CS 6350, DS 4350 Project: Android Malware Detection

1 Introduction and Task definition

Each example in this classification task is an Android app. The task is to predict whether the application is malicious or not. This data was published at a workshop on security and privacy analytics[1]. Each application was fingerprinted by observing the system calls it made during execution. Each feature in the feature vector corresponds to the number of times a particular system call was made by it. (The paper discusses two feature sets, we provide only the first one here.)

The goal of this project is to explore how well different learning algorithms can learn to detect malicious apps.

2 Data

The data directory contains the following three data files (one training set and two test sets):

1. `data/train.csv`: This is the training set, in a CSV format. There are 7597 training examples.
2. `data/test.csv`: This is the set of examples on which you will report results in your final report. There are 2531 test examples.
3. `data/eval.anon.csv`: These 2532 examples are all labeled positive in the provided data set. You should use your models to make predictions on each example and upload them to Kaggle. See below for the format of the upload. Half of these examples are used to produce the public leader board. The other half will be used to evaluate your results.

All the CSV files have one example per row. The column `label` contains the label, and all the other columns (whose names start with `x`) are features.

In addition to the files with the examples, the directory also contains a file called `data/eval.ids`. This file has as many rows as the `eval.anon.csv` file. Each line consists of an example id, that uniquely identifies the evaluation example. The ids from this file will be used to match your uploaded predictions on Kaggle.

In all, there are 360 features. Note that as part of your project, you are welcome to try feature space expansions, neural networks, etc.

[1] Dimjašević, M., Atzeni, S., Ugrina, I. and Rakamaric, Z., 2016, March. Evaluation of Android Malware Detection Based on System Calls. In Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics (pp. 1-8). ACM.

3 Evaluation

The data is imbalanced—there are many more negative examples than positives. We will use the F_1 score[2] to evaluate the classifiers.

The F_1 score evaluates the quality of a classifier using its precision p and recall r. Precision is the ratio of true positives (tp) to all predicted positives (tp + fp). Recall is the ratio of true positives to all actual positives (tp + fn). The F_1 score is given by:

$$F_1 = 2\frac{p.r}{p+r}$$

where the precision p and recall r are defined as:

$$p = \frac{\text{tp}}{\text{tp} + \text{fp}}$$
$$r = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

Note: The examples are all split randomly among the three files. So we expect that the cross-validation performance on the training set and the F_1 scores on the test set and the public and private splits of the evaluation set will be similar.

4 Submission format

Kaggle accepts a CSV file with your predictions on the examples in the evaluation data. There should be a header line containing `example_id,label`. Each subsequent line should consist of two entries: The example id (from the file `data/eval.ids`) and the prediction (0 or 1).

Here's an example of the submission file (of course, this is not a valid submission because it does not contain all the entries in the `data/eval.ids` file:

```
example_id, label
2591,0
10174,1
4764,0
5429,1
11235,0
...
```

We have provided two sample solutions for your reference:

1. `sample-solutions/sample-solutions.all.positive.csv`: Where all examples are labeled as positive

2. `sample-solutions/sample-solutions.half-neg.csv`: Where the first half of examples are labeled false and the second half are labeled true

[2] https://en.wikipedia.org/wiki/F1_score

5 Project rules

- You should work **individually** on the project.
- You cannot sign up to Kaggle from multiple accounts and therefore you cannot submit from multiple accounts.
- You may submit a maximum of 4 entries per day.
- The end date for the project is the day of the final exam, i.e., ***April 22, 2025 11:59 PM Utah time***.
- You should to submit **at least** six different non-trivial submissions to Kaggle. Here are the rules for these submissions:

  1. You should use at least four **different** learning algorithms we see in class. You cannot use any machine learning library for these five algorithms, and instead implement them by yourself.
  2. For at most one of your six submissions, you are welcome to use a machine learning library such as PyTorch, TensorFlow or `scikit-learn`.

6 Milestones

The project is organized into several milestones summarized below. The deadlines and submissions will be managed via Canvas.

1. **Project information** (10 points): You will need to have registered for the Kaggle competition and made a dummy submission. On canvas, you should also submit your kaggle user details and the score you get for the dummy submission.

2. **Project checkpoint 1** (15 points): For this milestone, you will need to have downloaded the data, and also perhaps run some initial pre-processing on it. You should also have made at least one non-dummy submission on Kaggle. You should submit a one page report on Canvas that describes what you did so far, descriptive statistics about the dataset, and your plan till the next milestone.

3. **Project checkpoint 2** (30 points): This milestone is similar to the previous one. You will need to have made at least two additional submissions to Kaggle. You should submit a one-page report detailing updates after the first milestone, any challenges you have faced, and your plan for the rest of the semester.

4. **Final report** (45 points): By this time, you should have made at least six non-dummy submissions on Kaggle totally. You should submit a final report of at most six pages that is structured like a small research paper. Broadly speaking it should describe:

   (a) An overview of the project.
   (b) What are the important ideas you explored?
   (c) What ideas from the class did you use?
   (d) What did you learn?

(e) A summary and discussion of results

(f) If you had much more time, how would you continue the project?

Each of these components will be equally weighted in the report grade.