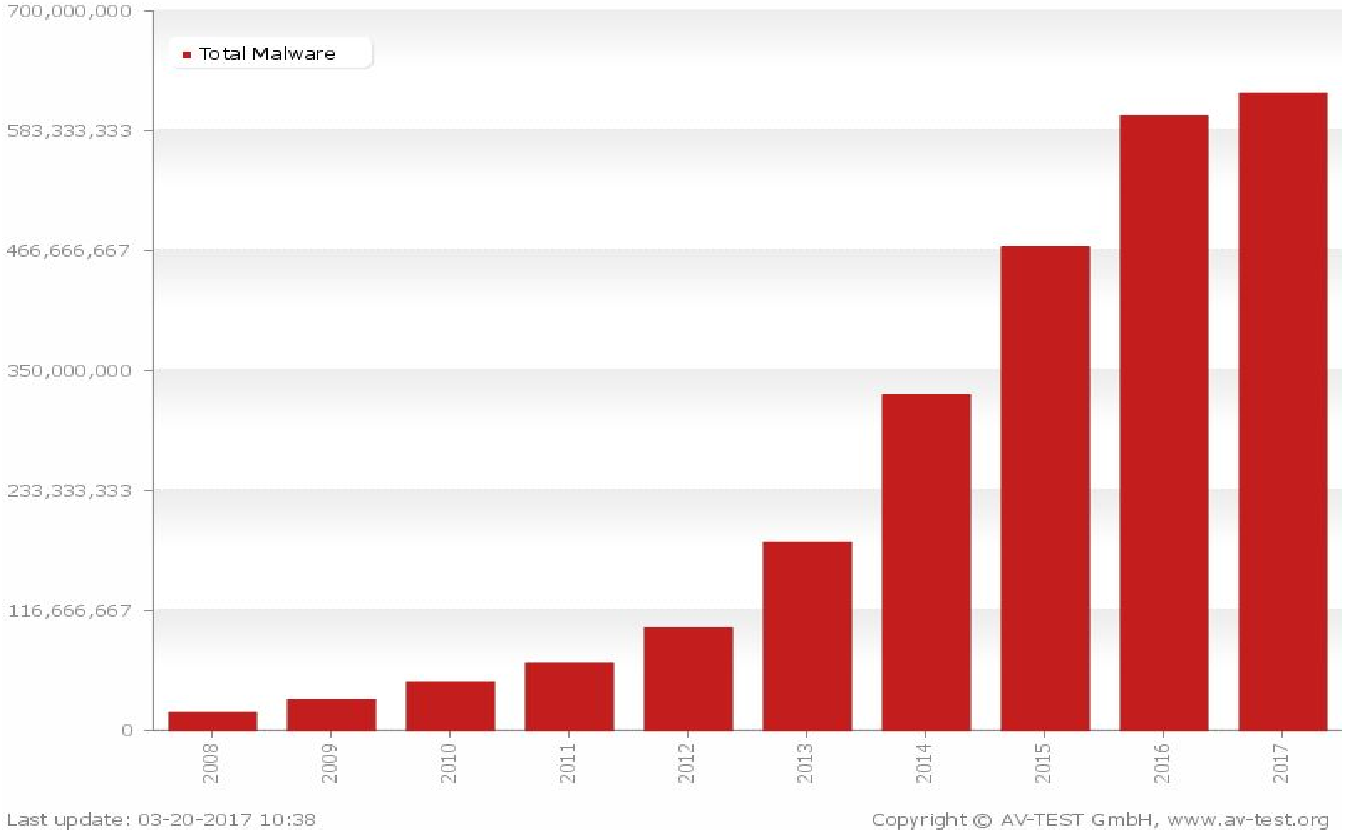
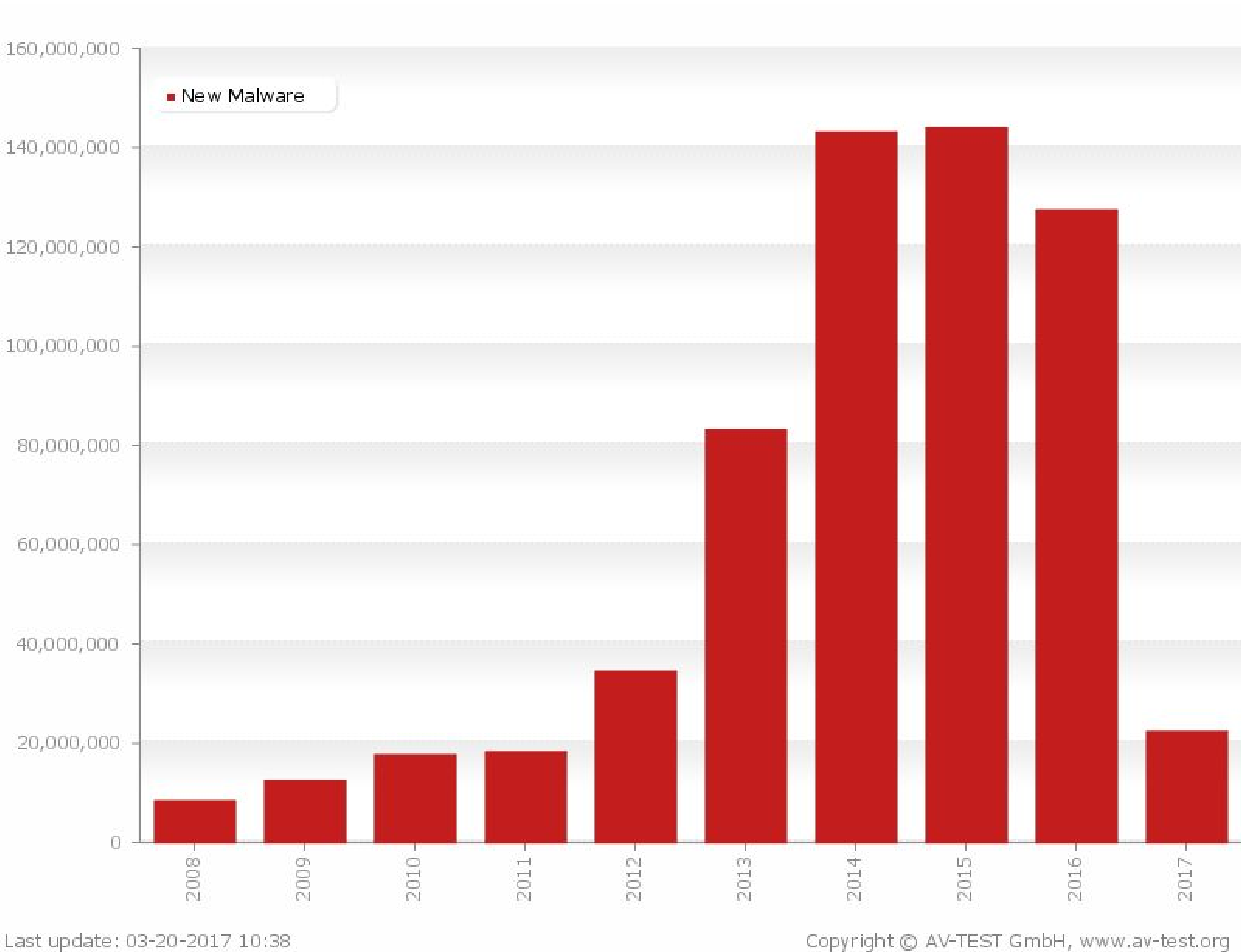
Machine Learning Nanodegree
Udacity Connect Intensive

Capstone Project
Using Machine Learning for Malware detection

Dileep Tiku

Malware-the most common threat to IT security




New Malware
160,000,000
140,000,000
120,000,000
100,000,000
80,000,000
60,000,000
40,000,000
20,000,000
0
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
Last update: 03-20-2017 10:38
Copyright © AV-TEST GmbH, www.av-test.org

Problem Statement

Develop a behavior analysis model that can identify whether a software is malware or not.

- Allow organizations to use the malware detection model, to assist in advanced threat protection not possible only by traditional methods of malware detection, which are signature based.
- Malware poses the biggest threat to security in organization.
- With the advent of Industrial internet of things besides compromising information technology assets, malware can impact operational technology which has potential of causing harm to society at large.

Datasets and Inputs

The dataset used for this project is a collection 388 malware traces

The malware traces were obtained from the CSMining website.

Data

Diagnosis	*No. of Malware Traces*	*Percent*
Infected*	320	82.47%
Benign	68	17.53%

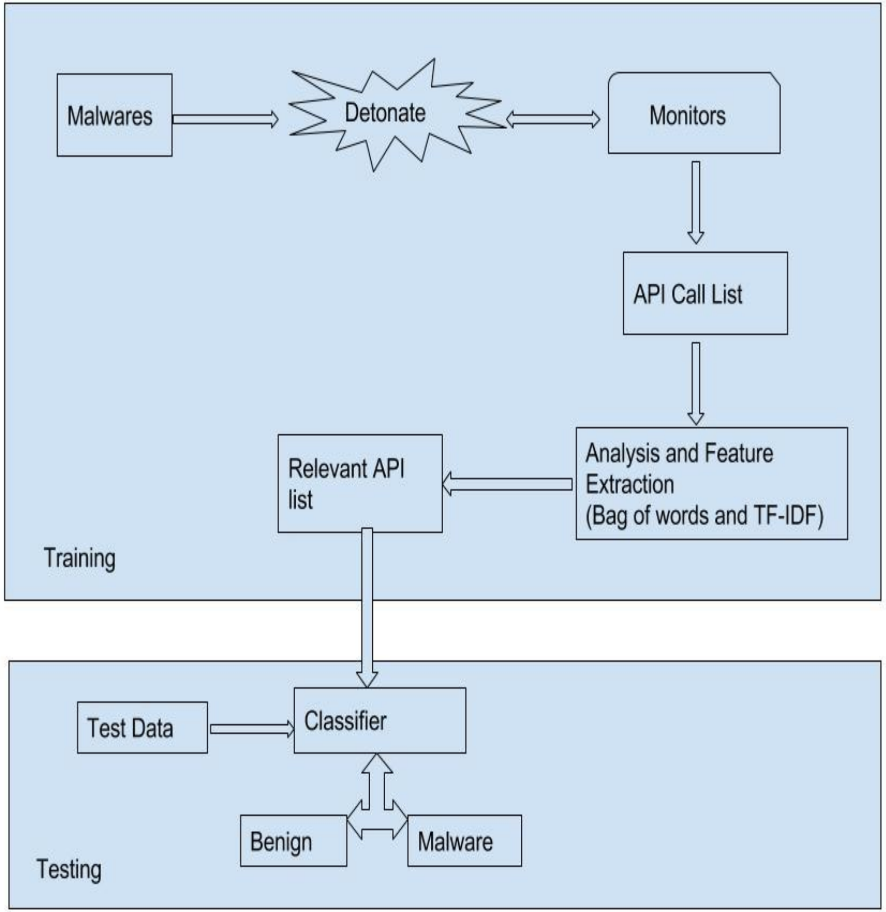
*Worms, Trojans, Viruses are labeled as infected

Utilize behavior analysis to diagnose malware

Malware is primarily diagnosed by using signatures of the known types. A classification model will be developed based on the behavior data of the software in a sandbox environment, where monitoring tools will record the API Calls made by the malware.

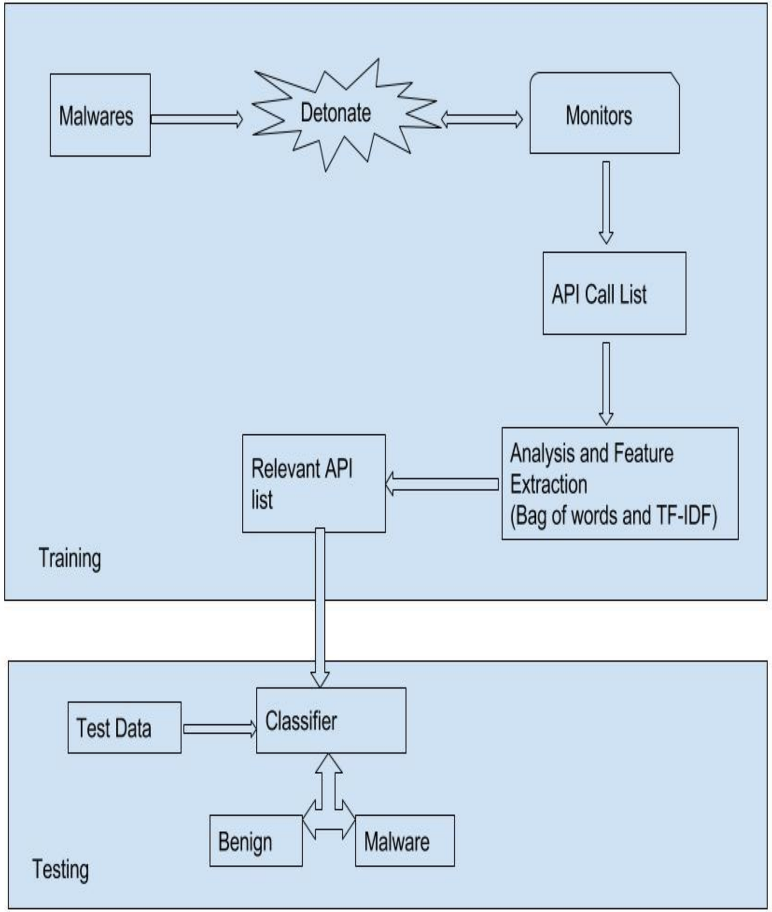
The training data set from CSMining web site contains a list of API Calls, which has been generated by executing malware and benign software in a controlled environment. The sequence of API Calls are labeled as malware or benign software.

The first step of the solution is to extract features from the list of API calls. The features are extracted by generating bag of words from the training data. Ultimately classifier like support vector machines is used to classify the software. Before selecting SVM, I worked with Multinomial NaiveBayes classifier, but the results were better with SVM.



Algorithms and Techniques

1. Extract the features from CSMining dataset.
    - The feature set will extracted from the list of API calls by using a text mining technique utilizing bag of words.
    - The technique involves computing the tf-idf (term frequency times the inverse document frequency)
2. Train a Multinomial Naive Bayes classifier with the extracted features from the list of API calls.
3. Test the classifier with the test data set and compute the accuracy.
4. Using the extracted feature set, train the SVM classifier with Stochastic Gradient Descent with loss = 'hinge' to notice any difference in test accuracy.
5. Test the new model with test set and compare the performance metrics with the results from step 3.



Benchmark: model

The benchmark model will be the detection accuracy achieved by using the Multinomial Naive Bayes classifier.

Evaluation Metrics

tp: true positive
fp: false positive
fn: false negative
tn: true negative

		Predicted Class	
		Malware	Benign
Actual Class	Malware	tp	fn
	Benign	fp	tn

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \quad \text{Precision} = \frac{tp}{tp + fp} \quad \text{Recall} = \frac{tp}{tp + fn}$$

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Type II errors are associated with false negatives. In this project it is when a software is diagnosed as benign but actually is malware. These errors are measured by recall; the more type II errors the lower the recall will be and vice versa.

Malware removal

Infected files should be quarantined to keep the IT assets safe.
Type I errors are not generally an issue
If malware is not detected, it can potentially cause huge damage.

$$\text{Recall} = \frac{tp}{tp + fn}$$

Performance

Accuracy with Mutinomial Naive Bayes = 84.6%
Accuracy with SVM classifier = 89.7%

Metrics from SGD Classifier

	precision	recall	f1-score	support
Benign	0.83	0.42	0.56	12
Malware	0.90	0.98	0.94	66
avg / total	0.89	0.90	0.88	78