The Kozinec-SVM Model for Detecting Malicious URLs

Lekshmi A R
M. Tech Student
Dept. of CSE
LBS Institute of Technology for Women
Trivandrum, India

Seena Thomas
Assistant Professor
Dept. of CSE
LBS Institute of Technology for Women
Trivandrum, India

***Abstract*— Cyber security is one of the most fast growing field at present. Their main focus is to find the reasons of losing secure information. Malicious URLs usage is one of the main causes of such problems. This types of URLs having a short life span, and are generated daily. Researchers uses different techniques for detecting such URLs. Among those methods, Blacklist is the traditional and simplest approach. But due to its simplicity, it cannot work properly and so that the various Machine Learning techniques are introduced. This paper presents perceptron model known as Kozinec algorithm for classifying and detecting malicious URLs. The perceptron algorithm is used for learning of a linear classifier. Once implemented, the advantage of this algorithm is once implemented, it will also be used for learning of a non-linear extension of the classifier for the case of a quadratic discriminant function. Using this method the proposed system reduces the complexity of detecting malicious URLs. For reducing false positive rate an SVM method is also used. So that the proposed system gives an efficient classification of malicious URLs from benign one.**

Keywords*— *Malicious URLs, Blacklist, Machine Learning, False positives, Kozinec, SVM.

I. INTRODUCTION

More than half of the population having access on the Internet so the usage of the World Wide Web (WWW) has increasing day by day. Among this most of the people uses internet in a good way. But everything in this world having good and bad face. So a few portion of population using internet with bad intentions. For searching information in a browser or opening email messages can happened only after accessing an URL (Uniform Resource Locator). But this people with bad intention can make some malware activities in this URLs. Malware, or malicious attacks are propagated by the usage of Internet. The explicit hacking attempts, drive-by exploits, social engineering, phishing, watering hole, man-in-the middle, SQL injections, loss/theft of devices, denial of service, distributed denial of service, and many others are variety of techniques used to implement website attacks. The limitations of traditional security management technologies are becoming more and more serious given this exponential growth of new security threats, rapid changes of new IT technologies, and significant shortage of security professionals. These attacking techniques are realized by spreading malicious URLs. For avoiding the loss in secure information it is important to identifying the malicious URLs. Identification of such URLs are not an easy task and its always being an important area of Cyber security. Some of the most popular types of malware or malicious URLs attacks are Drive-by Download, Phishing and Social Engineering, and Spam. A drive-by download is a program that is automatically downloaded to our computer without our knowledge. It can be initiated by simply visiting a Web site or viewing an HTML e-mail message. It is installed along with a user-requested application. For affecting the genuine web pages, the phishing and social engineering attacks ploy users into disclosing their sensitive information. Spam is used as voluntary messages for the purpose of advertising or phishing. Detecting malicious URLs in a timely manner is a very big concern exists in today's world. This paper attempts to build a system that uses two layer analysis to analyse the URL in the internet. The systems is learns on itself. By correctly classified URL the database can be updated. For further classification of the new URLs this database is again used for finding matches. This approaches ensures that the system approaches higher levels of accuracy as the system proceeds on to classify. To attain the final results the system uses two detection and classification models. The research paper is organized as follows: Section 2 describes the related works, Section 3 describes the proposed technique, Section 4 presents the results analysis and finally Section 5 describes the conclusion.

II. RELATED WORKS

From Jian Zhang et al. [1], Blacklist method is one of the traditional method for detecting malicious URLs. In this method the set of URLs that are malicious in past can be stored in a database. A search happens in this database can be performed whenever a user visits into a new URL. A warning will be generated to show the URL is malicious if the URL is present in Blacklist. The advantage of this method is it can be implemented very easily and having low positive rate. URLs are generated in a daily basis. Even though it is impossible to detect new ultimatum in daily generated URLs. So for such cases this method is useless.

For identifying phishing sites by extracting phishing site features, Seifert et al [2] introduces a Heuristic or rule based approach which is an extended version of [1]. In this method, a blacklist of signatures is created whenever a new URL is arrived. With the available signature lists, an analysis takes place. As the result of this analysis phase, if a matching will be shown then the URL is referred as malicious. For each recognized attacks, this method assigns a signature. Signature based method and Behavior based method is used in [2] for detecting malicious URLs. The Signature based method is described in [3], which uses additional amount of resources such as time, money and work force for bring out distinctive signature. So Behavior based method is mostly used. This method can be described in [4]. In this method the URLs showing same behavior are collected for detection. But this method takes huge amount of scanning time and it doesn't give the details of false positives. And another drawback is that whenever user visits in a website an immediate attack is launched and it causes the failure in detecting the generated attacks.

Most of the researchers then started to apply different machine learning techniques in detecting and classifying malicious URLs from benign one. Machine learning methods [14] focus on providing a system to learn by itself and improve from experience without having any specific program. This method provides ability to the computer to learn automatically without any interference with humans. Important machine learning techniques are supervised earning, unsupervised learning and semi-supervised learning. The key difference between supervised and unsupervised learning in machine learning is the use of training data [15]. Supervised learning having trained data. Prior training data is not present in unsupervised learning. Semi-supervised learning contains both labelled and unlabeled set of data.

False positive rate determines the efficiency of a method which detect malicious URLs. Several researchers give importance to reduce false positives. False positives are more dangerous than false negatives when considering real life situations. So the methods which doesn't give information about false positives is not that much efficient. According to various studies by different researchers, detecting malicious URLs using different machine learning algorithms must give more importance to false positive rates.

In [5], the researchers uses a unique approach to get a low false positive rate. In this method the average can be taken for the binary classification returned by individual filters, then by taking average of log-odd estimates based on scores returned by individual filters and also using a static method like logistic regression. Overall this method combines results of 53 different spam filters as an alternative of using one filter.

Two method introduced in [6] for minimizing false positive rate as well as false negative rate. Both this method is based on a logistic regression and Naïve Bayes techniques. First method consists of stratifications. Here it gives more importance to weighting good messages than spam data. The second method gives more concern to easily classifying the samples which are discarded and this classification focus on similar ones and this methods can be used to classify even the harder regions.

For detecting malicious URLs, [7] uses different offline supervised and unsupervised learning methods. This is based on memory footprint. Here URL strings only considered for the feature extraction stage and not considering any external information's. In this paper for zero false positives and for good detection rate, One side Class (OSC) perceptron is used. For generating clusters and for URLs classification, Fingerprint algorithms (FASV) is also used. Most of the URLs having short life span. So according to this paper such URLs can be easily detected.

By extracting both lexical form of URLs and using URLs external information Ma.et al [8] find a new possibility. In this paper, the URL detection can be done by using Online supervised learning. Perceptron algorithms and Confident weighted algorithm are two online learning algorithms used here. A real time URL stream is used for updating the elementary model. This method give good detection rate and give a detailed study about the importance of retraining the algorithms with new features. But the one and only dis-advantage of this paper is it doesn't give any detailed information about false positives, thus this method is inefficient.

Blum et al. [9] also follows the same method of [8] without using host based information of URLs. But their studies also doesn't give any details about false positives.

Cost Sensitive Online Active Learning (CSOAL) is another framework used in [10]. CSOAL is focus on detecting real-world online malicious URLs. The feature extraction phase consider both URLs host based information and URLs string information. The ratio between malicious and benign section are highly disproportional while accessing URLs from real word. So in this paper another performance metric such as the sum of weighted sensitivity and specificity can be optimized instead of maximizing online accuracy. This paper correctly classifies malicious URLs from benign one.

In [11] a perceptron algorithm used for zero false positives. This goal of this approach is to classifying all the inputs as benign and malicious more correctly by regulating separate plans for each class. This paper concluded that after a certain period of time, the detection rate is reduces and gives a low false positive rate. Because at the final stage of this method gives a plane. Here one part of the plane contains malicious files only and the other part contains clean and malicious files. It can again analyzed and then they comes with the above conclusion.

SVM-AR ensemble learning approach proposed in [12] is a very good method for detection of malicious URLs. Some of the researchers tries to increase the controlling of malware detection and introduces this method. SVM-AR is a combination of SVMs (Support Vector Machine) and association rules. To classifying benign and malicious files SVM gives a hyper plane. There are some false positives occurred in this SVMs hyper plane. So using association rules, the local optima filters reduces this false positives. This method give more accurate details about detection rate and false positives rates than other methods. Even though this system is very complex.

In [13] Anton et al. proposed a semi-supervised machine learning method to detect malicious URLs. Their main goal is to identifying new and prevalent malicious URLs. For this a system can be created which uses an easily trained detection model. The authors using One side Class (OSC) Perceptron algorithm as training algorithm. The feature extraction phase uses lexical based features of URL. They uses OSC-3 model for detection and error correction. This method uses a cached database for collecting URLs for limited period of time. This system is based on a live stream of URLs. For the limited period of time it gives the exact URLs classification as two classes such as malicious and benign. Due to the usage of OSC-3 perceptron model this method is very complex.

III. PROPOSED METHODOLOGY

To solve the problems discussed in section II, this paper propose a perceptron model to identify, classify and detect malicious URLs. This paper have attempted an approach that employs two machine-learning algorithms. The proposed system is a self-learning or self-adjusting system. In the previous attempts, by Anton Dan Gabriel et al, have very well used the features of the OSC Perceptron models. This paper attempt is inspired by this. Here the system have employed the SVM method over the perceptron based detection system. While main attention is get good detection rates and at the same time keep the false positives at the minimal.

The perceptron model that use here is Kozinec algorithm for the Binary classification of the input URLs. The Kozinec algorithm is explained in section IV. The advantage of perceptron and Kozinec algorithm is that it is not required to know all the inequalities from the system to change values. It suffices to know just a single inequality out of them.

The SVM paradigm in pattern recognition presents a lot of advantages over other approaches some of which are: 1) the unique solution, 2) good generalization properties, 3) rigid theoretical foundation based on SLT and optimization theory, 4) common formulation for the class separable and the class non-separable problems as well as for linear and non-linear problems (through the so called “kernel trick”) and, last but not least, 5) clear geometric intuition of the classification problem. Due to these very attractive properties, SVM have been successfully used in a number of applications.

The lexical features extracted here is type of protocol, country code, domain, sub-domain, relative URL, section and sub-section. Whenever the users enters a new URL, all the above features can be extracted and check a match with the available set of malicious URLs features. If a match exists then it will block the user to use that URL. Otherwise the features extracted from new URL can be quantified and give for training and testing purpose.

The perceptron model is one of the type of machine learning approach. This model can be used for the linear classification. It make its predictions based on a linear predictor function combining a set of weights with the feature vector.

Kozinec algorithm is a perceptron model which is used here for classifying the input URLs as malicious or benign one. The Kozinec algorithm is explained in section A. Then the output from Kozinec algorithms is again put into the Support Vector Machine (SVM) for better results and reduce false positives. The SVM is explained in section B in detail. SVM produce a hyperplane which is shown in figure 5.

A. Kozinec Algorithm

The input is data set T = {(x1, y1)… (xl, yl)} of binary labeled yi = {1, 2} training vectors xi 2 Rn . The Kozinec’s algorithm builds a series of vectors $\mathbf{w}_1^{(0)}, \mathbf{w}_1^{(1)}, \ldots, \mathbf{w}_1^{(t)}$ and $\mathbf{w}_1^*$ and $\mathbf{w}_2^*$, , which converge to the vector $\mathbf{w}_1^*$ and $\mathbf{w}_2^*$ respectively. The vectors are the solutions of the following task:

$$\mathbf{w}_1^*, \mathbf{w}_2^* = \underset{\mathbf{w}_1 \in X_1, \mathbf{w}_2 \in X_2}{\arg\min} ||\mathbf{w}_1 - \mathbf{w}_2||$$

Where X1 stands for the convex hull of the training vectors of the first class X1 = {xi: yi = 1} and X2 for the convex hull of the second class likewise. The vector $\mathbf{w}^* = \mathbf{w}_1^* - \mathbf{w}_2^*$ and the bias $b^* = \frac{1}{2}(||\mathbf{w}_2^*||^2 - ||\mathbf{w}_1^*||^2)$ determine the optimal hyperplane, the below equation

$$(\mathbf{w}^*, b^*) = \underset{\mathbf{w},b}{\arg\max}\, m(\mathbf{w}, b)$$

$$= \underset{\mathbf{w},b}{\arg\max} \min\left(\min_{i \in y_1} \frac{\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b}{||\mathbf{w}||}, \min_{i \in y_2} - \frac{\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b}{||\mathbf{w}||}\right)$$

Separating the training data from the maximal margin. The Kozinec’s algorithm is proven to converge to the vectors $\mathbf{w}_1^*$ and $\mathbf{w}_2^*$ in infinite number of iterations t = ∞. If the e-optimal optimality stopping condition is used, then the Kozinec’s algorithm converges in the finite number of iterations. The Kozinec’s algorithm can also be used to solve a simpler problem of finding the separating hyperplane in equation

$$q(\mathbf{x}) = \begin{cases} 1 & \text{if } f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \geq 0 \\ 2 & \text{if } f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b < 0 \end{cases}$$

Therefore, the following two stopping conditions are implemented:

- The separating hyperplane is sought for e < 0. The Kozinec’s algorithm is proven to converge in a finite number of iterations of the separating hyperplane that exists.
- The e-optimal hyperplane is sought for e >= 0. Note that setting e = 0 forces the algorithm to seek the optimal hyperplane which is generally ensured to be found in an infinite number of iterations t = ∞.

B. Support VectorMachine

One of the best classification method is Support Vector Machine (SVM). The principle of SVM is fitting a boundary to a region of points which belongs to the same class. By checking the already fitted boundary (on the training sample), the new points (test samples) can be classified. i.e., by simply checking if it is present inside the boundary or not. In SVM, once a boundary is established, most of the training data is redundant. This is one of the advantage of SVM. For identifying and setting the boundary SVM only uses some set of points. This points is called Support Vectors. SVM is called vectors because it supporting the boundary, each of this data points is a vector. i.e., Data in each row contains the value for a number of attributes. All it needs is a core set of points which can help identify and set the boundary.

Traditional name of the boundary shown in figure 3 is hyperplane. In case of having two attributes (2-D), this boundary can be a straight line or a curve as shown as the above figure 3. It can be a plane or a complex surface in 3-D. In this paper the hyperplane will give a straight line and the points are shown in two side of the plane.

Linearly Separable: For the data which can be separated linearly, select two parallel hyperplanes that separate the two classes of data, so that distance between both the lines is maximum. The region b/w these two hyperplanes is known as "margin" & maximum margin hyperplane is the one that lies in the middle of them.

$$\vec{w}x_i - b \geq 1 \text{ if } \theta_i = 1$$
$$\vec{w}x_i - b \leq 1 \text{ if } \theta_i = -1$$

Where $||\vec{w}||$ normal vector to the hyperplane, θi is denotes classes & xi denotes features. The Distance between two hyperplanes is $\frac{2}{\|\vec{w}\|}$, to maximize this distance denominator value should be minimized. For proper classification, we can build a combined equation:

$$\|\vec{w}\|_{min} \text{ for } \theta_i \left(\vec{w}x_i - b\right) \geq 1 \ \forall i = 1,\ 2,$$

Non-Linearly Separable: To build classifier for non-linear data, we try to minimize

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\vec{w}\cdot\vec{x}_i - b)\right)\right] + \lambda\|\vec{w}\|^2,$$

Here, max () method will be zero (0), if x_i is on the correct side of the margin. For data that is on opposite side of the margin, the function's value is proportional to the distance from the margin. Where, λ determines trade-off b/w increasing the margin size and that $\vec{x}_i$ is on correct side of the margin.

IV. RESULTS

Firstly we analyzed the lexical features of the input URL by capturing seven features of the URL. Then quantify each feature value depending on the features' resemblance with the already known benign and malicious set of URLs. The systems is trained with the existing list of correctly pre-classified URLs. And it is tested with two layers of the learning models, first model is based on the Kozinec algorithm and later trained and test with the SVM model.

It is observed the detection maintains a steady 80% - 85% detection rate for the first 100 input urls. Then it drops to 60% where it remains for almost next 300 URLs. As the Input URLs are analyzed the detection rate reduces further. However, the drop of detection is takes place over a period of time. This means that we can adjust the model and be better prepared for the types of URLs that are to come. The results are better when the two layer model is used than the single layer model alone. Figure 5 shows the classification of malicious and benign URLs in a hyper plane.

In figure 5, the above portion of hyper plane shows the benign set of URLs and the below portion of hyper plane shows the malicious set of URLs.

The results of the system were manually tested and the overall detection rate were found to be improved to 84%. The number of false positives are more, but when considering the total number of URLs test, the false positives amount to be less than one percent (0.85%). The number of false positives were found to be decreasing as the test progressed.

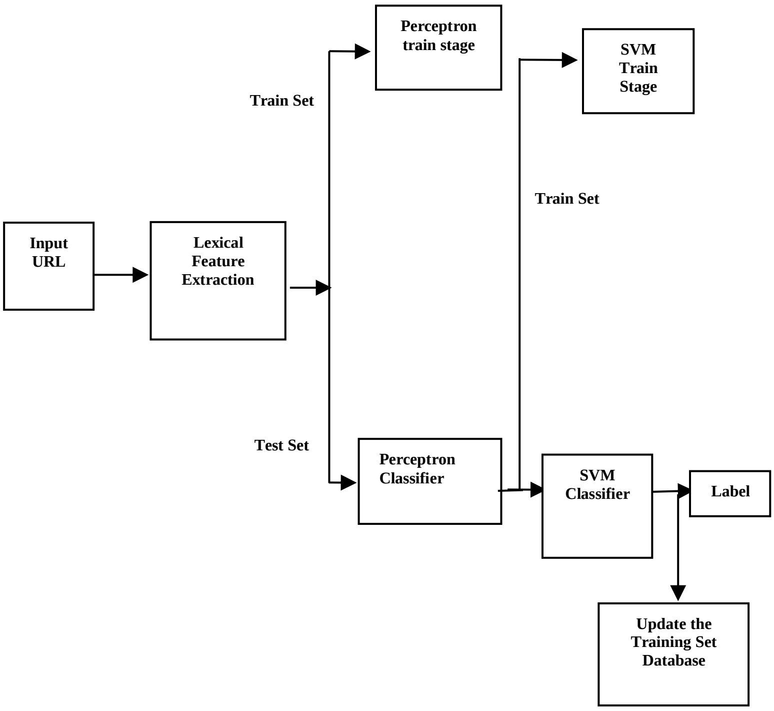

Figure 1: Block diagram of Proposed System

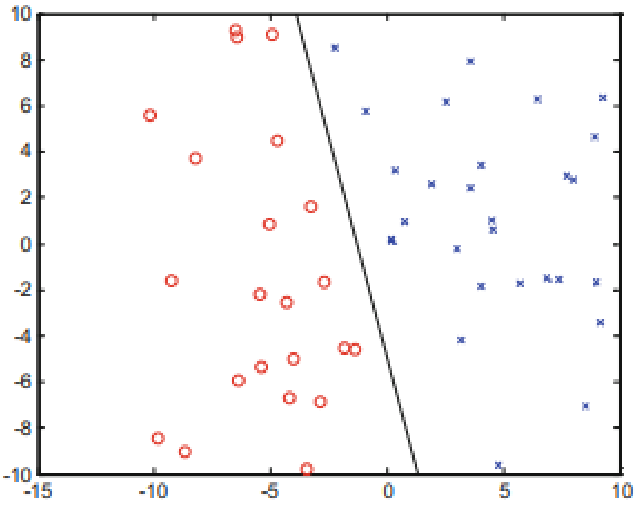

Figure 2: Kozinec's algorithm for linear classification

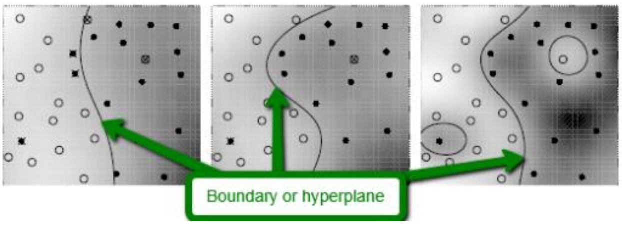

Figure 3: Hyper plane in 2-Dimension

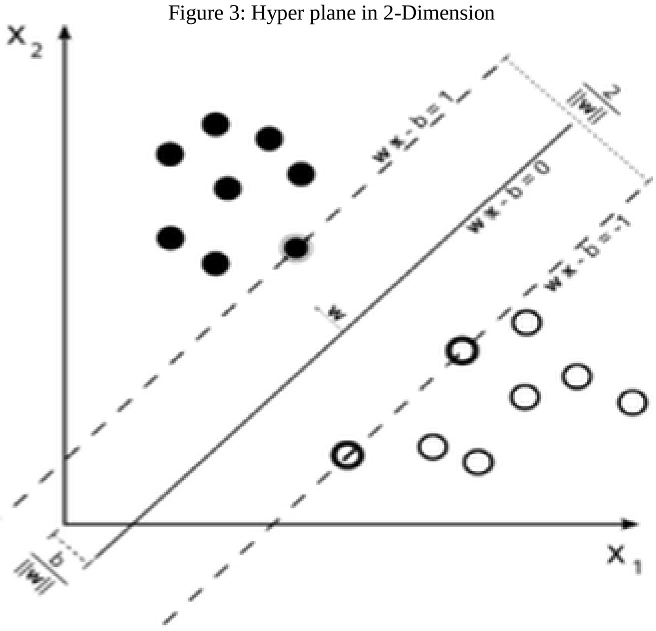

Figure 4: Selecting the SVM hyperplane

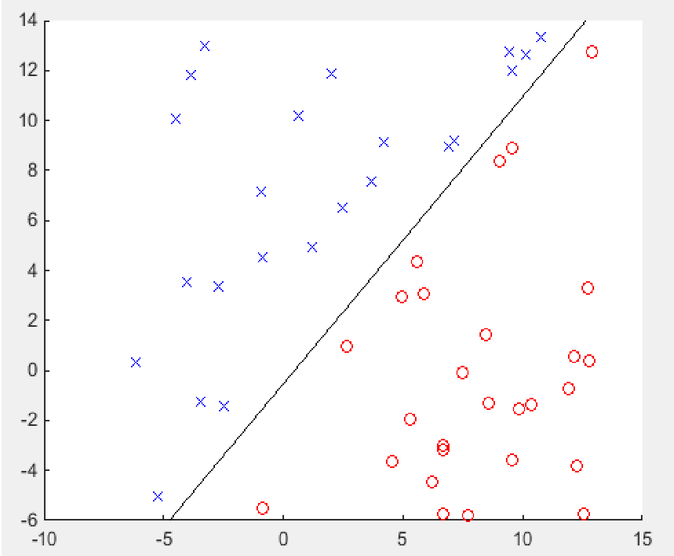

Figure 5: Hyper plane shows malicious and benign URLs classification

V. CONCLUSIONS

This paper implemented as a twofold URL cleanness detection system with one of the perceptron models called Kozinec algorithm for the classification and detection of malicious URLs from benign one. By using this method, the newly entered URLs can be also be classified as malicious or benign and updated into the available datasets of malicious and benign URLs. So this makes future users to identify whether the entered URL is malicious or not and it will overcome the security issues happening at that time. This paper also implemented an SVM model in the second phase. By using this two perceptron model i.e., Kozinec-SVM model the false positive rate is reduced and it gives better classification of malicious URLs. It can be found that by using these two model together the malicious URLs can easily be detected and updated into the datasets. Also it was found that the test results improved as the training data set is updated with the test results, after every iteration.

REFERENCE

[1] Jian Zhang, Phillip A. Porras, and Johannes Ullrich. Highly predictive blacklisting. In Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA, pages 107122, 2008.

[2] C. Seifert, I. Welch, and P. Komisarczuk, Identification of malicious web pages with static heuristics, in Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian. IEEE, 2008, pp. 9196.

[3] P. Gutmann. The Commercial Malware Industry., 2007.

[4] KALPA, Introduction to Malware, 2011.

[5] Thomas R. Lynam, Gordon V. Cormack, and David R. Cheriton. On-line spam filter fusion. In SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, pages 123–130, 2006.

[6] Wen-tau Yih, Joshua Goodman, and Geoff Hulten. Learning at low false positive rates. In CEAS 2006 - The Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California, USA, 2006.

[7] Adrian-Stefan Popescu, Dumitru-Bogdan Prelipcean, and Dragos Teodor Gavrilut. A study on techniques for proactively identifying malicious urls. In 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2015, Timisoara, Romania, September 21-24, 2015, pages 204–211, 2015.

[8] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Identifying suspicious urls: an application of large-scale online learning. In Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009 Montreal, Quebec, Canada, June 14-18, 2009, pages 681–688, 2009.

[9] Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. Lexical feature based phishing URL detection using online learning. In Proceedings of the 3rd ACM Workshop on Security and Artificial Intelligence, AISec 2010, Chicago, Illinois, USA, October 8, 2010, pages 54–60, 2010.

[10] Peilin Zhao and Steven C. H. Hoi. Cost-sensitive online active learning with application to malicious URL detection. In The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013, pages 919–927, 2013.

[11] Dragos Gavrilut, Mihai Cimpoesu, Dan Anton, and Liviu Ciortuz. Malware detection using machine learning. In Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT 2009, Mragowo, Poland, 12-14 October 2009, pages 735–741, 2009.

[12] Yi-Bin Lu, Shu-Chang Din, Chao-Fu Zheng, and Bai-Jian Gao. Using multi-feature and classifier ensembles to improve malware detection. Journal of C.C.I.T., 39(2), 2010.

[13] Anton Dan Gabriel, Dragos Teodor Gavrilut, Baetu Ioan Alexandru, Popescu Adrian Stefan. Detecting malicious URLs. A semi-supervised machine learning system approach. 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2016.

[14] Marco Varone, Daniel Mayer, Andrea Melegari https://www.expertsystem.com/machinelearningdefinition/

[15] https://www.techopedia.com/what-is-the-difference-between-supervised-unsupervised-and-semi-supervised-learning/7/33373

[16] https://searchenterprisedesktop.techtarget.com/definition/drive-by-download

[17] https://searchenterprisedesktop.techtarget.com/definition/drive-by-download