Machine Learning

Home Work I

Android Malware Detection

Ahmed Mohamed Galal Osman

Master of Engineering in Computer Science

Matricola: 1853277

Our Mission

Android market share reached 85.3% in 2016. Android has been more and more indispensable with its open source character and advantages of free in our daily life. However, the number of malicious software is also growing rapidly. Therefore, how to detect the Android malware with the high accurate rate is a hot issue.

In this mission we will be using the Support vector machine algorithm to create a model that can classify Drebin dataset as malware or non malware, based on the training we give to the model.

Being able to identify malware or non malware is a binary classification problem as apps are classified as either 'Malware ' or 'Non Malware ' and nothing else. Also, this is a supervised learning problem, as we will be feeding a labeled data-set into the model, that it can learn from, to make future predictions.

Data Pre-processing

we 'll implement the one_hot_encode function. The input, x, are a list of labels. Implement the function to return the list of labels as One-Hot encoded Numpy array. The possible values for labels are 0 to 9. The one-hot encoding function should return the same encoding for each value between each call to one_hot_encode.

Support Vector Machines

SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes,

The SVM classifier consists of training and testing phases. In the training phase, a model learns from sufficient number of training data containing both benign and malicious android apps. Then during testing or detection phase, the model infers whether the given test app is benign or malign using the model learnt during the training.

Results

	Accuracy	F1 score
Support vector machine	0.9	0.90

Accuracy measures how often the classifier makes the correct prediction. It's the ratio of the number of correct predictions to the total number of predictions (the number of test data points).

Precision tells us what proportion of messages we classified as spam, actually were spam. It is a ratio of true positives(words classified as spam, and which are actually spam) to all positives(all words classified as spam, irrespective of whether that was the correct classification), in other words it is the ratio of

[True Positives/(True Positives + False Positives)]

Recall(sensitivity) tells us what proportion of messages that actually were spam were classified by us as spam. It is a ratio of true positives(words

classified as spam, and which are actually spam) to all the words that were actually spam, in other words it is the ratio of

SVM pros

Accuracy is Good

Works well on smaller cleaner data-sets

It can be more efficient because it uses a subset of training points

The Code was taken from

sontung/drebin-malwares