SAPIENZA UNIVERSITY OF ROME

Machine Learning

First Homework Report

Author

Silvio Severino

ID number 1705967

Date 18, Nov 2018

Abstract

This report will show some considerations about malware analysis using machine learning. In particular will show tree techniques of machine learning and their respective results.

Contents

List of Figures

Chapter 1

State of art

1.1 Malware analysis

Malware analysis is the study the origin and the potential impact of a given malware sample, such as a virus, worm, trojan and so on. The principle of malware is to harm the operating system or to steal sensitive data. Moreover, this kind of software, can steal user information without permission. Nowdays exist two kind of analysis: the static and dynamic malware analysis.
The first kind is usually performed by dissecting the different resources of the binary file whitout executing it and, after, studying each component. This kind of analysis it was used in this relation.
While, the second kind of analysis is performed by oberving the behavior of the malware while it's actually running on a host system, ensuring that it does not infect the system in use [wik18c].

1.1.1 Malware in Android system

In particular, in addition to windows system, the OS most attacked by malware is the Android one. The reasons why this phenomenon is common in those systems are mainly two. The first reason is that, nowdays, Androis is the most popular operating system in the world and mostly that it allows that the apps can be downloaded by third-party markets. Thanks by the last one reason, the app can be modify, installing a malware

inside. The Android platform, to remedy, provides several security meansures that harden the installation of malware, most notably the Adroid permission system. To perform certain tasks on the device, such as sending a SMS message, each application has to explicitly request permission from the user during the installation. However, many users tend to blindly grant permission and thereby un-dermine the purpose of the permission system [DA14] [MS13].
According to a recent study over 55,000 malicious applications and 119 new malware family have been discovered in 2012 alone [DA14].

1.1.2 Machine learning in malware analysis

The **machine learning** can help the detection of a malicious app. Teaching a system to recognize a malware, it is possible to prevent automatically the infection of a smartphone or others informatic system. To be able to train a machine learning system, a dataset must used. In this case, for this report, the author has used the dataset provided by the drebin system: a research of the University of Göttingen and the Siemens CERT. The dataset contains some features of the Android app, extracted from the various manifest.xml. In the next chapter it will be described how it was possible to use the features of Drebin to learn a machine learning system.

Chapter 2

Methodologies and Technologies

2.1 Drebin data set

The **Drebin data set** is a collection of about 120,000 files, which each one represent an Android application. For each application there is a text file. The text file describes al the properties of the application. Each property belong to one of 8 categories, from S1 to S8: S_1 Hardware components, S_2 Requested permissions, S_3 App components, S_4 Filtered intents, S_5 Resctricted API calls, S_6 Used permissions, S_7 Suspicious API calls, S_8 Network addresses. For this report, each text file that represent an Android app, it was transformed in a feature vector thus to enumerate the occurrences of each category in each file. For example, one application can have a feature vector formed by $\{0, 10, 2, 1, 4, 7, 3, 10\}$. It means that S_1 category occurs 0 time, S_2 category occurs 10 times, S_3 category occurs 2 times and so on [DA14].

2.2 Classification methods

Understand if an application contains malware or not, is a classification problem. Thanks of the feature vectors described in the previous section, it was possible to use some classification method to train the machine learning system. Among the many have been chosen three, explained in detail below.

2.2.1 Logistic regression

This is a regression model applied in cases where the dependent variable y is of the dichotomous type attributable to the values 0 and 1, as they are all the variables that can assume only two values: true or false, malware or not, and so on. For this model, it's used the **logit** function:

$$logit(p) = \beta_0 + \beta_1\chi_1 + \beta_2\chi_2 + \ldots + \beta_\kappa\chi_\kappa = X\beta$$

Where p is the probability that the event occurs [wik18b].
It was the first method that it was used in this homework because of its simplicity and fast computation.

2.2.2 Random forest

The **Random forest** is an enseble learning method for classification, regression and other tasks. Their functioning is based on the decision tree. In fact, at training time a multitude of decision trees are built, and outuputting the class that is the mode of the classes (for classification) or mean prediction (for regression) [wik18d]. RF is a robust algorithm and usually perform very well, but, this algorithm is complicated to tune. For this homework it was used two parameters for tuning: n_estimators and max_features. Another pecularity of RF is that it allows to classify the importance of variables; in this case the features. In the following plot is illustred the features importance. In particular, it's possibile to view that the mainly features that allows to classify a malware: the number of hardware components that the application required (S_1), the number of suspicours API calls (S_7) and the technique of request permissions (S_2), mentioned by importance.

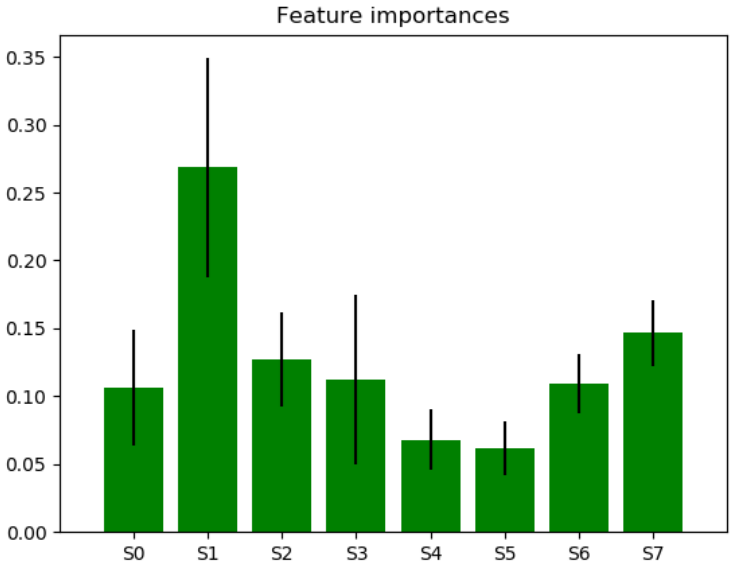

Figure 2.1: Feature importance

2.2.3 Support Vector Machine

The **Support Vector Machine** is a supervised learning models that analyze data used for classification and regression. Given a set of training examples, each marked as belonging to one or the other of two categories, the SVM algorithm builds a model that assigns examples to one category or the other, making it a non-probabilistic binary linear classifier. Its pecularity is that it manages to classify the examples by finding the best hyperplane that divides them [wik18e]. It was used in this homework for its precision and its perform.

2.3 Libraries

For this homework the following open source libraries have been chosen.

2.3.1 Numpy

Numpy is the most python library used for the math problems. Is including in that some math functions for vectors, matrices and so on.

2.3.2 Scikit

Scikit-learn is the most open source python library used for machine learning problems. Is including in that some algorithm for classification, regression, SVM and so on.

2.3.3 Matplot

The **Matplot** is used to create a plot in easy way.

Chapter 3

Problems and Performance

3.1 Malware classification

In the first part of homework it was necessary to create an application that can predict if a given file it has malware or not. A good metrics for this problem is F1 score. F1 score is a measure of a test's accuracy. It considers both the precision p, where p is the number of correct positive results divided by the number of all positive results, and the recall r, where r is the number of correct positive results divided by the number of all relevant samples [wik18a]. This kind of metrics was to be used because, with the drebin datased, there was a problem of class imbalance: more than 5k positive examples and more than 100k negative examples. Experiment showed that trained the model with different size of positive or negative examples, the accuracy didn't change much (about 0,02 points), unlike F1 score (about 0.3 points).

3.1.1 Algorithms and Performance

The first algorithm used, has been **Logistic regression**. But, after its fast tuning, its result model increased the F1 score of only 0,001 points, with an accuracy of about 0.8%. For these reasons I chose to try another algorithm.
The second algorithm used, has been **Support vector machine**. Thanks SVM it was possible to obtain some success. In fact, both the F1 score and accuracy ranged between the 0.9% and 0.91%. The only one limit of this algorithm was the difficult of training.

The last one algorithm used, has been **Random forest**. Has been the best algorithm in term of performance. In fact, it was possible to obtain both a F1 score and accuracy of over than 0.9%, about of 0.94%.
The details of accuracy and execution times are shown below.

Algorithm	Accuracy	F1 score
Logistic regression	0.82%	0.81%
Support vector machine	0.91%	0.90%
Random forest	0.94%	0.94%

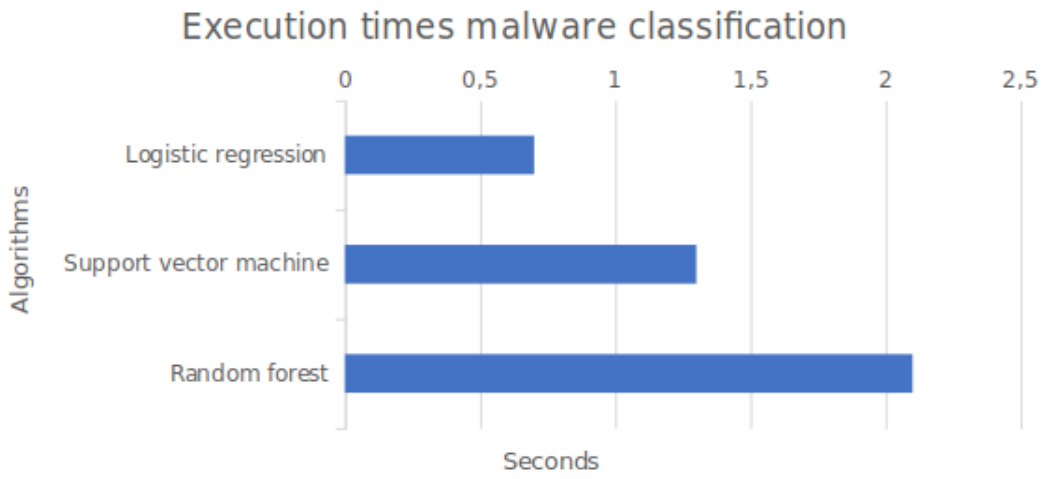

Figure 3.1: Execution times for malware classificaton

3.2 Family classification

In the last one part of homework it was necessary to create an application that can classify a given malware which family it belong to. Also for this application, has been used the Drebin dataset, as it provides more than 5k examples belonging to one of 179 malware families.

3.2.1 Algorithms and Performance

The first algorithm used has been **Logistic regression**. Like for the malware classification, this algorithm was resulted the most easy in terms of computation. However, the results were disappointing, with an accuracy of 0.0631%.
The second algorithm used has been **Support vector machine**. This algorithm has obtained results a lot better than logistic regression without modify the default parameters. It was obtained an accuracy about of 0.86% just finding the best combination of C and γ parameters. So an increase of 0.2% in terms of accuracy compared to logistic regression algorithm.
The last one algorithm used has been **Random forest**. This algorithm has obtained the best result, with an accuracy of 0.88%. However, the support vector machine algorithm remains a good choice.
The details of accuracy and execution times are shown below.

Algorithm	Accuracy
Logistic regression	0.631%
Support vector machine	0.864%
Random forest	0.889%

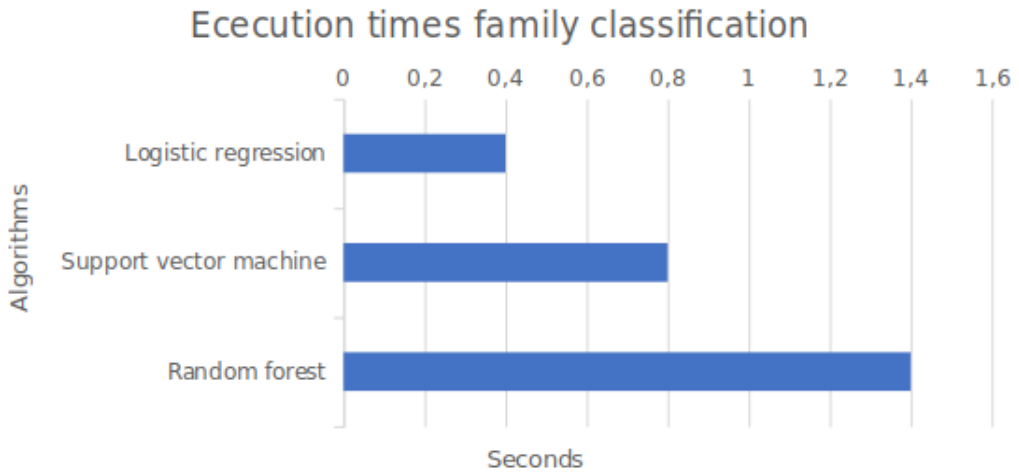

Figure 3.2: Execution times for family classificaton

Conclusions

Results

In all two different problems in this homework, the results are similar. Both random forest and support vector machine algorithms, performed better than logistic regression, both in malware classification problem than in malware family classification.

Final considerations

This homework was very useful for understanding some techniques of machine learning.

Bibliography

[DA14] M. Hübner D. Arp, M. Spreitzenbarth. Drebin : Effective and explainable detection of android malware in your pocket. *21th Annual Network and Distributed System Security Symposium (NDSS)*, February 2014.

[MS13] F. Echtler M. Spreitzenbarth. Mobilesandbox: Looking deeper into android applications. *28th International ACM Symposium on Applied Computing (SAC)*, March 2013.

[wik18a] F1 score, 2018.

[wik18b] Logistic regression, 2018.

[wik18c] Malware analysis, 2018.

[wik18d] Random forest, 2018.

[wik18e] Support vector machine, 2018.