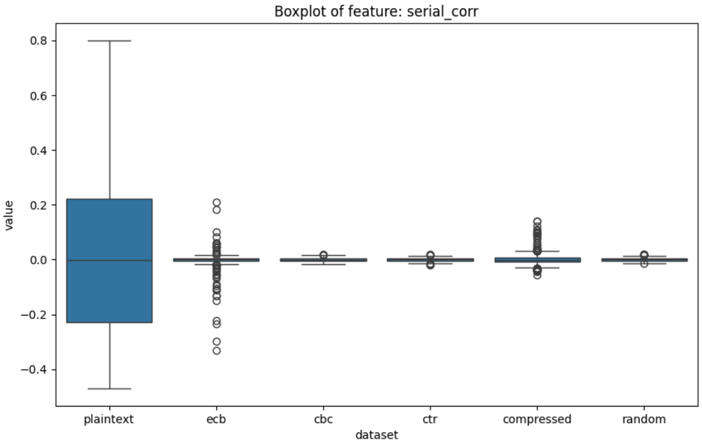
- Shannon entropy
- Chi-square
- compressibility (zlib ratio)
- serial correlation
- FFT flatness
- 16-byte block duplicates

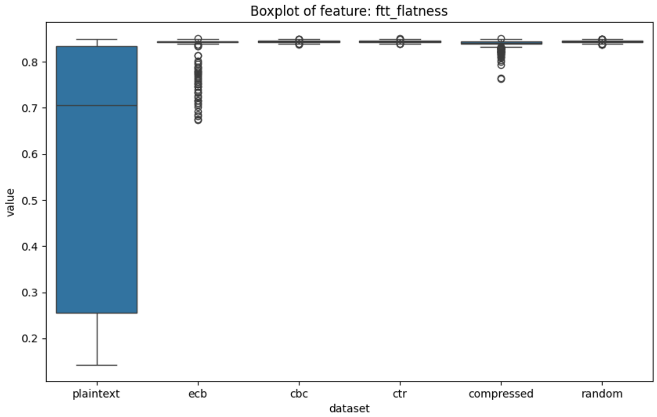
Graphs below showcase distributions of values of each of that feature for each of the dataset types

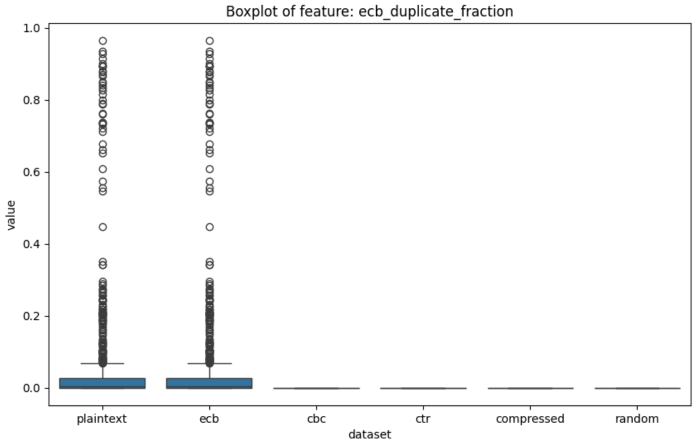








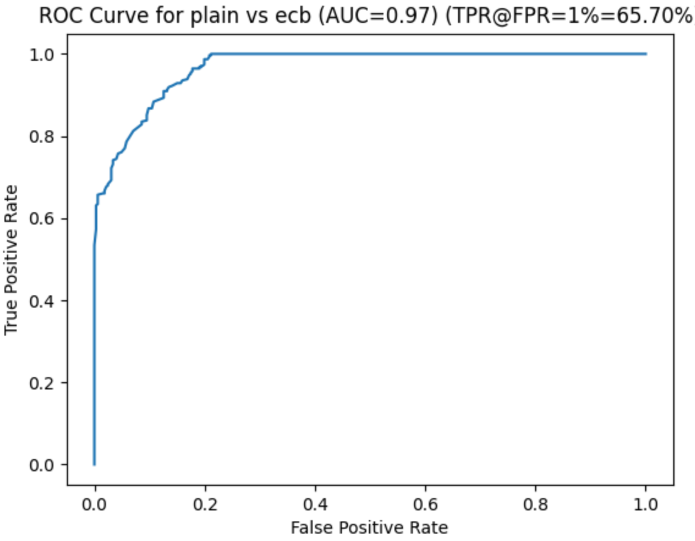


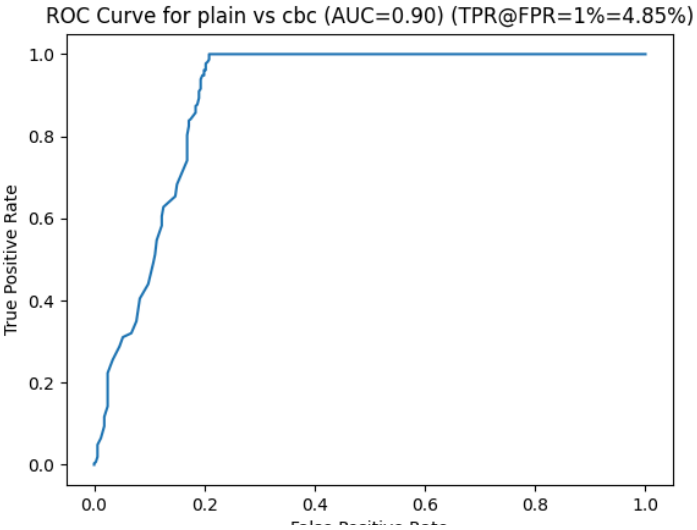


As you can see the high-entropy datasets share similar means and standard deviations for most of these features, while plaintext one is quite different from them (for example smaller entropy, and better compressibility ratio). We can see that the duplicate fraction is exactly the same for plaintext and ecb because ECB maps the same plaintext blocks to the same encrypted blocks.

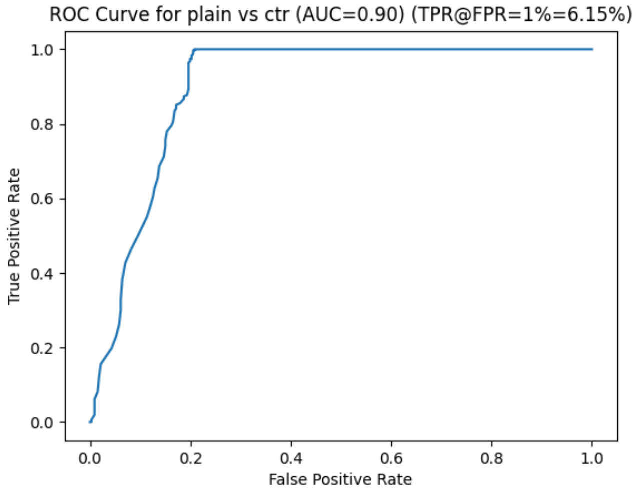
4. Random Forest results
   The Random Forest classifier was trained on different pairs of datasets to find out how difficult it is differentiating between them. Starting with detecting plaintext vs all high-entropy datasets:

   AES modes of operation



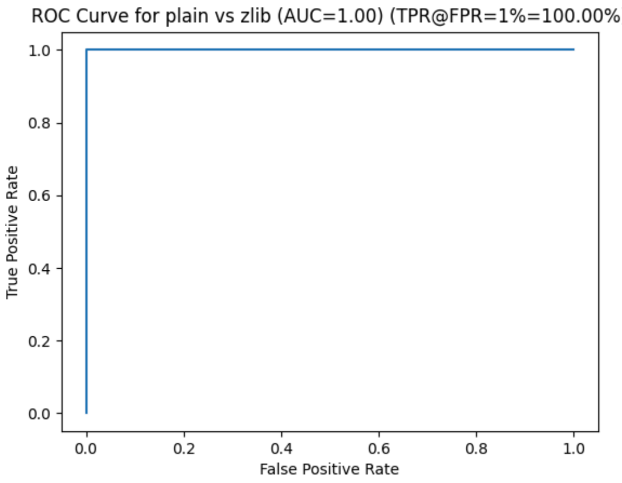


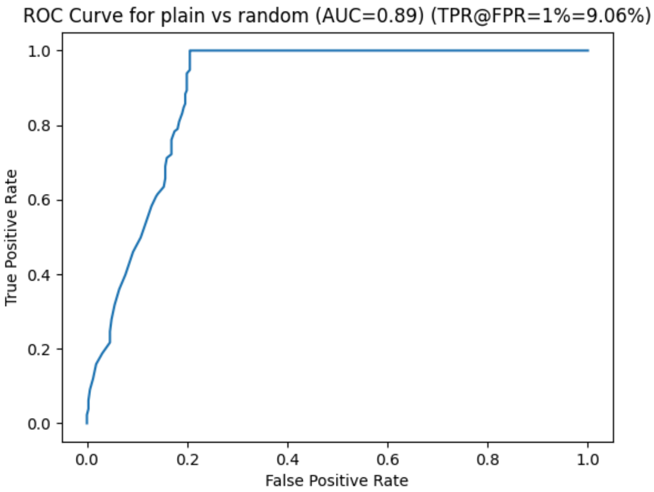
It seems that distinguishing between plaintext and EBC was the easiest with AUC=0.97 and TPR@FPR=1% = 65.7% while CBC and CTR seemed to be similarly difficult with AUC=0.90 and TPR@FPR=1% respectively 4.85% and 6.15%

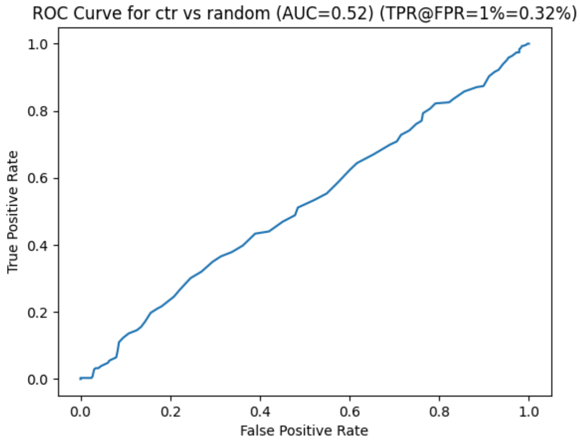


When it comes to distinguishing between plaintext vs zlib random forest was able to classify all the examples correctly, this is probably because feature compressibility ratio is always 1 for zlib class

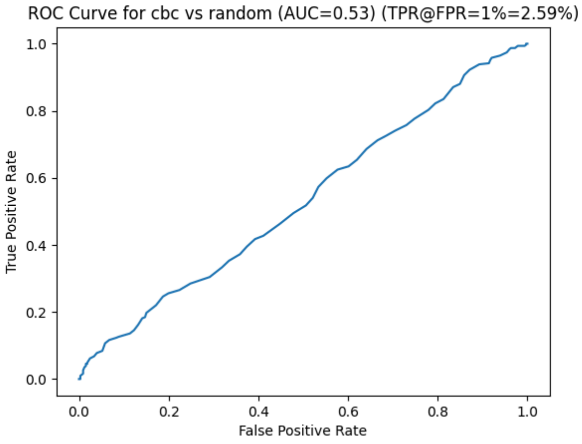
in detecting totally random blocks classifier obtained following results: AUC=0.89 and TPR@FPR1%=9.06%



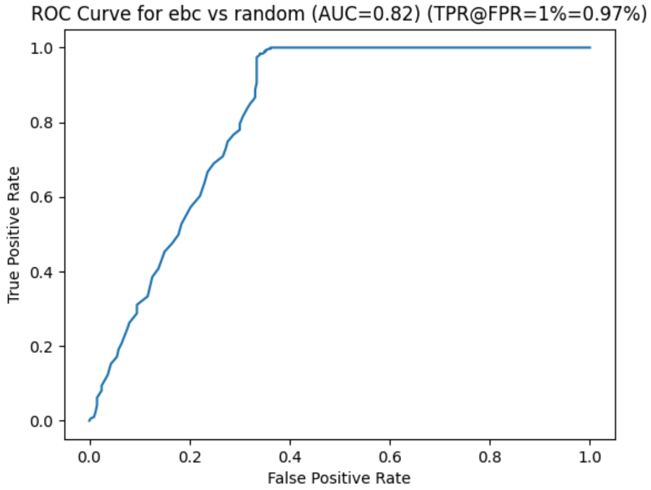




ROC Curve for ctr vs random (AUC=0.52) (TPR@FPR=1%=0.32%)



ROC Curve for cbc vs random (AUC=0.53) (TPR@FPR=1%=2.59%)



ROC Curve for ebc vs random (AUC=0.82) (TPR@FPR=1%=0.97%)

In these charts you can see that it was almost impossible for the model to distinguish between random data and CTR or CBC which makes these modes of operation favourable choices if we want to securely encrypt the data. EBC again in this category performed poorly.