International Journal of Innovative Research in Information Security (IJIRIS) ISSN: 2349-7017
Issue 12, Volume 4 (December 2017) www.ijiris.com

USING BLACK-LIST AND WHITE-LIST TECHNIQUE TO DETECT MALICIOUS URLS

HUSSAIN AHMED TARIQ
School of Computer Science & Engineering,
University of Electronic Science and Technology of China. Sichuan, China
Hussain_aahmed@yahoo.com;
WONG YANG
School of Computer Science & Engineering,
University of Electronic Science and Technology of China. Sichuan, China
IMRAN HAMEED
School of Computer Science & Engineering,
University of Electronic Science and Technology of China. Sichuan, China
BILAL AHMED
School of Computer Science & Engineering,
University of Electronic Science and Technology of China. Sichuan, China
RIAZ ULLAH KHAN
School of Computer Science & Engineering,
University of Electronic Science and Technology of China. Sichuan, China
rerukhan@outlook.com;

Manuscript History
Number: **IJIRIS/RS/Vol.04/Issue12/DCIS10081**
DOI: 10.26562/IJIRIS.2017.DCIS10081
Received: 13, November 2017
Final Correction: 27, November 2017
Final Accepted: 04, December 2017
Published: December 2017
Citation: TARIQ, H. A., YANG, W., HAMEED, I., AHMED, B. & KHAN, R. U. (2017). USING BLACK-LIST AND WHITE-LIST TECHNIQUE TO DETECT MALICIOUS URLS. IJIRIS:: International Journal of Innovative Research Journal in Information Security, Volume IV, 01-07. doi: 10.26562/IJIRIS.2017.DCIS10081
Editor: Dr.A.Arul L.S, Chief Editor, IJIRIS, AM Publications, India

Abstract: Malicious URLs are harmful to every aspect of computer users. Detecting of the malicious URL is very important. Currently, detection of malicious web pages techniques includes black-list and white-list methodology and machine learning classification algorithms are used. However, the black-list and white-list technology is useless if a particular URL is not in list. In this paper, we propose a multi-layer model for detecting malicious URL. The filter can directly determine the URL by training the threshold of each layer filter when it reaches the threshold. Otherwise, the filter leaves the URL to next layer. We also used an example to verify that the model can improve the accuracy of URL detection.
Keywords: Malicious URL; Black-list and White-list Technology; Machine Learning; Multi-layer Filtering Model;

I. INTRODUCTION

In recent years, the Internet has been playing a bigger and bigger role in people's work and life. Currently, we observed that not every website is user-friendly and profitable. More and more malicious websites began to appear and these malicious websites end angering all aspects of the user. This can lead the user towards economic classes, even some can create confusion over the management of the country. Detecting and stopping malicious websites has become an important control measure to avoid the risk of information security [10],[8], [7].Generally, the only entrance to the website is URL, which can be malicious URL.

At this level of entrance, the identification of URL is the best solution to avoid information loss. So the malicious URL identification has always been a hot area of information security. Spams, malicious web pages and URLs that redirect or mislead the legitimate users to malware, scams, or adult content. It is perhaps correlated with the use of the internet. To identify malicious URLs, ML-based classifiers draw features from web pages content (lexical, visual, etc.), URL lexical features, redirect paths, host-based features, or some combinations of them. Such classifiers usually act in conjunction with knowledge bases which are usually in-browser URL BLs or from web service providers. If the classifier is fed with URL-based features, it is common to set a URL aggregator as a per-process or before extracting features. Mostly using supervised learning paradigm, Naïve Bayes [1], [13], Support Vector Machine with different kernels [9], Logistic Regression are popular Machine Learning classifiers for filtering spam and phishing [6]. Mean while, GT- based learning to deal with active attackers is also evaluated in spam filtering. In this paper we solve this problem using multi-layer filtering model to do each classification which are able to handle their own relatively good data.

II. PROBLEM DESCRIPTION

Spams, malicious webpage's and URLs that redirect or mislead un-suspecting users to malware, scams, or adult content are perhaps as old as civilian use of the Internet. To identify malicious URLs, ML based classifiers draw features from web page content (lexical, visual, etc.), URL lexical features, redirect paths, host-based features, or some combinations of them. Such classifiers usually act in conjunction with knowledge bases which are usually in-browser URL BLs or from web service providers. If the classier is fed with URL-based features, it is common to set an URL aggregator as a preprocessor before extracting features.

Table 1: DIFFERENT MACHINE LEARNING TECHNIQUES USED FOR MALWARE DETECTION

Author	Goal	Features	System Components	ML Techniques
Ma ICML'09 [5]	Detect malicious websites by URLs in active learning	Lexical and Host-based features, no content features	Live URL feed, labelling engine, feature extractor, classifier with feedback loop	Live URL feed, labelling engine, feature extractor, classifier with feedback loop
Ma KDD'09 [6]	Detect malicious websites from URLs	Lexical and host-based features, no content features	Complementary to BLs	NB, SVM-RBF, linear SVM, L1-regularized LR
Thomas SP'11 [13]	Real-time URL spam filter	URL Lexical, content, hosting property, browser, DNS resolver, IP analysis features	Web service with URL stream, URL aggregator, feature collector and extractor, classifier, feedback loop(BL annotation training)	L1-regularized LR, LR- SGD
Lee NDSS'12 [4]	Detect malicious URLs in streams	Properties of redirect chains of URLs	URL stream data collection, feature extraction, training	L2-regularized LR
L2-regularized LR[3]	Use search engines to find other malicious webpage's	Content- and link-based features	Crawler, profile, search engine's BL, initial set of malicious pages	N-grams, term extraction
Zhao KDD'13 [17]	Online active learning for malicious URL detection	Lexical- and host-based features	Live data feed, feature collector, cost-sensitive update, active learning module, classifier	Customized online active cost-sensitive algorithm
Zhang NDSS'14 [16]	Detect malware by file relation graphs	Content features	Content filter, inverted index engine, clustering	Shingling, POS tagging (metric: Jaccard index)
Whalen AISec'14 [14]	Distributed content anomaly detection(CAD)	N-gram of payloads	Distributed models over application servers	Aggregated RF, LR, Bloom filter

Mostly using supervised learning paradigm, NB, SVM with different kernels, and LR are popular ML classifiers for filtering spam and phishing. Meanwhile, GT based learning to deal with active attackers is also evaluated in spam filtering.

In this paper we solve this problem using multi-layer filtering model is designed to make each classification, which able to handle their own relatively good data, for their own number of not good at according to not doing the processing.

III.BACKGROUND

3.1 Naive Bayesian

Naive Bayes algorithm [4][13] is a very classic machine learning classification algorithm, which is based on the conditional probability formula and the conditional independence assumption. The mathematical proof, classification accuracy, is a good learning efficiency of the current point's classifier. The main idea of Naïve Bayes classification is as follows: Suppose the data samples haven-dimensional vectors{x1,x2,...,xn},which are divided into m Classes i.e. C1, C2,...,Cm. For the sample to be classified, extract its vector X, Calculate the conditional probability of each class under the vector, and calculate the formula as: P(X) appears for the X vector in all training samples probability, which is constant for all classes, and Naïve Bayes assuming that all features are independent of each other, they can be transformed into formula:

$$P_i = P(C_i)\prod_{k=1}^{k} P(X_k \mid C_1).....2$$

Calculate the probability of each class i.e. find the maximum X for each class probability x Pi, then Naïve Bayes classifier will be assigned to this Class.

3.2 CART Decision Tree

Decision tree algorithm is based on a sample of induction learning algorithm. It uses a top-down recursive approach to the nodes of the decision tree row characteristic value comparison, and according to different values from the node down point's branch; leaf node is to learn the division of the class. CART algorithm [9], [1] is a kind of more famous decision tree. The best feature here is, it is usually able to make the training set in the child node as pure as possible. The most commonly used measure in CART algorithms is the indicator of training set purity i.e. GINI coefficient. GINI coefficient calculation as Formula (3): Where pi denotes the probability of S in category i. GINI coefficient is the greater representative of the training. The c means that all the samples belong to the same class, so in the build CART tree in the process needs to be calculated for each feature which can bring GINI gain, GINI gain can be determined by equation(4):

$$Gini = 1 - \sum_{i=1}^{c} p_i^2 .....3$$

$$Ginigain(\mathrm{S}) = \mathrm{gini}(\mathrm{S}) - \frac{N_1}{N} gini(\mathrm{S}_1) + \frac{N_2}{N} gini(\mathrm{s}\,2)).....4$$

Where *gini*(*S*) is the GINI coefficient before division, N is the total number of samples, *N*1 and *N*2 are the number of samples of the two left and right subdivisions, *gini*(*S*1) and *gini*(*S*2) are the GINI coefficients of two child nodes, respectively. When the node contains d at a records belong to the same category or category CART establishments to ps when it is independent of value.

CART building steps are:

1) To determine the current sample set to meet the termination conditions, if not then calculate the GINI gain of each feature of the current sample set.
2) Select the feature with the highest GINI gain as the segmentation feature.
3) The two samples divided by this feature are taken as new sample weight respectively.

After building the CART tree, we can classify the samples for classification: Extract the vectors of the samples to be classified and put them in the CART tree. Finally, the proportion of leaf nodes in the category, take the leaf nodes in the highest proportion of categories as the classification of the sample.

3.3 Support Vector Machines

Support Vector Machine (SVM), is a two-class classification model. The basic model is defined as the feature space on the interval. The largest linear classifier uses the learning strategy to maximize the interval. Due to SVM's solid mathematical theory, it should be used in many areas for good performance. The main idea of SVM is as follows:

$$(\mathrm{x}_i \mathrm{y}_i), \mathrm{i} = 1, 2, ..., \mathrm{n} : \mathrm{x}_i \in R^N : y_i \in \{-1, 1\}$$

The optimal classification function $f(x)=sgn((w^*)^T x+b^*))$ can be derived. Note that, for those linearly in separable sample sets, one can choose to introduce a kernel function mapping or punishment coefficient of the way to carry out training. After setting up the classification function, we can classify the sample to be classified the feature vector X of the sample to be classified is extracted and substituted into the classification function. The classification letter can be determined from the category by the number of returned value.

IV. MULTI-LAYERFILTERMODEL

Malicious URL detection [12],[11],[5] is a typical classification application scenario. The URL may be malicious URL or normal URL. The first part of several machine learning classification algorithms is very useful. A wide range of applications have also been applied to malicious URL detection scenarios.

In order to give a brief overview to the advantages of each classifier, this article designed a malicious URL Multi-layer detection model. This model is mainly composed of 4-layer classifier, where these classifiers are also called filters in this model, so the model consists of 4 Layer filter composition.

4.1 Stratified filter

1) **Black and white list filter:** The model's first-level filter is a black-and-white list filter which will be validated by recognizing the normal URLs and Malicious URLs. The normal URL addresses a restored in the white list file, while the malicious URL addresses a restored in the black list file. To detect the URL, we traverse the list of black and white to determine whether the URL in the black list or in the white list.

2) **Naïve Bayesian filter:** The second layer filter in this model is a naïve Bayes is an filter that trains the model by dividing into two main steps: By training the URL samples, we use two-dimensional arrays C1 and C2 to store the probability of each value of malicious website and normal website, such as formula (1) below:

$$c_i = \begin{pmatrix} P_{11}, & P_{12}, & \dots P_{1m} \\ \vdots & \vdots & \vdots \\ P_{n1} & P_{n2}, & \dots P_{nm} \end{pmatrix} \quad \dots\dots\dots\dots\dots 1$$

4.2 Alpha N-Bayes threshold training

Let $\alpha_{nbayes} = \max(p1/p2, p2/p1)$ (where P1 and P2 represent respectively what is calculated by the naive Bayesian formula model as malicious URL and normal URL probability value), so the size of n bayes can represent this classification judgment of the credibility. The n bayes describes that the URL belongs to one of the categories. The probability is much greater than the probability of belonging to another class. So, when n bayes arrives at certain threshold size, we can consider that URL is Naive Bayesian good data. If this threshold is set to too small, it may not be reached to the ideal classification accuracy. If this threshold is set to too large, naive shellfish the data that yeast good filters will be very small. This article discusses another group training data to train this threshold. The specific method is discussed briefly in Section 3 Assume that the appropriate training threshold is α_{nbayes}. For the upper filter down to detect URL, put it into the trained naive Bayesian model, and calculate n bayes if $\alpha_{nbayes} > \alpha_{nbayes}$, then we think the URL is a good Bayesian model of good data, otherwise, the URL cannot be considered as a good data for a naive Bayesian model, i.e. it cannot determine the nature of the URL, so record the classification results, and move to the next filter.

4.3 CART decision tree filter

The model's third-level filter is CART Decision Tree Filter.

The model is further split into two steps:

1) **Model training:** The CART tree is constructed by training samples with URLs, and the leaf nodes are decision nodes, and store the CART tree in the file system.

2) **cart threshold training:** Let $\alpha_{nbayes} = \max(n/m, m/n)$ (n represents the number of malicious URL leaf nodes, m represents the number of normal URL leaf nodes), so the size of a cart can characterize the type of occupation that a leaf node decides. Similar to naive Bayesian filter, this threshold can neither be set too small nor too large. So, to train the threshold, through the same training data group, the specific method is discussed in Section 3. if $\alpha_{nbayes} > \alpha_{nbayes}$, then we think that URL is CART Decision tree model has a good at data, otherwise, record the classification results, and the URL is filtered to the next filter.

4.4 SVM filter

The final filter of this model is SVM filter. SVM training model is mainly classified models. It Derived classification function for the upper Layer filter down the URL, record classification results, combined with Naive Bayesian Filter and CART Decision Tree filtering collectively determine the classification of the URL.

V. INSTANCE VALIDATION

5.1 Data sources and experimental environment

The malicious URL dataset in this experiment is downloaded from the malicious website lab (Http://www.mwsl.org.cn/), the normal URL data set is collected from first category directory (http://www.dir001.com/).

10000 samples are taken from each dataset i.e.10000 from malicious URLs and 10000 from normal URLs. This article uses features extraction and data modeling. Python language is used as the implementation programming. Windows 1064 bit as an operating system and Corei5 with 16 GB RAM was used as personal computer.

5.2 Feature Selection

Garera[2] and Gattani[3] make comparisons of comprehensive study regarding URL feature selection. This article mainly from the perspective of domain name cost, malicious website. The creator knows that the domain name of his site has a great risk of being banned, so in the purchase When buying a domain name, you often buy cheaper or even free domains for cost savings Name, and this domain often has the following characteristics: 1) TLD is not the mainstream Domain name; 2) domain name with special characters; 3) domain name length is very long; 4) The main domain consists of meaningless letters or numbers; 5) There are many "." To confuse the domain name structure. Based on the above information, this paper chooses seven features, as shown in Table 1. The calculation method of F7 in Table 1 is as follows: This text chooses the commonly used 5492 English words and 187,207 Chinese word phonetic together constitute a meaningful word Tree, the string of the URL's main domain name is a meaningful list of the tree of words to match, if the match can be considered match these characters are there Meaning, and then use the rules of the characters and the total length of the characters that have ratio Meaning coefficient.

Table I – FEATURE VECTOR EXTRACTION RULES

No	Features
F1	The domain names contained more than 4 consecutive numbers
F2	The domain name contains special characters (#, $, @, ~, _, -)
F3	Top Five domain name (com,en, net,org,cc)
F4	The number of"." In domain name
F5	Domain name total length
F6	The Length of longest domain name segment
F7	Meaningful coefficients in primary domain names

For example, book dsxihuan.com, because the domain name does not contain 4. More than one consecutive number, so F1 is set to 0; due to the domain name does not contain Any of the following special words: #, $, @, ~,-, _, so F2 is set to 0; This example contains one of the top five domain names "com", so F3 is set to 1; The domain name contains a "." Therefore F4 is set to 1;F5 is the total length of the domain name, This example is 16;in this case, the longest domain name is "book dsxihuan" and its length is 12, so F6 is set to 12; the main domain name of this example is" bookds xihuan" with two A meaningful word "book", "xihuan" match, meaningful length of 10, The primary domain has a length of 12, then it has a meaningful coefficient of 10/12 = 0.83. From this, the feature vector of this domain name is expressed as {0,0,1,1,16,12,0.83}.

Table II - URL DATA SET

Dataset Name	Dataset Description
Training model dataset	8000 malicious URL, 8000 normal URL
Training threshold dataset	1000 malicious URL, 1000 normal URL
Testing dataset	1000 malicious URL, 1000 normal URL

Table III- TEST RESULTS FOR VARIOUS MODELS

Model Name	Accuracy Rate %	Rexall Rate %	Precise rate %
Multi-layer filtering model	79.55	68.80	87.64
Simple Naive Bayes	77.30	66.40	84.91
Single Decision Tree	79.35	69.00	87.01
Single SVM	76.80	79.40	75.48

5.3 Training model and threshold

In this paper, we divide the 20,000 URLs collected in section 3.1 into three in the experiment Part, as shown in Table II. If the URL which is to be tested can directly be judged as malicious or normal, so the experimental tuning process does not consider black and white list filter. The blacklist filter can be considered authoritative in this model. The experimental process is as follows: Perform the data on the three datasets using the feature extraction rules in Section 3.2. Train a separate machine learning with the train-model-set sample set Classifiers, including the naive Bayesian model, the CART decision tree model and SVM model Build a separate machine learning model into a multilayer filter model, and at the very beginning give a very large threshold pair, so $\alpha_{nbayes}^{new} = 500, \alpha_{cart}^{new} = 10$ Substitute the train-threshold-set sample set into the multi-layer filter model and calculate the check measurement accuracy, and gradually reduce these two thresholds,

this article uses the formula in the program $\alpha_{nbayes}^{new} = 1 + 0.8\alpha_{nbayes}^{old}, \alpha_{cart}^{new} = 1 + \alpha_{cart}^{old}$, record each thresholds accuracy, and finally pick the combination of the highest accuracy of the function $(\alpha_{nbayes}^{*}, \alpha_{cart}^{*})$ Substituting the threshold combination $(\alpha_{nbayes}^{*}, \alpha_{cart}^{*})$ into the multi-layer filter model, test this multi-layer filter model with three separate test set sample, sets classifiers and record the accuracy rate, recall rate and accuracy.

VI. MULTI-LAYER FILTER MODEL PROCESS

Multi-layer filter model overall process shown in Figure 1. So for a new URL to be detected, the seizure in this model Test process is as follows:

1) Extract feature vectors; into the black and white list filter to determine whether the black and white list, such as If so, then directly determine and end the program, otherwise step 3)
2) Enter Naive Bayesian Filter to determine whether n bayes reaches the threshold αn bayes ∗ If you reach the direct judgment and the end of the process, or record the judge Result R1, perform step 4);
3) Enter the CART decision tree filter to see if cart has reached the threshold Value α cart ∗ If you reach the direct decision and end the procedure, record the decision Fruit R2, perform step 5);
4) Into the SVM filter, the URL feature into the discriminated function, too to the decision result R3, combined with R1 and R2 jointly vote to determine the URL belong Category, ending the program.

This research project has carried on the independence to the voting method, A lot of in-depth study, this article in the three filter classification based on the parallel Voting strategy, that is, when the detected URL into the SVM filter layer, by Park Results generated and recorded by BayesianR1. The CART decision tree is generated and recorded Results recorded R2 and SVM generated and recorded the results of R3 common cast A vote, according to the voting results based on the principle of the majority decision, the final decision URL category. This kind of parallel voting strategy determines the URL category and can be better to share the risk of judgment. In particular, when the SVM classifier produces results that are worse than those of the naive Bayes classifier and the CART decision tree classifier, avoid situations that ultimately lead to the worst conclusion by the SVM alone, In this case, the parallel voting method

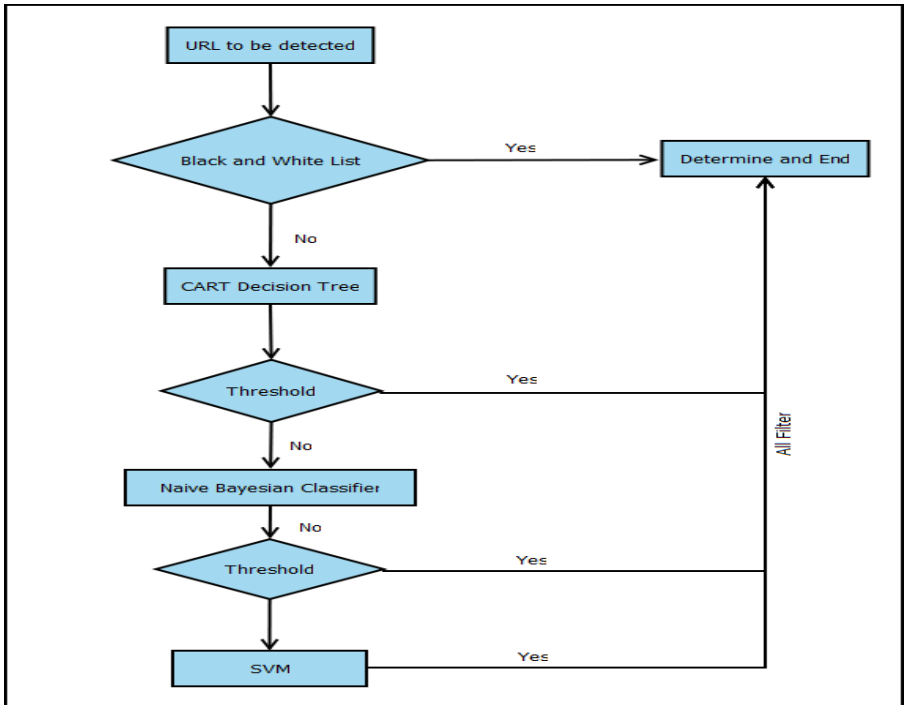

Figure 1: Multi-layer filter model flow chart gives a judgment based on the results of the three kinds of filters. The end result, reducing the risk of judgment.

VII. EXPERIMENT RESULTS

The optimal threshold pair trained after Section 3.3 is $(\alpha_{nbayes}^{*} = 400.2, \alpha_{cart}^{*} = 1.203)$, the results are shown in Table III. In the multi-layer filter model, Naive Bayesian filter determined 308 URL, decision tree filter determined the 1370 URL, another 322 The URL is jointly detected by three filters.

This result is also in line with Table III. The performance of the separate classification models can be seen in three separate models. Decision tree model is comparatively best performing model than Naive Bayes and SVM. It is also observed in Table III that multi-layer filter model performs better than all the three classifier models. Multi-layer filter model can let the classifier to deal with their own good data. Every layer of the model plays a beneficial role for classification of the URLs and ultimately improves the detection of malicious URL in terms of accuracy.

VIII. CONCLUSION

In this paper, black and white list technology and machine learning algorithms were used and formed multi-layer filtering model for detection of malicious URLs. The model was trained for each machine learning algorithm i.e. naive Bayesian classification and decision tree classifier threshold and this threshold is used to refer to guide two classifiers for filtering URL. We combined the Naive Bayesian classifier, Decision Tree classifier and SVM classifiers in one multi-layer model to improve the malicious URL detection system in terms of accuracy. We observed from the real examples that multi-layer filtering models does effective detection of malicious URLs.

REFERENCES

[1] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. "Bayesian CART Model Search." Journal of the American Statistical Association, Vol. 93(443), pp 935–948, September 1998.
[2] Sujata Garera, Niels Provos, Monica Chew, and Aviel D. Rubin. "A framework for detection and measurement of phishing attacks." In Proceedings of the 2007 ACM workshop on Recurring malicious code - WORM '07, page 1, 2007.
[3] Abhishek Gattani, AnHai Doan, Digvijay S. Lamba, NikeshGarera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, AnandRajaraman, and VenkyHarinarayan. "Entity extraction, linking, classifica- tion, and tagging for social media." Proceedings of the VLDB Endowment, Vol. 6(11), pp 1126–1137, August 2013.
[4] David D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. pages 4–15. 1998.
[5] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. "Learning to detect malicious URLs." ACM Transactions on Intelligent Systems and Technology, Vol. 2(3), pp 1–24, April 2011.
[6] FadiThabtah Maher Aburrous, M.A.Hossain, KeshavDahal. "Intelligent phishing detection system for e-banking using fuzzy data mining." Expert Systems with Applications, Vol. 37(12), pp 7913–7921, Dec 2010.
[7] AnkushMeshram and Christian Haas. "Anomaly Detection in Industrial. Networks using Machine Learning: A Roadmap." In Machine Learning for Cyber Physical Systems, pages 65–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.
[8] Xuequn Wang Nik Thompson,Tanya Jane McGill. "Security begins at home: Determinants of home computer and mobile device security behavior." Computers & Security, Vol. 70, pp 376–391, Sep 2017.
[9] Dan Steinberg and Phillip Colla. "CART: Classification and Regression Trees." The Top Ten Algorithms in Data Mining, pp 179–201, 2009.
[10] D. Teal. "Information security techniques including detection, interdiction and/or mitigation of memory injection attacks," Google patents. Oct 2013.
[11] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. "Design and Evaluation of a Real-Time URL Spam Filtering Service." In 2011 IEEE Symposium on Security and Privacy, pp 447–462. May 2011.
[12] Sean Whalen, Nathaniel Boggs, and Salvatore J. Stolfo. "Model Aggregation for Distributed Content Anomaly Detection." In Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop - AISec '14, pp 61–71, New York, USA, 2014. ACM Press.
[13] Ying Yang and Geoffrey I. Webb. "Discretization for Naive-Bayes learning: managing a discretization bias and variance." Machine Learning, Vol. 74(1), pp 39–74, Jan 2009.

