# Machine learning algorithm to detect unknown malicious codes

Simon Khan, Uttam Majumder

Simon Khan, Uttam Majumder, "Machine learning algorithm to detect unknown malicious codes," Proc. SPIE 10185, Cyber Sensing 2017, 101850D (1 May 2017); doi: 10.1117/12.2267006

SPIE.

Event: SPIE Defense + Security, 2017, Anaheim, California, United States

# Machine Learning Algorithm to Detect Malicious Codes

Simon Khan*[a], Uttam Majumder[a]
[a]Air Force Research Laboratory (AFRL), 525 Brooks Rd, Rome, NY, USA 13441

## ABSTRACT

Modern computer and communication infrastructures are highly vulnerable to malicious codes and activities. There are many different ways malicious codes such as viruses, worms, Trojan horses etc. can damage a multitude of services, computers, financial structures, cyber infrastructure and data privacy. Signature based detection are more prevalent in preventing these types of attacks than machine learning detection. Anti-virus vendors are facing huge quantities (thousands) of suspicious files every day. These files are collected from various sources including dedicated honeypots, third party providers and files reported by customers either automatically or explicitly. The large number of files makes efficient and effective inspection of codes particularly challenging. In this paper, we propose a two part hybrid detection system that is in two parts. One part is a misuse detection system and the second part is an anomaly detection system. Misuse dependent detection is based on a random forest classifier and anomaly based detection is based on a single class SVM with bagging technique. We depart from the usual approach by using Correlation Feature Selection algorithm (CFS) for feature selection. Our experiment shows that our hybrid detection system outperforms the existing hybrid systems with other machine learning algorithms.

**Keywords:** Support Vector Machine (SVM), malicious code, machine learning, performance measurement

## 1. INTRODUCTION

There are many different types of malicious codes that are combined into some categories such as rootkits, worms, viruses, Trojans horses etc. Malicious codes are becoming complex enough not to be detected as some malicious codes can act as worms when spreading through a network; can act as viruses when attacking a computer; may also display botnet behaviors while communicating with command and control servers and may also control servers and hide from intrusion detection system (IDS) using rootkit behavior. The problem is to detect and destroy unknown malicious code that poses a significant threat for the host and network systems. The current security software generates false alarms without knowing the actual pattern of the malicious code. Machine learning offers tremendous potential to aid in unknown malicious code intrusion recognition, but there are still serious disjoints between many machine learning based intrusion detection "solutions" presented by the research community and those fielded in IDS software. Many factors favor the spread of malware such as the growth of the internet, the advent of social networks, the vulgarization of smart devices and the increasing use of storage media etc. Malware causes much damage to computer and network systems and around 390000 new malicious programs appear every day [1]. Moreover, a recent report is published that malicious codes are spreading more than the actual software developing code [2]. Recent research has proposed methods for detecting unknown malicious code using machine learning techniques. Given a training set of malicious and benign code, a classifier is trained to identify and classify unknown malicious codes [3].

## 2. THREAT OF MALICIOUS CODES IN CYBER SECURITY

The threat of malicious code for cyber security is enormous and increasing rapidly. As the anti-virus software is getting smarter to protect host systems from malicious codes, hackers are also innovating new ways to break into a system and create havoc. Under such conditions, vulnerabilities in cyberinfrastructure can be attacked in many ways. Therefore, cyberattacks can be evaluated from an attacker's perspective or from the victim's perspective. There are many types of adversarial agents such as foreign nations, terrorists, black hat and disgruntled employees who can compromise computer system via networks. For example, hackers can access personal computers to perform, hack, conspire, collect, recruit, wiretap to destroy and damage their adversaries. If there is no up-to-date cyber security, adversarial agents such as rogue governments and terrorist organizations can send cyberattacks to paralyze and cripple nation's cyber defense. Malicious codes can be generated by adversarial agents and pose threats to multi-national companies. Similarly, private

Cyber Sensing 2017, edited by Igor V. Ternovskiy, Peter Chin, Proc. of SPIE Vol. 10185,
101850D · © 2017 SPIE · CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2267006

organizations such as banks must protect confidential information of their customers. The same thing is applicable for the pharmaceutical industry, where any private information disclosure can cause a company lot of money and provide an upper hand to its competitors. Next, we take a look at the cyber security in terms of victims. A cyber threat may result in the loss of or damage to cyber components or physical resources. Most cyber threats are categorized into one of three groups according to the intruder's purpose in stealing confidential information, manipulating the components of cyberinfrastructure, and/or denying the functions of the infrastructure [3]. It is obvious that malicious code poses a significant threat and challenge in the cyber security and it becomes more difficult when these codes are unknown. This paper explains some of the machine learning algorithms and discusses a very promising approach to overcome such problems.

## 3. PREVIOUS RESEARCH ON MALICIOUS CODE DETECTION USING MACHINE LEARNING ALGORITHMS

Malicious code detection using machine learning is popular; however, previous attempts to build a machine learning malware classifier has had mixed results. Much research had focused on using n-gram features derived from a file's binary code. For instance, applied machine learning to n-grams of malicious and benign software and their model detected 98 percent of malware while only incorrectly guessing 5 percent of the benign software [4]. Malicious samples were identified correctly 74 percent of 1000 as malicious while correctly identifying all of 1000 benign software samples [5]. There are also several approaches for static malware detection using n-gram techniques proposed an n-grams-based signature method to detect computer viruses [6]. Data mining has already been used for detecting and protecting against malicious codes. A recent survey on intrusion detection systems summarizes recently proposed applications of data mining for recognizing malicious codes in single computers and computer networks [7]. There is a proposed framework consisting of data mining algorithms for the extraction anomalies of user normal behavior for use in anomaly detection, in which a normal behavior is learned and any abnormal activity is considered intrusive [8]. Another approach is to learn from metadata, mostly contained in the headers of the executable files, specifically the Windows Portable Executable 32-bit (PE32) file format. It is observed executable file metadata is highly discriminative between malware and benign software. Executable files with n-gram which, they reduced the features using fisher score, and trained the result with Bayesian network, artificial neural networks, and decision trees [9]. Utilization of Support Vector Machine (SVM) to determine the ability of n-grams was performed to correctly predict the presence of malware. They found that an n-gram size n=3 and n=4 presented the best results [10]. Investigation of the effectiveness of Opcode n-grams for detection of multifamily android malware was performed and their experiments on large data set show that a perfect detection rate was achieved for more than one malware family [11]. An intrusion detection system that uses a combination of tree classifiers and clustering algorithms can detect possible intrusions and anomalies [12].

## 4. MACHINE LEARNING ALGORITHMS

Machine learning algorithms compute and predict certain models based on sample data. Learning models use statistical functions or rules to describe the dependences among data and casualties and correlations between input and output. There are many machine learning algorithms that are out in the research areas such as artificial neural network (ANN), Support Vector Machines (SVM), Decision Tree (DT), Bayesian Network (BN) and Hidden Markov Model (HMM). We will describe these algorithms and determine the best solutions for our problem.

**Artificial Neural Network (ANN):** An ANN is a machine learning algorithmic model that takes input variables and matches with output variables through a nonlinear processing in a connected group of artificial neurons that are called "hidden" units. Efforts are made to determine the weights to reduce the classification error when utilized by ANN as a supervised machine learning model. It is common to many learning models to use least mean-square convergence. "The objective of ANN is to minimize the errors between the ground truth and the expected output $f(X:W)$ of ANN as $E(X) = (f(X:W)\text{-}Y^2)$. The behavior of an ANN depends on both the weights and the transfer function $T_f$, which are specified for the connections between neurons. For example, the net activation at the $j^{th}$ neuron of layer 1 can be presented as

$$Y^1{}_i = T_f\,(\sum x_{i.}.w^1{}_{ji}\,) \quad (1)$$

Where $w$ is a hidden layer, $x$ is output and $Y$ based on $w$, x and considered as an input. Subsequently, the net activation at the $k^{th}$ neuron of layer 2 can be presented as

$$Y^2{}_j = T_f\,(\sum y_{j.}.w^2{}_{kj}\,) \quad (2)$$

This transfer function typically falls into one of three categories: linear (or ramp), threshold or sigmoid. ANN methods perform well for classifying or predicting latent variables that are difficult to measure and solving nonlinear classification problem and are insensitive outliers [3]".

**Support Vector Machines (SVM)**: In machine learning, SVM is a supervised machine learning algorithm that uses data for classification and regression analysis. SVM algorithm builds a model that assigns new examples to one type or the other. Provided with some data points *x* in an n dimensional feature space, SVM separates these data points with an n-1 dimensional hyperplane. "In SVM, the objective is to classify the data points with the hyperplane that has the maximum distance to the nearest data point on each side. Subsequently, such a linear classifier is also called the maximum margin classifier. Any hyperplane can be written as the set of points X satisfying $w^t x+b=0$, where the vector *w* is a normal vector perpendicular to the hyperplane and b is the offset of the hyperplane $w^t x+b=0$, from the original point along the direction of $w^t$. Linear SVM is solved formulating the quadratic optimization problem as follows [3]":

$$argmin_{w,b}\ (1/2||w|^2) \tag{3}$$

$$\text{s.t. } y(w^t x+b=0)\geq 1 \tag{4}$$

**Decision Trees (DT):** A decision tree is a tree like architecture that has nodes, branches and leaves. Leaves represent decision and branches represent the path to get to decisions. Measurement to perform tree classification of an input vector is achieved by traversing the tree from root to leaf. Decision tree depends on if-then nodes, but requires no parameters or metrics. It is simple and easily understanding behavior help able to solve multi-variable attributes. Decision trees can also manage mission values or noise data. However, they cannot guarantee the optimal accuracy than the other machine learning methods can [3].

**Bayesian Network (BN):** The BN is known as a belief network uses factored joint probability distribution in a graphical model for decisions about uncertain variables [3]. The BN classifier is based on the Bayes rule that gives a hypothesis H of classes and data x, we have, then

$$P(H|x) = \frac{P(x|H)P(H)}{P(x)} \tag{5}$$

Where

P(H) denotes prior probability each class without information about a variable x
P(H|x) denotes posterior probability of variable x over the possible classes
P(x|H) denotes the conditional probability of x given likelihood H

**Hidden Markov Model (HMM)**: Previously, we discussed machine-learning methods for data sets that consist of independent and identically distributed (IID) samples from sample space. If data is sequential learning problems such as speech recognition, HMM performs better in supervised learning of sequential patterns. "Each node represents a random variable with the hidden state $x_t$ has a probability distribution over the observed samples, $y_t$ at time t. Statistically, HMM is based on the Markov Property that the current true state $x_t$ is conditioned only on the value of the hidden variable $x_{t-1}$ but is independent of the past and future states [3]".

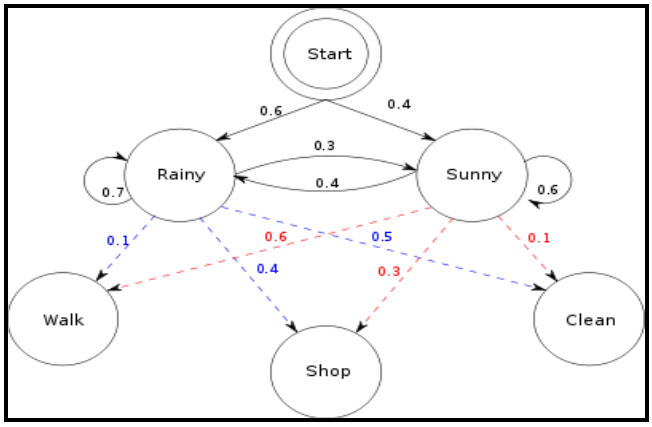


Figure 1: Model of Hidden Markov Model based on weight association [20]

## 5. APPLYING MACHINE LEARNING ALGORITHMS INTO HYBRID DETECTION SYSTEM

In a hybrid intrusion detection system, misuse and anomaly detection subsystems have a similar workflow. Both methods include five steps: data collection, data preprocessing, normal/anomalous behavior learning, identification of malicious/normal behaviors using detection techniques and decisions. Hence similar procedures are included in data collection and preprocessing, such as normalization and feature selection. Machine learning algorithms also play vital roles in building anomalous/normal profiles and intrusion detection. When a using hybrid detection system, a decision about one suspicious event is decided using combined systems. The objective is to detect whatever is missed by misuse machine detections system. By having a hybrid system does not guarantee of a robust malicious code detection system, however, it depends on effective incorporation of these subsystems. Given a set of cyber audit data $X$ as input, misuse detection system outputs three data sets $M_i$, $M_u \cap A_n$ and $M_u \cap$. $A_u$ and $A_n$ denotes the subset as unknown and normal by the anomaly detection system. We obtain the following property set $M_u \cap A_n$. The unknown audit data $M_u$ will be put to the anomaly detection system to detect the malicious data set. A misuse detection system can improve the accuracy of a framework, because most misuse detection systems produce high detection rate and low false-positive rate [3].

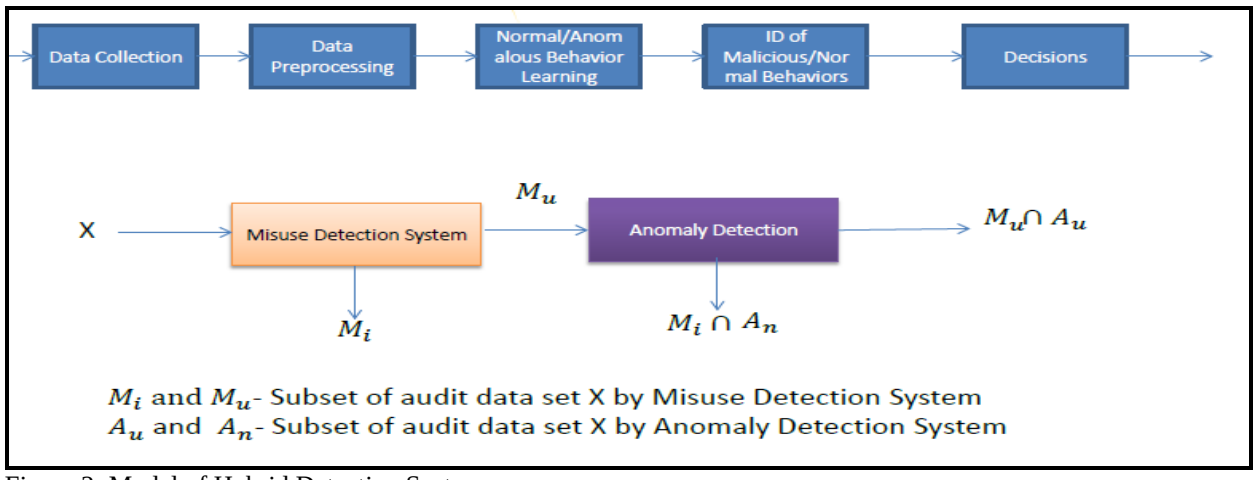

Figure 2: Model of Hybrid Detection System

## 6. EXPERIMENT

### 6.1 WEKA

We used Weka, a data mining tool, to obtain our results based on machine learning algorithm, which has different types of machine learning algorithms for data mining tasks. One can apply datasets directly into Weka or apply JAVA code manually to get results for the experiments. Weka contains tools for data pre-processing which comes with feature selection and normalization, create models, learn behavior and identify malicious codes. It is also well-suited for developing new machine learning schemes. We ran auto-Weka that conducted all the machine learning algorithms and provided a best result based on accuracy. Based on the results from auto-Weka, the misuse detection is based on random forests classifier and the anomaly detection is built on bagging technique with ensemble of one-class support vector machine classifiers. Data preprocessing is done using automatic feature selection and normalization [13].

### 6.2 NORMALIZATION

In public, some features of the data sets are normalized and others are not. Normalization of data is a crucial pre-processing step. Data normalization takes places by discretizing values of each attribute to put them in certain range so that they do not dominate each other. For our system, we used min-max based linear data normalization technique. The formula for data normalization is :

$$X' = \frac{X - X'_{min}}{X_{max} - X_{min}} \tag{6}$$

$X$ and $X'$ are values to be normalized and the normalized attribute value respectively and $X_{min}$ and $X_{max}$ are possible min and max values for attribute X before normalization [13].

### 6.3 FEATURE SELECTION

There can be multiple features but our goal is to pick the feature selection that will provide the best accuracy, In our experiment we used CFS which measures selection for classification tasks in machine learning can be accomplished based on correlation between features and that such a feature selection procedure can be beneficial to common machine learning algorithm. CFS is a simple filter algorithm that ranks feature subsets according to a correlation based on heuristic evaluation function. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class but uncorrelated with each other. Irrelevant features should be ignored because they will have low correlation with the class. The acceptance of a feature will depend on the extent, which it predicts classes in areas of the instance space not already predicted by other features [19]. CFS's feature subset evaluation function is:

$$Merit\, s_k = \frac{\overline{kr_{Cf}}}{\sqrt{k + k(k-1)\overline{kr_{ff}}}} \tag{7}$$

Here, $\overline{r_{Cf}}$ is the average value of all feature-classification correlations and $\overline{r_{ff}}$ is the average value of all feature-feature correlations. The CFS criterion is defined as:

$$CFS = (max s_k) \frac{r_{Cf_1} + r_{Cf_2} + \cdots + r_{Cf_k}}{\sqrt{k + 2\,(r_{f_1f_2} + \cdots + r_{f_if_j} + \cdots + r_{f_kf_1})}} \tag{8}$$

### 6.4 SETUP

As soon as the relevant features are identified, the feature selection model will forward the data with the selected features to the misuse intrusion detection system. It is implemented using random forests classifier, which is a kind of decision tree. The proposed system is evaluated using the publicly available NSL-KDD intrusion detection dataset, which is an enhance version of the KDD99 intrusion detection data set. The KDD99 dataset is the only well-known and publicly available dataset in the area of intrusion detection. This data set has 41 features and one class attribute. The training data contains 24 types of attacks and the testing data contains an extra 14 types of attacks [13]. The random forests classifiers operate by constructing multitude of decision trees at training time and excluding well-known attribute from being reprocessed by the subsequent one-class SVM based on anomaly detection. Only patterns which are classified as normal by the random forests classifier are forwarded to anomaly detection module for final decision. The anomaly detection unit also goes thru CFS algorithm for feature selection and normalization process. "The anomaly detector is built using only normal training instances, which are also classified by the misuse detection as normal. The anomaly detection will block detected attacks which are considered as normal traffic by the misuse detector. The signature of these attacks will be used for updating the original training data [13]".

## 7. RESULTS AND PERFORMANCE MEASUREMENT

Throughout the experiment, we used Weka 3.9.1 for our testing. We divided the NSL-KDD data set into two categories: two third of the data as training set and one third as a test set. Then the proposed data set was run through the CSF algorithm, which provided the selected data set with attributes below:

Table 1: Experimental data set attributes

| Data | Normal | Anomaly |
|---|---|---|
| Training Data | 67,343 | 58,630 |
| Test Data | 95151 | 12663 |

After applying a machine learning algorithm to the NSL-KDD datasets, accuracy rate for misuse detection is 99.4 % and anomaly detection rate is 100 %. 100 % accuracy is highly improbable but possible in our case. We used a post misuse based data set and made all the attributes to "NORMAL" before sending through anomaly based system as data sets. We change the attributes manually to observe if the anomaly based detection system flags any abnormality within its normal values but it did not. Our conjecture is based on the above result that single classifier SVM has been able to learn using 10 fold cross-validations and predict accordingly. Another reason of 100 % accuracy rate that the attacks are not dynamic, thus they are not adaptive. It is highly plausible by integrating CSF algorithm during preprocessing enabled to have such high accuracy rate. Performance comparison for the full test data is added by our data according to Tesfahun et al [13].

Table 2: Comparison table

| Model | Attack Detection Rate | False Positive Rate |
|---|---|---|
| Naïve Bayes | 0.588 | 0.063 |
| J48 decision tree | 0.647 | 0.029 |
| **Random Forest using CFS** | **0.994** | **0.006** |
| SVM | 0.591 | 0.068 |
| RF_1_SVM | 0.9213 | 0.0642 |
| **LibSVM using CFS** | **100** | **0** |

## 8. CONCLUSION

In this paper, a hybrid detection system is proposed by combining both a misuse and an anomaly intrusion detection system. The random classifier was used to detect previously known attacks and the anomaly detection ensemble of one-class SVM was built. The result is promising in terms of hybrid detection. The false detection rate was low for random forest classifier and zero for one class SVM. We compared our method with some well-known methods and the proposed system can be approximately 31 % better to detect unknown malicious codes than predecessors.

Though the proposed system takes new data and updates with new attack types, it is not adaptive. Future work will focus on system adaptation to cope with dynamic attacks. We also would like to insert "anomaly labeled attributes" data set after it is processed through misuse detection system to observe the hybrid detection system's behavior with the current accuracy. We also want to create a system to preprocess datasets into many classifiers (which performed badly in the first iteration due to benchmarking process) and preprocess again through an ANN. Finally, a prototype based on a hybrid detection system might be useful to the DoD community for research purposes.

## ACKNOWLEDGMENT

We are grateful to Rajesh Kumar for his help during the design, implementation, and performance evaluation of the proposed methodologies. Rajesh Kumar is currently a Ph.D. candidate in Computer and Information Sciences and Engineering (CISE) at Syracuse University, NY. His research interests include behavioral biometrics, authentication, identification, anomaly detection, and wireless sensor networks. He has published research articles in several transactions and conferences including ACM TOPS, ACM CCS, IEEE CVPR, IEEE BTAS, and Springer-WPC.

## REFERENCES


[1] Quinlan, J.R., [C4.5: programs for machine learning], Morgan Kaufman Publishers, San Francisco, CA, USA (1993).

[2] Maskovitch et al., "Host Based Intrusion Detection using Machine Learning," Proc IEEE ISI (2007).

[3] Dua, S. and Du.Xian., [Data Mining and Machine Learning in Cybersecurity], CRC Press, Florida, 10-150 (2011).

[4] Kolter. J.Z. and Maloof.A.M ., "Learning to detect malicious executables in the wild," Proc ACM SIGKDD, 470-478 (2004).

[5] Santos et al., "N-grams-based file signatures for malware detection," S3Lab, Deusto Technological Found., 2009.

[6] Assaleh et al., "N-grams-based detection of new malicious code." Proc COMPSAC, 41-42 (2004).

[7] Kabriri, P., Ghorbani, A.A., "Research on intrusion detection and response A survey," International Journal of Network Security, vol. 1(2), pp. 84-102.

[8] Lee at,. "A data mining framework for building intrusion detection models. Proc IEEE (1999).

[9] Elovici et al., "Applying machine learning techniques for detection of malicious code in network traffic." KI 2007: Advances in Artificial Intelligence, Vol 4667, pp.44-50, 2007.

[10] O'Kane et al., "N-gram density based malware detection," In World Symposium on Computer Applications & Research (WSCAR), pp 1-6, (2014).

[11] G.Canfora et al., "Effectiveness of Opcode ngrams for Detection of Multi Family Android Malware," Proc ARES (2015)

[12] Xiang C. and Lim.M.S., "Design of Multiple-Level Hybrid Classifier for Intrusion Detection System," Proc IEEE MLSP (2005).

[13] Tesfahun. A and Bhaskari.L.,"Effective Hybrid Intrusion Detection System," Proc.IEEE TDSC (2015).

[14] Qin.M. and K.Hwange, "Frequent Episode Rules for Internet Traffic Analysis and Anomaly Detection," Proc.IEEE NAC (2004).

[15] Boujnouni et al., "New Malware Detection Framework based on N-grams and Support Vector Domain Description," Proc IEEE IAS (2015).

[16] Rudd et al., "A Survey of Stealth Malware Attacks, Mitigation Measures and Step Toward Autonomous Open World Solutions," Proc IEEE CST (2016).

[17] Jung et al., "Deep Learning for Zero-day Flash Malware Detection." Proc IEEE SSP (2015).

[18] Hwang et al., "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes," Proc IEEE TDSC (2007).

[19] Hall,M.,"Correlation Feature Selection Algorithm,"1999 <http://www.cs.waikato.ac.nz/~mhall/thesis.pdf> (03 March 2017).

[20] "Hidden Markov Mode".,< https://en.wikipedia.org/wiki/Hidden_Markov_model> (04 March 2017).


## DISCLAIMER

The views expressed in this article are those of the authors and do not reflect official policy of the United States Air Force, Department of Defense or the U.S. Government.