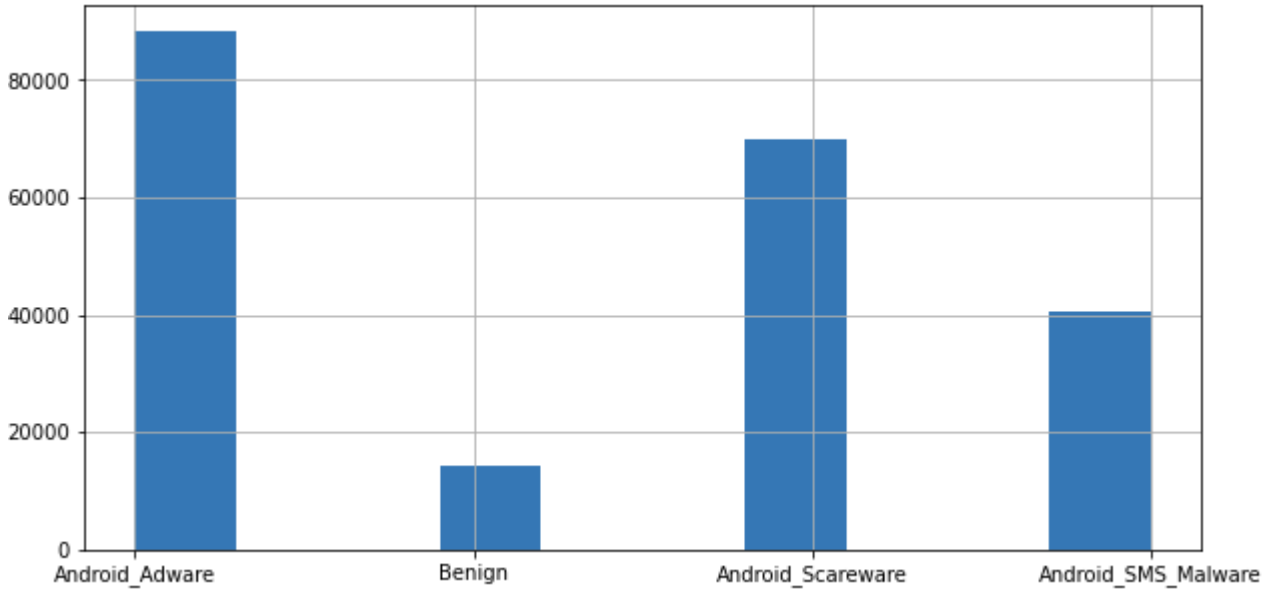
CLASSIFICATION MODEL ANDROID_MALWARE

Michał Iwicki, Mateusz Nizwantowski

ABOUT DATA SET

- This data set contains 86 columns and over 350 000 observations. It gives us informations about the path and sizes of packages of data between android device and potential malware.
- There are 4 classes of android applications:

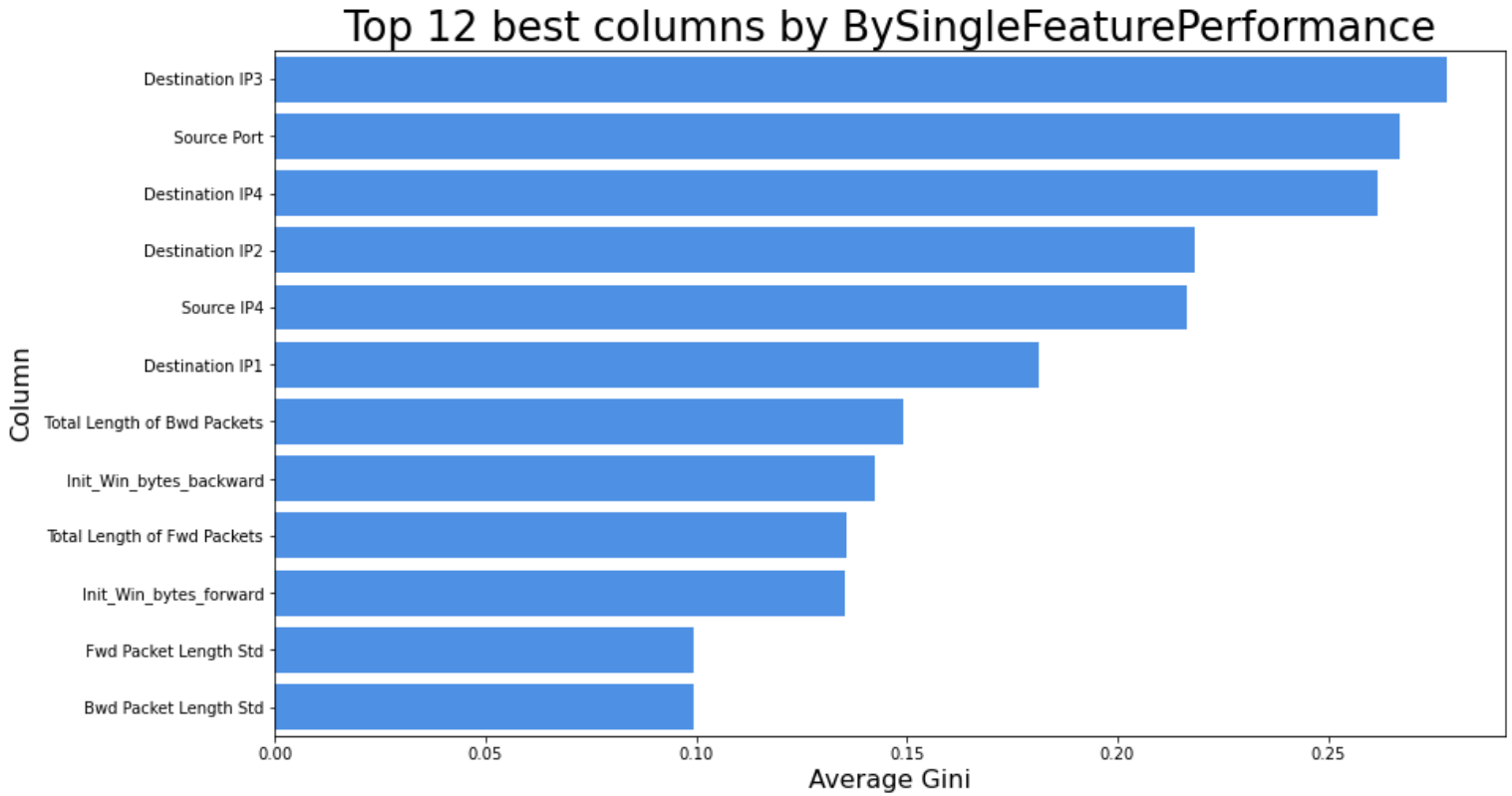
1. Benign – is not harmful at all, coded as 0
2. Adware – shows unwanted adds to a user, coded as 1
3. Scareware – sends fake threating messages hoping to gain some benefits from user, coded as 2
4. Malware – is the most dangerous one, coded as 3



CHOOSING COLUMNS FOR OUR MODEL

After preprocesing we reduced highly corelated columns using spearman method with 0.95 threshold. After that we started searching for the most useful variables. It turned out that most of them are connected with path of packages not sizes.
We chose following columns:

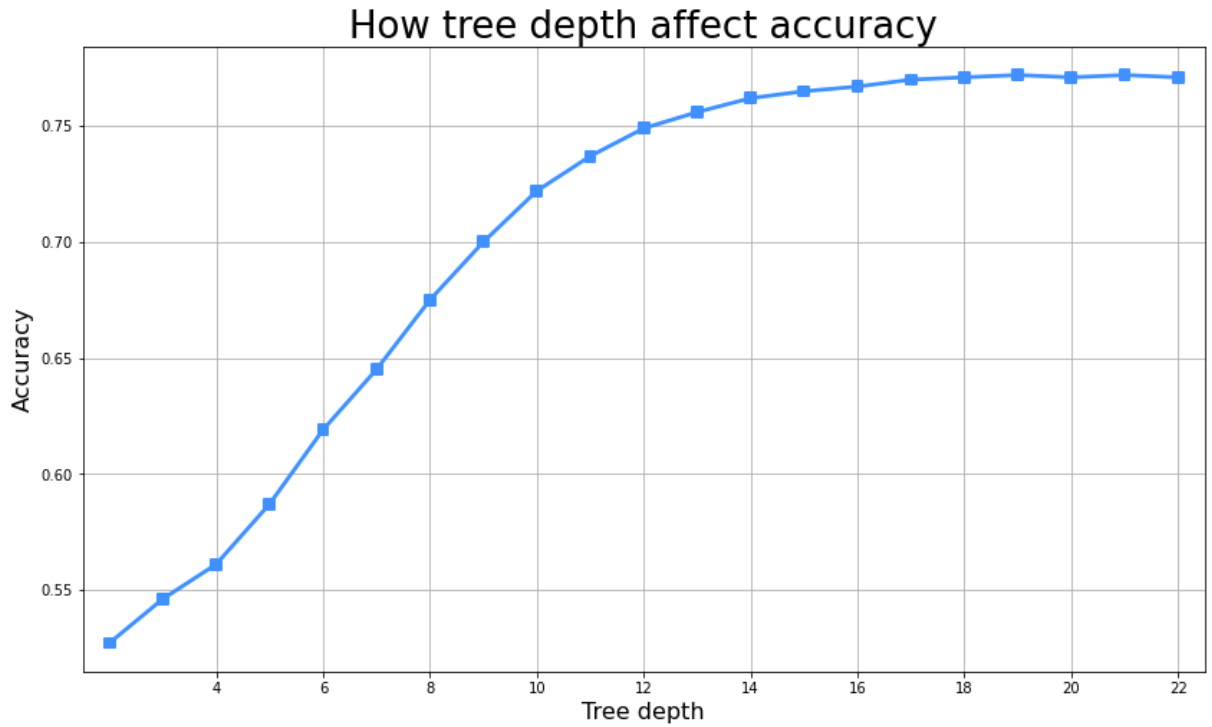
```
['Destination IP3',
 ' Source Port',
 'Destination IP4',
 'Destination IP2',
 'Source IP4',
 'Destination IP1',
 ' Destination Port',
 'Source IP3',
 'Source IP2',
 'Source IP1']
```



FINDING THE BEST HIPERPARAMETERS

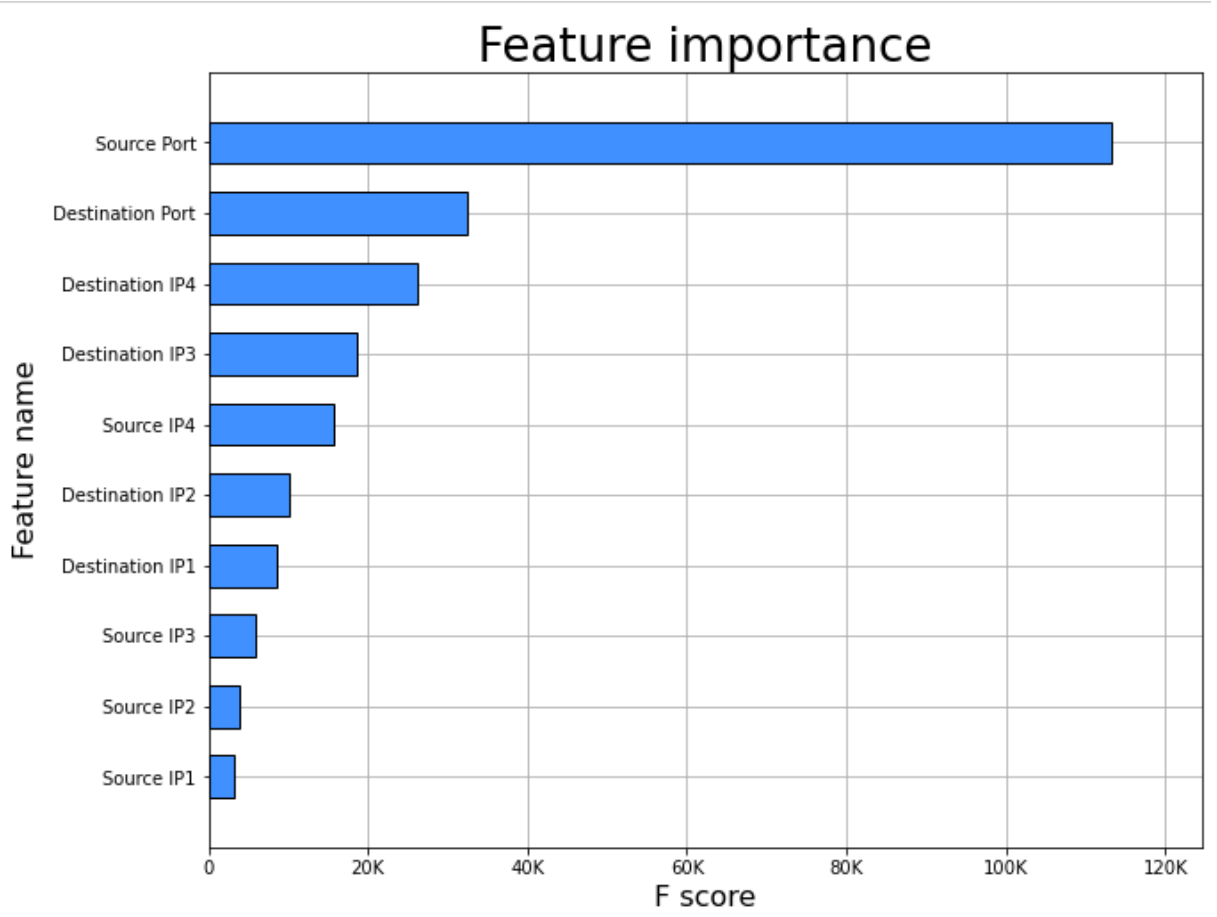
After using
sklearn.model_selection.GridSearchCV
(with crossvalidation)
on XGBoostClassifier with tree booster
we got following parameters:

- learning_rate= 0.67
- max_depth = 16
- n_estimators = 120

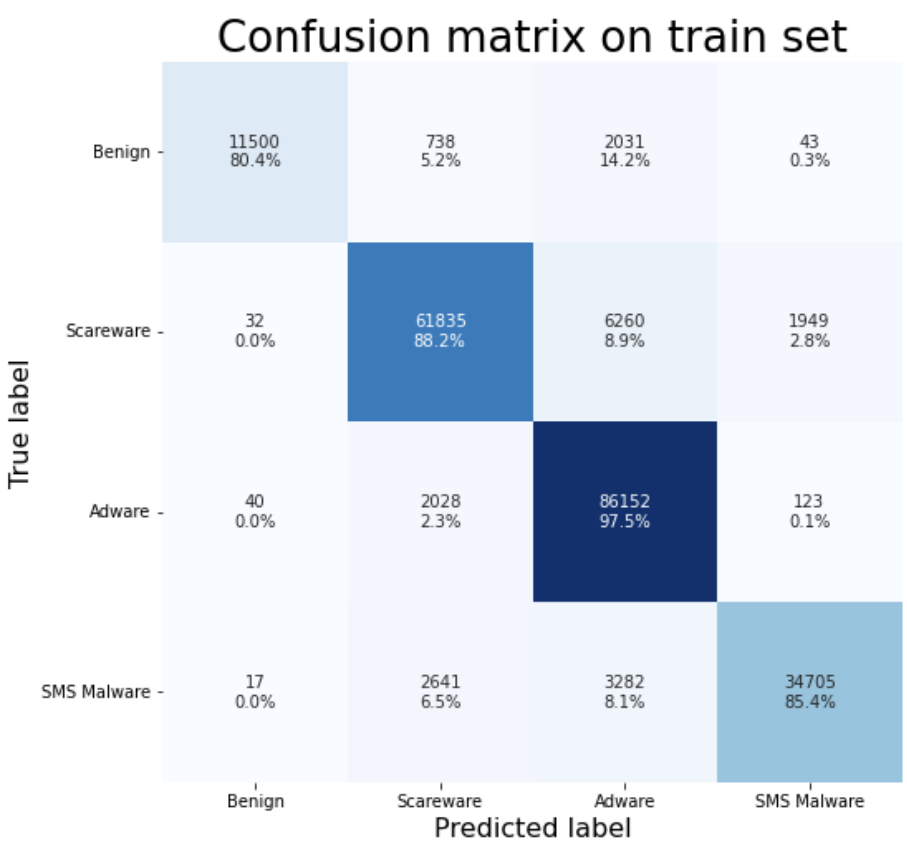


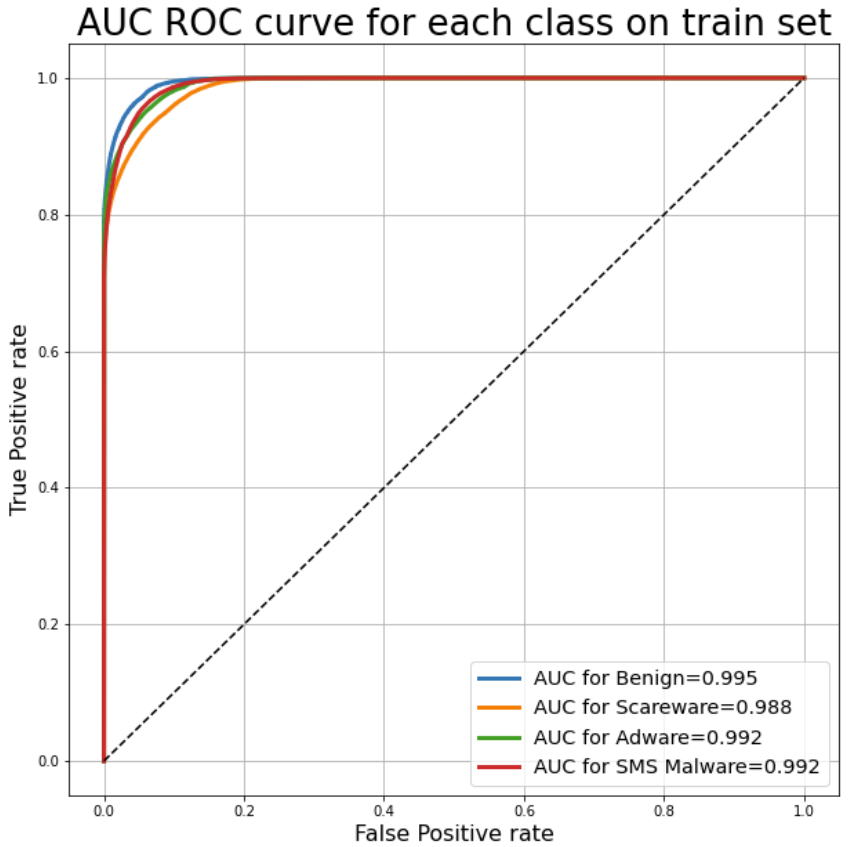
IMPLEMENTATION

- Our model achieved following results:
  - Accuracy on test set = 80%
  - Accuracy on train set = 91%
  - Accuracy on validation = 80%
  - Average accuracy on crossvalidation = 79%
  - Std on crossvalidation = 1.6%
  - Multiclass roc_auc_score on test set = 0.94
  - Multiclass roc_auc_score on train set = 0.99
  - Multiclass roc_auc_score on validation = 0.94

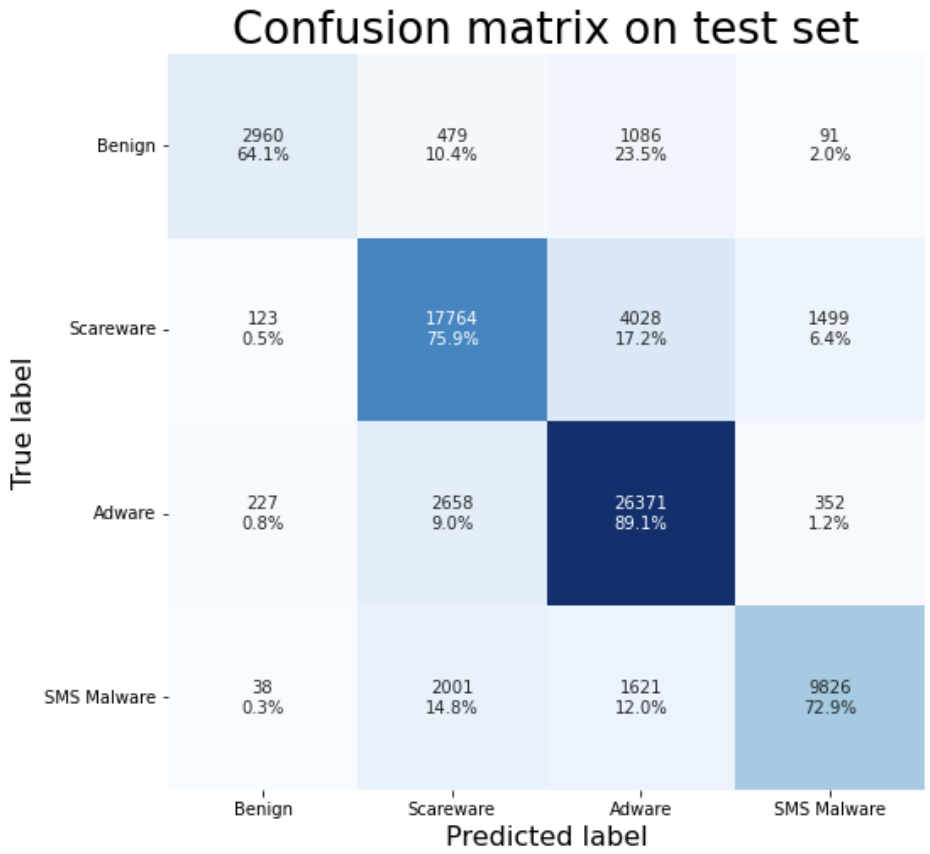


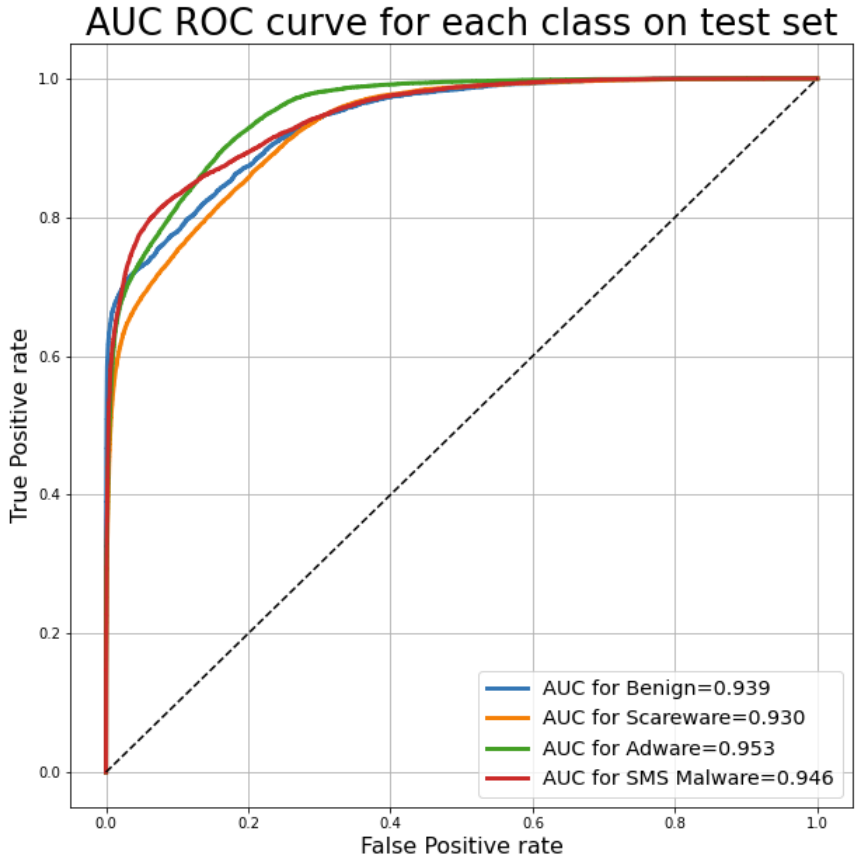
RESULTS ON TRAIN SET





RESULTS ON TEST SET





CONTRIBUTION OF VALIDATION TEAM

- Throughout the project, we worked closely with a validation team who provided us with valuable feedback on our approach. One of the suggestions from the validation team was to try data normalization and standardization to improve the performance of our model. We implemented this suggestion and tested the model again, but unfortunately, we did not observe any significant improvement in the model's performance.

- The validation team also provided us with feedback on the model selection process. We took their suggestions into consideration and adjusted our model accordingly. Overall, the collaboration with the validation team was instrumental in helping us refine our approach and improve the performance of our model.

CONCLUSION

Results of our model are satisfyingly good. We focused on detecting any possible threats and we achieved it. Our model detects (on test set) more then 98% off all dangerous activities and got 80% accuracy in predicting every category. Crossvalidation showed similar results and proved stability of our model. Unfortunately as every tool our model has some potential issues, which we are aware of. For example it has a slight tendency to overfit, what is more it's hard to predict the scores on data set with different malware and enviroment. We think that the problem of detecting malware is a battle between producers of harmful software and IT security specialists. For every solution of one side another one sooner or later will find good answer.