Web based Malware Detection using Important Supervised Learning Techniques on Online Web Traffic

R.M. Yadav
M.A.N.I.T. Bhopal

R.K. Bhagel
M.A.N.I.T. Bhopal

ABSTRACT

Malwares on the websites can be harmful for the host machine. It may result in security breach, data loss, or denial of service at the host end. Many approaches for malware prediction have been applied in the past. Supervised machine learning approaches are popular and efficient in terms of accuracy. These techniques can be very useful for malware prediction using web traffic. Alarm for malware can be generated well before the attack and damage by simply just monitoring the web traffic. In this paper comparative analysis of supervised machine learning approaches which includes Naïve bayes, Support vector machine, PART and J48 is done. These methods are compared in terms of accuracy of prediction, false positive, false negative, true positive and true negative. This analysis is done using Weka tool.

Keywords

Web Based Malware, Supervised learning, Naive Bayes, SVM, J48, PART

1. INTRODUCTION

The term web based "malware" covers all sorts of malicious software designed to harm a computer or network via internet access. Kinds of malware are viruses, worms, spyware and Trojan horses. Web based malware has been growing in scale and complexity spurred by the unabated popularity of internet. Web based malwares are distributed over the worldwide through URLs via various web links. The reasons of growing web based malware in web world are to increase cyber crimes. The real world applications mostly effected from malware are banking, electronics communication and online shopping etc [1].

There are two basic classification techniques named supervised and unsupervised which are used to detect the malware. Most important or popular supervised techniques are K-nearest Neighbor, Linear SVM, Radial Basis Function SVM (Support Vector Machine) and Naive Bayes technique. In Supervised technique, classification defines the effect of one set of observations called inputs and another set of observations called outputs. Based on their input and output, some training is done for further classification of malware. In unsupervised classification observations are assumed for malware classified [2].

Bayesian classification provides leaning algorithm and prior knowledge and the observed data can be combined. It calculates explicit probabilities for hypothesis and provide the classification data to classify the malware. the Support Vector Machine (SVM) method to classify malicious activities by separating input in to two classes: benign and malware. This classification is used for separating hyper plane of the input data. Hyper plane defines the support vectors. J48 classification is a simple decision tree based classification approach. It is modified form of Weka tool. This algorithm generates a classification decision tree for the given data set by recursive partitioning of data. This decision tree is useful in the classification [3].

In this paper, some important supervised learning techniques like Bayesian, SVM and J48 are described for malware detection. The comparative analysis of these methods are shown and define which method is better in comparison to other method.

2. RELATED WORK

In [4] malicious web pages which containing dynamic. HTML code, can be harmful for computers are detected using machine learning. Machine learning is used to classify a web page in malicious and non malicious depending on the feature extracted in this paper. The aim of this paper is to propose a technique which is resilient to code of the web pages. In this paper Naive Bayes, SVM, Decision tree and boosted decision tree approaches are applied to classify web pages and their performances are compared in terms of accuracy, false positive, true positive and some other parameters. It is found that boosted decision tree approach gives the best classification results.

In [5] activities on web servers and systems which are connected to internet are tracked using honey pot. These web activity logs are classified as malicious and non malicious. Supervised machine learning approaches are used to classify logs into vulnerable activity and attack. in this paper 43 different features are extracted and studied. All the data collected by honey pot is having these 43 attributes. The Naive Bayes, SVM, Decision tree, PART approaches for classification are applied and data is classified. It is concluded that decision tree J48 approach and PART performs better than other in terms of accuracy, number of features used and execution time.

In [6] viruses and cyber security threats and other malwares are detected using SVM. Normally virus detection mechanism use signature based approach. In this paper supervised learning approach is used. Signature patterns are generated by machine learning and behavior detection methods. These patterns are compared in terms of accuracy.

In [7] Random forest, SVM Decision tree methods are used to classify malwares integrated with static and dynamic features. In this paper static and dynamics feature extraction techniques are used to extract features. The above mentioned approaches are applied and compared in term of accuracy and time. It is concluded that Random forest machine learning approach yields the best results.

In [8] a survey is done on various machine learning algorithms and different phases in detection of malware. The three phases discussed in paper is file representation technique, Feature selection method and classification. It is observed that each phase has significant effect on the accuracy.

3. NAIVE BAYES APPROACH

Naive Bayes is one of the most important and real practice model in bayes machine learning. Mitchell introduces bayes learning methods in detail in book [11] Michie, Spiegelhalter, et al practiced in Naive Bayes machine learning model and they compared Naive Bayes classification model with other learning algorithm, such as neural network and decision tree. Their analysis shows that the Naive Bayes's performance is as good as other models in most of cases, and is better in some cases. One typical application of Naive Bayes is text document classification. This classification approach is based on Naive Bayes which is the most effective method for text document classification currently.[12]

The processing objective of malware behavior classification is the system call oriented behavior report which is generated through behavior monitoring. Most monitoring systems provide the report in XML format and the contents of the report are the system call name, parameter list, return value and other additional information. Inspired by the application of Naive Bayes in text documentation classification, we applied Naive Bayes in our approach for malware behavior classification. The Naive Bayes algorithm that we shall present in the following general setting. Consider the instance space R consisting of all malware behavior monitoring reports. We are given training examples of some unknown target function f(r), which can take any value from the finite set C. The target function f is considered classifying unknown behavior reports as known category.

The task of our general setting is to learn from the training set to predict the target value for subsequent behavior report. There are two key issues involved in applying the Naive Bayes classifier to our behavior classification problem that first to decide how to represent the behavior monitoring result

in terms of attribute values, and second to decide how to estimate the probabilities required by the Naive Bayes model.[12]

It is a probability based classification technique. It considers all features independent of each other. It calculates probability of each feature independently for a particular class label. Mathematically it can be denoted as:

P(x/y) which denotes probability of feature x in the feature set given a class label ‘y’. Then for all the features total probability will be:

$$P(x/y) = \prod_{k=1}^{d} P\left(\frac{x_k}{y}\right)$$

Then the posterior probability of class ‘y’ given that x feature is in the feature set is given by:

$$P\left(\frac{y}{x}\right) = \frac{P\left(\frac{x}{y}\right)P(y)}{P(x)} = \frac{P\left(\frac{x}{y}\right)P(y)}{\sum_j P\left(\frac{x}{y_j}\right)P(y_j)}$$

The features for which P(y/x) is the most deciding features and can also be considered as principle components.

Since this approach is based on the probability it can be applied to a wide variety of domains and results can be used in many ways.

Naïve Bayes is used for malware prediction using web traffic data. These are the steps behind the Naïve Bayes algorithm [9]:

1. Training data set is taken as nput.
2. Features are extracted from that training data. In this paper web traffic data consists of 43 features.
3. Then from the training data for every feature Naïve bayes calculates probability that if feature has particular value then the dataset class will be malicious or not.
4. If every feature has limited possible values then above probabilities can be calculated. But if the large number of values is there for every feature, range of values can also be taken.
5. Then for every row of test data set after the training phase. On the basis of average probabilities calculated from training data decision is taken.

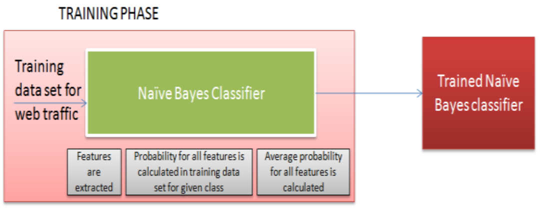
TRAINING PHASE
Training data set for web traffic
Naïve Bayes Classifier
Trained Naïve Bayes classifier
Features are extracted
Probability for all features is calculated in training data set for given class
Average probability for all features is calculated

Figure 1: Naïve Bayes Training Phase

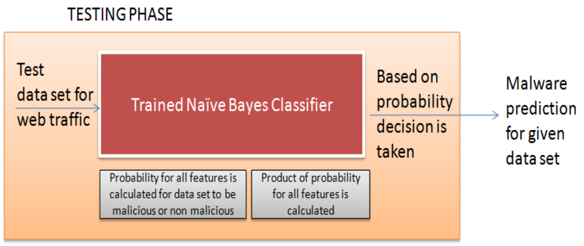
TESTING PHASE
Test data set for web traffic
Trained Naïve Bayes Classifier
Based on probability decision is taken
Malware prediction for given data set
Probability for all features is calculated for data set to be malicious or non malicious
Product of probability for all features is calculated

Figure 2: Naïve Bayes Testing Phase

4. SVM APPROACH

SVM (Support Vector Machine) is the technique for data classification. The data classification process involves training and test datasets. Each element of dataset consists of multiple features and classification attributes. The principle of SVM is to create the model for predicting classification based on the given features of current element of test dataset.

The SVM algorithm can have various kernels, but the linear, polynomial, RBF, sigmoid are basically dominates [13]. Linear SVM have performed well on massive datasets with many features [14]. In practice, the nonlinear SVM has inadequate result applying to data with more than 10000 entries. Therefore, we choose linear SVM in our research.

It is a classifier which finds a hyper plane that clearly separates the sample points of different labels. It divides such that sample points of both labels or classes on different sides of hyper plane. The hyper plane is generated such that it satisfies two constraints:

a. It should separate sample point of both labels.
b. Distance of closest sample point of both labels should be maximum.

Mathematically hyper plane is denoted as : w.x – b = 0 where denotes the dot product, w is normal vector and the parameter b/||w|| determines the offset of the hyper plane from the origin along the normal vector w. w and b should be chosen such that margin should be maximum and distance between parallel hyper planes should be maximum and should still separate the sample points of labels given. Biggest limitation of SVM is appropriate selection of kernel according to the dataset. Second speed is slow and gets even slower with size of testing and training dataset. SVM can also be used for web based malware prediction using traffic data. It involves following steps [6]:

1. Training data set is taken as input.
2. Features are extracted from that training data.
3. Classifier is generated which separates the data into malicious and non malicious data.
4. The best classifier is the one which has maximum margin and successfully separates the 2 classes.
5. Test data is given and every instance of the data is classified according to the generated classifier.

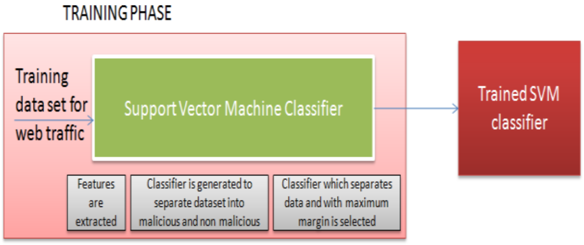
TRAINING PHASE
Training data set for web traffic
Support Vector Machine Classifier
Trained SVM classifier
Features are extracted
Classifier is generated to separate dataset into malicious and non malicious
Classifier which separates data and with maximum margin is selected

Figure 3: SVM Training Phase

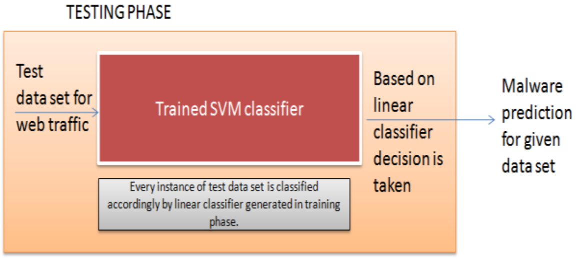
TESTING PHASE
Test data set for web traffic
Trained SVM classifier
Every instance of test data set is classified accordingly by linear classifier generated in training phase.
Based on linear classifier decision is taken
Malware prediction for given data set

Figure 4: SVM Testing Phase

5. DECISION APPROACH

This type of classifier models data with the help of a tree. Tree is having features as the internal nodes and edges indicate the values of features. And edges separated nodes based on the values. All the leaf nodes of the decision tree represents a class which is expected to be obtained if we have all the features having respective values which are in the path from the root to that class having intermediate feature nodes.

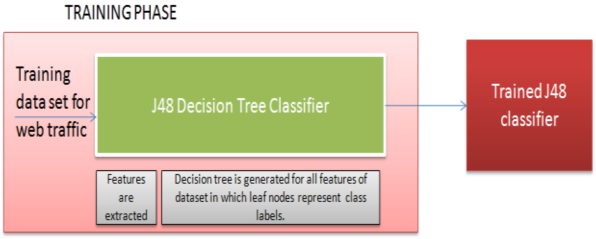
TRAINING PHASE
Training data set for web traffic
J48 Decision Tree Classifier
Trained J48 classifier
Features are extracted
Decision tree is generated for all features of dataset in which leaf nodes represent class labels.

Figure 5: J48 and PART Training Phase

Some of the most popular decision tree algorithms are ID3, C4.5, and CART. ID3 is one of the simplest decision tree approaches it uses concept of information gain as the splitting criteria. C4.5 is the evolution of ID3. It works on the principle of gain ratio. C 4.5 is a J48 algorithm. All decision tree approaches are simple to understand and easy to interpret. Most of the decision tree algorithms require features to have only discrete values.

Decision tree[5] can be used to detect malwares using following steps:

1. Training data set is taken as input.
2. Features are extracted from that training data.
3. Decision tree is generated based on the relation between the features such that leaf nodes of tree represent class labels.
4. Test data is given and every instance of the data is classified according to the generated decision tree.

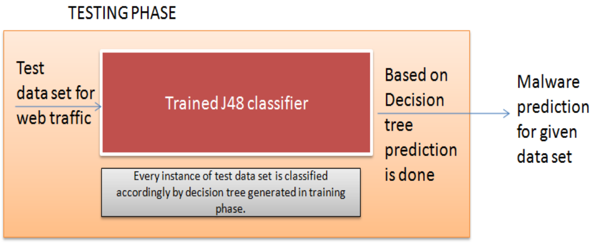
TESTING PHASE
Test data set for web traffic
Trained J48 classifier
Every instance of test data set is classified accordingly by decision tree generated in training phase.
Based on Decision tree prediction is done
Malware prediction for given data set

Figure 6: J48 and PART Testing Phase

6. RESULT ANALYSIS

(a) Naïve Bayes Based Web Malware Detection

We conduct a experiment to compare four different supervised learning techniques for malware detection with other systems: In this experiment is to compare the ability and efficiency of detecting the variants of known malware and previously unknown malware with these existing techniques.

Table 1: Naive Bayes Results

S.No.	True positive	False positive	Precision	Recall
Malicious	0.99	0.167	0.664	0.99
Non-malicious	0.833	0.01	0.996	0.833
Weighted average	0.873	0.049	0.913	0.873

(b) Support Vector Machine Based Web Malware Detection

Table 2 : SVM Results

S.No.	True positive	False positive	Precision	Recall
Malicious	0	0	0	0
Non-malicious	1	1	0.75	1
Weighted average	0.75	0.75	0.563	0.75

(c) PART Based Web Malware Detection

Table 3 : Part Results

S.No.	True positive	False positive	Precision	Recall
Malicious	1	0	1	1
Non-malicious	1	0	1	1
Weighted average	1	0	1	1

This experiment is conducted under the environment of Windows 7 operating system plus Intel i5 2.20 GHz and 4GB of RAM. All the results are evaluated on National Vulnerability database[10] Web 2.0 I & Web 2.0 II. These results are shown in table1,table2,table3 and table4.

(d) J48 Based Web Malware Detection

Table 4 : J48 Results

S.No.	True positive	False positive	Precision	Recall
Malicious	1	0	1	1
Non-malicious	1	0	1	1
Weighted average	1	0	1	1

Table 5 : Web Malware Accuracy Tables

TECHNIQUE	ACCURACY (%)
NAÏVE BAYES	87.25
SVM	75
PART	99.9
J48	99.9

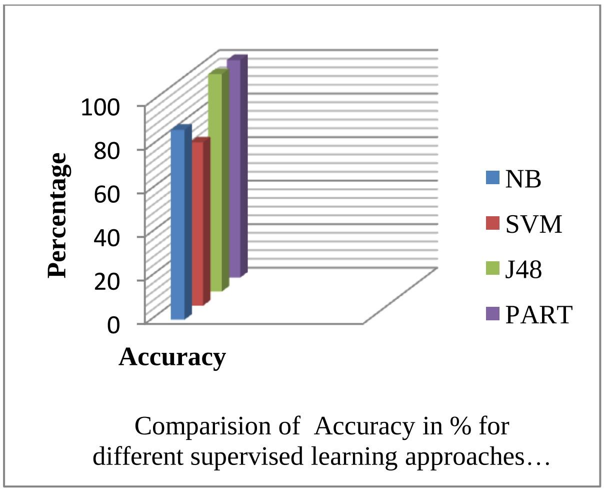

Figure 7: Comparison Bar Graph of Web Based Malware Detectors Accuracy

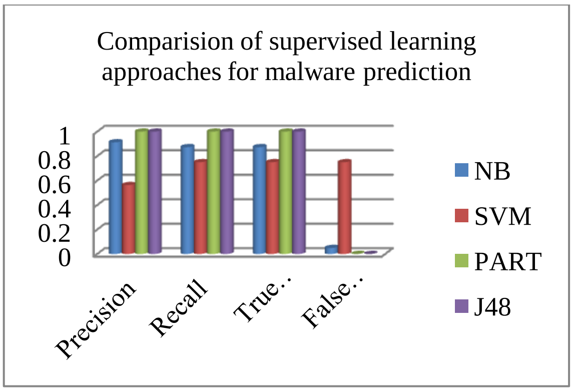

Figure 8: Comparison Bar Graph of precision, recall, true positive and false positive

7. CONCLUSION

In this paper supervised machine learning approaches are applied on web traffic data for malware prediction. Data sets which are analyzed here contain 43 features. Web traffic data set are divided in to training data and test data for malware prediction. Classifiers are training using complete feature set and then testing is done using Weka tool. It is found that the probabilistic technique like Naïve Bayes has accuracy of 87.25 %, linear classification using support vector machine has only 75 % accuracy. But the decision tree approach J48 has 99.9 % accuracy and similar is that of PART which is a rule based supervised learning approach.

8. REFERENCES

[1] M. Christodorescu, S. Jha, and C.Kruegel, "Mining specifications of malicious behavior" In Proceedings of ESEC/FSE07, pages 5-14, 2007.

[2] J. Kolter and M. Maloof, "Learning to detect malicious executables in the wild" In Proceedings of KDD'04, 2004.

[3] M. Schultz, E. Eskin, and E. Zadok "Data mining methods for detection of new malicious executables" In Security and Privacy, 2001. Proceedings. 2001 IEEE Symposium on 14-16 May, pages 38-49, 2001.

[4] Yung-Tsung Hou, yimeng Chang, Tsuhan Chen, Chi-Sung Laih, Chai-Mei Chen, "Malicious web content detection by machine learning" on Expert Systems with Applications 37 (2010) page 55-60.

[5] Katerina Goseva-Popstojanova, Goce Anastasovski, Ana Dimitrijevikj, Risto Pantev, Brandon Miller,"Characterization and classification of malicious Web traffic" in Computer and Network Security 42 (2014) page92-115.

[6] PingWang, Yu-ShihWang, "Malware behavioral detection and vaccine development by using a support vector model classifier" in Journal of Computer and System Sciences 81 (2015) page 1012–1026.

[7] Rafiqul Islam, RonghuaTian, Lynn M.Batten, Steve Versteeg ,"Classification of malware based on integrated static and dynamic features" in Journal of Network and Computer Applications 36 (2013) page 646–656.

[8] Asaf Shabtai, Robert Moskovitch, Yuval Elovici, Chanan Glezer,"Detection of malicious code by applying machine learning classifiers on static features: A state-of-

the-art survey" in information security technical report 14 (2009) page 16 – 29.

[9] Chang-Hwan Lee," A gradient approach for value weighted classification learning in naïve Bayes" Knowledge-Based Systems 85 (2015) page 71–79.

[10] National Vulerabilty 2013 database http://nvd.nist.gov/.

[11] Tom Michael Mitchell, "Machine Learning 1 Edition", McGraw Hill. New York, March, 1997: 112-143.

[12] Lewis David Dolan, "Representation and Learning in Information Retrieval", Ph.D. Thsis, Department of Computer and Information Science, University of Massachusetts, COINS Technical Report 91-93, 1991.

[13] CoW. Hsu, C-C Chang, and C-l. Lin, "A Practical Guide to Support Vector Classification," Taipei, Apr. 2010.

[14] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," Journal of Machine Learning Research, vol. 9,pp. 1871-1874, 2008.