2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

Behavioral-Based Classification and Identification of Ransomware Variants Using Machine Learning

Hajredin Daku, Pavol Zavarsky, Yasir Malik
Information Systems Security and Assurance Management
Concordia University of Edmonton, Edmonton, Alberta, Canada
hdaku@csa.concordia.ab.ca, {pavol.zavarsky, yasir.malik}@concordia.ab.ca

Abstract—Due to the changing behavior of ransomware, traditional classification and detection techniques do not accurately detect new variants of ransomware. Attackers use polymorphic and metamorphic techniques to avoid detection of signature-based systems. We use machine learning classification to identify modified variants of ransomware based on their behavior. To conduct our study, we used behavioral reports of 150 ransomware samples from 10 different ransomware families. Our data-set includes some of the newest ransomware samples available, providing an evaluation of the classification accuracy of machine learning algorithms on the current evolving status of ransomware. An iterative approach is used to identify optimum behavioral attributes used to achieve best classification accuracy. Two main parts of this study are identification of the behavioral attributes which can be used for optimal classification accuracy and classification of ransomware using machine learning algorithms. We have evaluated classification accuracy of three machine learning classification algorithms.

Keywords— behavioral analysis, malware classification, machine learning, malware, ransomware, malware detection

I. INTRODUCTION

Ransomware attacks are becoming a serious cyber threat to organizations and individuals around the globe. Unlike traditional malware, ransomware attackers hijack the system and demand money to reverse the attack. A recent study, reports an immense increase and steady growth in new ransomware samples and attacks [1]. Moreover, the ability of the attackers to create new variants of existing malware using metamorphic, polymorphic, obfuscation and other masking techniques makes it harder to effectively detect and classify these attacks using static detection systems, since they may have different content or signature from their previous versions [2][3]. Often, new variants of malware are not distinguished from their predecessors due to the limitations of classification systems relying only on static analysis. Thus, techniques like content-based, signature-based and pattern matching techniques for malware analysis are becoming less effective to detect and classify new variants of ransomware and provide insight information about the threat, goals and behaviors of ransomware.

Efforts have been made to develop behavior-based classification techniques [4] [5] [6] [7]. Classification of malware samples based on their behavior requires implementation of algorithms that are capable to produce models and learn through the classification process. The ability of machine learning to learn with data during the process of classification, makes them attractive and effective for malware classification. Using machine learning classification algorithms, ransomware samples can be identified with different behaviors from other samples that are part of the same family.

The aim of this study is to identify new modified variants of ransomware based on their behavior. In this research, we studied behavioral reports of different ransomware families and classified modified variants of ransomware based on their behavior. Based on the existing classification of ransomware, it is logical to think that samples from the same family would have very similar behavior. However, during our analysis it is noted that this is not always true. Using machine learning classification algorithms, we can identify ransomware samples that behave different from other samples that are part of the same family. We used an iterative approach to identify optimum behavioral attributes to achieve best classification accuracy. We performed experiments using machine learning classification algorithms to identify the behavioral attributes set which can be used to achieve the optimal accuracy of classification. Furthermore, classification accuracy of machine learning algorithms is evaluated using our ransomware dataset on our selected behavioral attributes.

The rest of the paper is organized as follows. In Section II, we present discussion on related work on static and dynamic techniques used for malware classification, and the use of machine learning for malware classification. In Section III we discuss our classification approach including the steps that were followed to complete our study. Section IV present experiments and results. Section V, discusses the results of behavioral attributes selection process and potential future work as an extension of this study. Finally, in Section VI we conclude our work by highlighting research findings.

II. RELATED WORK

In this section we review some of the techniques/approaches used for malware classification. Traditional approaches use signature-based and pattern matching techniques to identify and classify malware. Often, new variants of malware are not distinguished from their predecessors due to the limitations of classification systems relying only on static analysis. Moreover, obfuscation techniques are used to create new obfuscated variants of existing malware which may not be detectable from static detection systems, since they may have different content or signature from their previous versions. Baig et al. identified some of the techniques that can be used to evade static detection systems. Their study reports that static detection systems can be evaded by modifying the properties of a packed portable executable [3]. In another study, Moser et al. explored the limitations of static analysis for malware detection. They

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00224

developed a binary modification tool to make the necessary changes to the malware before testing them against virus scanners and other advanced static analysis tools. Using a binary obfuscation scheme, they could prove that advanced semantic-based malware detectors can be evaded, because they couldn't identify the new modified malware samples [2]. Similarly, Sung et al. modified the original malware using obfuscation techniques, and proved that all tested commercial anti-virus tools and scanners failed to detect the new modified variants of malware [8].

Unlike static analysis, dynamic analysis combined with machine learning techniques are considered to be more effective to achieve even better results for malware classification. Joshua et al. presented a comparison analysis of Anubis [9] and Cuckoo [10] sandbox behavioral analysis reports when used for classification, and achieved similar results without being able to conclude if one of the sandboxes performs better than the other [11]. In another work, Rieck et al. proposed a framework for automatic analysis of malware behavior using machine learning. This framework generates detailed reports of the monitored behavior that is embedded in a vector space, where the similarity of the behavior of the malware can be evaluated geometrically. Using clustering and classification approaches, novel classes of malware can be identified [12]. Nari et al. presented an automated malware classification system that uses network behavior to classify malware into their respective families. Behavioral attributes like graph size, root out-degree, average out-degree, maximum out-degree number of specific nodes were used as an input in the next step of classification [5]. J48 decision tree algorithm implemented in WEKA [13] library was proven to perform better than other classifiers. Bayer et al. proposed a scalable clustering approach to effectively identify and group malicious binaries that exhibit similar behavior [6]. Information provided by a tainting-propagation system combined with the other information from analysis reports, is used to create a behavioral profile. The behavioral profiles from the previous step, serve as an input to a clustering algorithm which is used to group malware samples with similar behavior. Pan et al. presented an automated malware classification approach based on the behavior analysis [7]. They perform dynamic analysis to obtain a behavior profile of the malware samples, which serves as an input to a back-propagation network model which is used for the classification.

It is clear from the research reviewed that malware classification systems based on static analysis are no longer effective to correctly classify malware. Furthermore, behavioral-based systems are considered to be more effective than content based. Since new malware is always designed with improvements to evade classification systems, further research is required to evaluate the effectiveness of classification approaches such as the implementation of machine learning algorithms and behavioral analysis reports for malware classification.

III. Behavioral Based Classification of Ransomware

Dynamic malware analysis can be more effective than static analysis. Automated clustering and classification techniques can be used to identify malware samples with similar behavior. These techniques can help analysts differentiate new malware from modified versions of existing malware. The framework of our study consists of three main phases: data collection, extraction of behavioral attributes and selection of behavioral attributes for optimal classification accuracy. In the data collection phase, we collect behavioral reports from VirusTotal [14] for every ransomware sample. In the next step, behavioral attributes are extracted from the behavioral reports. For ultimate classification accuracy, we perform behavioral attributes selection analysis to identify behavioral attributes which should be used for classification in the next phase. Using the selected behavioral attributes, we evaluate classification accuracy of machine learning algorithms.

Two different approaches can be used for the collection of malware samples. The first one is to collect malware samples from the wild, which are not classified from any anti-virus vendor and then try to classify them into new families. The second approach, and the one we use for our experiments is to collect malware samples that are classified from anti-virus vendors to their respective families. For every ransomware sample within a specific family, a folder named with the SHA256 of the sample is used to store behavioral reports files. We use the SHA256 to eliminate duplication of data in our dataset. The information contained in the behavioral analysis reports section is organized in our dataset along with other information like the analysis section which includes the information if an antivirus detected the malware or not and if it detected, which family the malware was assigned to. To be able to feed our data to the classification algorithms, for each of the behavioral reports mentioned above, we need to extract the behavioral attributes. These attributes are used to represent the data as a two-dimensional matrix, where the columns represent the attributes and the rows represent the value for each attribute.

The main goal of the behavioral attributes extraction phase is to obtain a dataset which best represents the behavior of a ransomware sample without missing any relevant information. Therefore, we spent a considerable time and effort extracting the behavioral attributes from all the behavioral reports. Identified behavioral attributes appear at least in one of the behavioral reports. For each of the behavioral attributes, based on the type of information contained in the behavioral reports, we determine the attribute type to be used to assign a value to the attribute.

Selection of behavioral attributes in a dataset is often a crucial step in the process of classification. Guyon and Elisseeff analyzed several approaches proposed for selection of attributes by performing extensive testing on a wide variety of datasets [15]. We have considered some of their findings during our behavioral attributes selection experiments. We need to perform behavioral attributes selection to improve the performance of the system, reduce time and const, and provide a better understanding of the classification process and the results obtained during experiments. Non-redundancy is also an important requirement to have optimal behavioral attributes set. Having behavioral attributes that are closely dependent on each other, can lead to inaccurate results.

One of the objectives of our study is to determine which behavioral attributes can be used to achieve the optimal accuracy of classification using machine learning algorithms. We have performed multiple experiments and testing using an iterative approach to be able to identify a set of behavioral attributes which

contributes to an improved classification accuracy of machine learning algorithms used for experiments. More details on the experiments performed for behavioral attributes selection are presented in the next section.

IV. EXPERIMENTS

For experiments, we have collected behavioral reports of 150 ransomware samples from 10 different ransomware families listed on Table I. The dataset which contains information extracted from behavioral reports of ransomware samples is processed in WEKA for classification and identification of ransomware variants.

TABLE I: RANSOMWARE FAMILIES AND NUMBER OF SAMPLES IN DATASET

1. Cerber (14)	6. Locky (20)
2. Cryptowall (15)	7. Petya (11)
3. Crysis (16)	8. Sage (16)
4. Jaff (17)	9. Torrent Locker (20)
5. Jigsaw (11)	10. Wannacry (10)

A. *Behavioral attributes selection*

One of the challenging parts of our experiments was the behavioral attributes selection phase. During behavioral attributes selection experiments, we have considered mainly two approaches: an iterative approach to identify attributes which contribute to a better classification accuracy by performing extensive testing and experimental analysis, and grouping attributes which are closely dependent on each other. As we will describe in the sections below, our approach of behavioral attributes selection was to first identify irrelevant behavioral attributes (in some cases, groups of behavioral attributes) until we are left with the optimal behavioral attributes set. During attribute selection process, any attribute selection or grouping was done without any accuracy lost. Every event of attribute selection or grouping should maintain the same classification accuracy or improve it. That is why we check classification accuracy every time we do changes in our behavioral attributes set.

We started our behavioral attributes selection experiments by selecting the behavioral attributes which appeared in over 95% of the behavioral reports and the event of selecting these behavioral attributes was effective as the classification accuracy of J48 algorithm was slightly improved. On the next steps we have selected attributes based on the analysis of the results of classification of J48 algorithm. We use an iterative approach to select behavioral attributes that contribute to a higher classification accuracy. Based on the decision trees obtained from J48 we identify behavioral attributes with high importance located at the top levels of the tree and consider them for the next experiments until we are left with the optimal behavioral attributes set. An iterative testing process is used to evaluate behavioral attributes for classification until all the attributes that appear in the lower nodes of the decision tree also appear on the higher levels of the tree.

After all the experiments and testing during attributes selection process, we were able to identify 12 attributes listed on Table II. We have identified these attributes out of 27 initially discovered in the behavioral analysis reports. Based on our testing and experiments, the list of our selected behavioral attributes is identified as the list which can be used to achieve the highest classification accuracy.

TABLE II. LIST OF BEHAVIORAL ATTRIBUTES SELECTED FOR CLASSIFICATION

	Attributes	Type of attribute
1.	Family	Nominal
2.	File Access	Binary
3.	Created Processes	Numeric
4.	Required permissions	Binary
5.	Injected Process	Binary
6.	Shell Commands	Numeric
7.	Searched Windows	Binary
8.	Opened Service Managers	Binary
9.	Created Mutexes	Numeric
10.	Opened Mutexes	Numeric
11.	Open Services	Numeric
12.	Runtime DLL	Numeric

As mentioned previously, to verify that our behavioral attributes selection approach is effective, we have evaluated classification accuracy of machine learning algorithms during the behavioral attributes selection process. Fig. 1 shows how the classification accuracy of J48 and K-Nearest Neighbor classification algorithms was improved after we perform behavioral attributes selection.

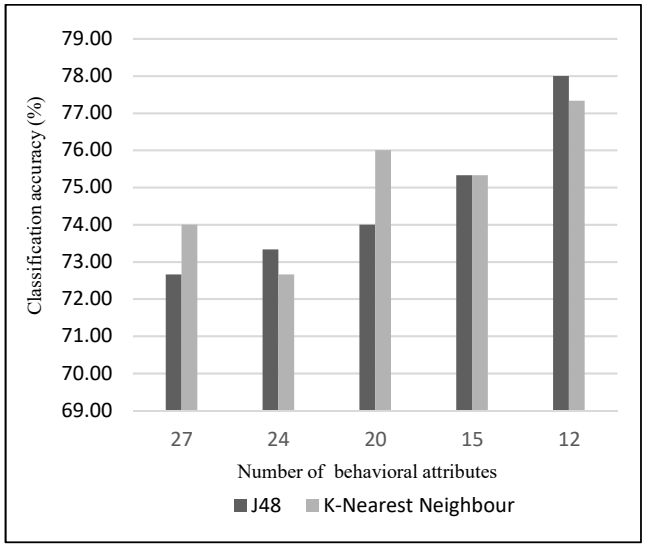

Fig. 1. Accuracy of J48 and K-Nearest Neighbor during behavioral attributes selection

Classification accuracy of J48 was improved from 72.66% when using 27 behavioral attributes for classification, to 78% when using 12 behavioral attributes selected for classification. Our iterative approach to select behavioral attributes using decision trees from J48 algorithm, is also effective in improving overall classification accuracy of K-Nearest Neighbor algorithm.

The selection of behavioral attributes process was evaluated in certain milestones as pointed in Fig. 1. The first step of selecting 24 attributes from 27 was done based on the appearance of the attributes in the dataset. In the next two steps selecting 20 out of 24 and 15 out of 20 behavioral attributes, we used our iterative approach to select behavioral attributes based on the results of J48. In the last phase we select 12 attributes by grouping attributes to reduce redundancy.

B. Classification

After behavioral attributes selection experiments, we evaluated classification accuracy of different machine learning classification algorithms available in WEKA. The optimized dataset is used to classify the ransomware samples into their families. Accuracy of classification is noted for each of the algorithms and the best classifier is used for further analysis and discussion.

Algorithms we choose are J48 Decision Tree, Naïve Bayes and K-Nearest Neighbor. J48 Decision Tree algorithm is an implementation of the C4.5 algorithm in WEKA. C4.5 is used to generate decision trees using a training dataset for the creation of the tree [18]. Naïve Bayes is a machine learning classification algorithm that is based on the application of Bayes Theorem [17]. It relies on the idea of considering each of behavioral attributes independently when evaluating the probability of each behavioral attribute. K-Nearest Neighbor is a simple machine learning algorithm used for classification and regression [19]. In the classification process, an instance is being assigned to the class of the k nearest neighbors. For our experiments we have used k=1, the default value in WEKA and with the best accuracy results.

We evaluated the classification accuracy of the above-mentioned classification algorithms using n-fold cross validation to test the models in the training phase. We have chosen n=10, the default value in WEKA. Below is a comparison of the performance of our classifiers, when using our 12 selected behavioral attributes for classification. The J48 classification algorithm has the best performance with 78% classification accuracy followed by the K-Nearest Neighbor with 77.33%. Naïve Bayes algorithm performed with the lowest accuracy.

TABLE III. PERFORMANCE OF CLASSIFICATION ALGORITHMS

Algorithm	Correctly Classified Instances	Incorrectly classified Instances	Classification Accuracy
J48 [18]	117	33	78%
Naïve Bayes [17]	92	58	61.33%
K-Nearest N [19]	116	34	77.33%

Table IV. below shows the classification accuracy of J48 for every ransomware family. Cerber family was identified to have 50% classification accuracy and is much differentiated from other families which have classification accuracy over 68%. So, a question was raised: is Cerber trying to evade machine learning? According to a detailed report from Sison [16], new versions of ransomware from Cerber family, have adopted a new technique to evade machine learning detection solutions. The report posted on March 2017, shows that the new versions could evade static machine learning malware detection systems, but not systems which use dynamic analysis combined with machine learning. Our results show that Cerber has been evolving since then, and now it might be trying to evade detection systems which use machine learning combined with dynamic analysis.

TABLE IV. CLASSIFICATION ACCURACY OF J48 PER FAMILY

Crysis	100%	Petya	72.72%
Jigsaw	90.9%	Jaff	70.58%
Torrent Locker	90%	Wannacry	70%
Cryptowall	86.66%	Sage	68.75%
Locky	75%	Cerber	50%

During classification and behavioral attributes selection experiments, we compared results by verifying the changes that were observed in the confusion matrices. Fig. 2 below shows the confusion matrices of J48 before and after feature selection process. Numbers located in the diagonal of the matrix, bolded and highlighted with the darkest gray, show the number of ransomware samples classified in their respective families. Numbers outside the diagonal show the number of ransomware samples which were classified in families other than the one they were assigned to in our labeled dataset. Fig. 2(a) shows the confusion matrix of J48 that is the result of classification when using 27 behavioral attributes, before we start behavioral attributes selection experiments. Fig. 2(b) shows the confusion matrix of J48 when using the 12 selected behavioral attributes for optimal classification accuracy. Comparing the results from the confusion matrices, we see that overall classification accuracy was improved.

In the confusion matrix of optimal classification in Fig. 2(b), samples that are not classified in their family, are identified as variants of a ransomware family. These modified ransomware variants have a unique behavior compared to their predecessors, possibly to evade detection systems. From the classification matrix, we can find that 4 out of 14 Cerber samples were classified as Locky and only half of the samples were classified as Cerber. Cerber appears to have very different variants that can easily be misclassified in other families which might lead in the evasion of detection techniques designed to detect Cerber malware only. Also, 5 Locky samples were classified in 5 different families showing 5 different variants of this family discovered in 20 samples considered in our study. From all the families in dataset, samples from Crysis family were all correctly classified, which means they share similar behavior.

V. DISCUSSION

During our experiments and evaluation of classification accuracy, we considered reducing the number of attributes to less than 12 and verify classification results. To reduce the number of attributes from 27 to 12, we used results from classification algorithms (J48 in particular) and merging of attributes to reduce redundancy. Each of the 12 selected attributes contained relevant information for at least one of the ransomware families. To verify that by removing one or more of these attributes the classification accuracy would start to decline, we performed more experiments. In one of our experiments, we reduced the number of attributes to only 9 attributes and evaluated the classification results. We verified that not all the families maintained or increased classification accuracy, using only 9 attributes for classification.

Classified as→		a	b	c	d	e	f	g	h	i	j
a=Wannacry	a	7	1	0	0	1	0	0	1	0	0
b=Locky	b	1	15	1	1	1	0	0	0	1	0
c=Cerber	c	1	4	6	0	0	0	1	0	0	2
d=Cryptowall	d	0	2	0	10	0	2	0	0	1	0
e=Crysis	e	0	0	0	0	16	0	0	0	0	0
f=Jaff	f	0	1	0	1	0	11	3	0	0	1
g=Jigsaw	g	0	0	0	0	0	0	9	1	0	1
h=Petya	h	0	0	1	0	1	1	0	7	0	1
i=Sage	i	0	1	2	0	0	0	2	0	11	0
j=Torrent L.	j	0	0	2	0	0	1	0	0	0	17

(a)

Classified as→		a	b	c	d	e	f	g	h	i	j
a=Wannacry	a	7	0	0	1	1	0	0	1	0	0
b=Locky	b	1	15	1	1	1	0	0	0	1	0
c=Cerber	c	0	4	7	0	0	0	1	0	0	2
d=Cryptowall	d	1	0	0	13	0	0	0	0	1	0
e=Crysis	e	0	0	0	0	16	0	0	0	0	0
f=Jaff	f	0	0	0	2	0	12	2	0	0	1
g=Jigsaw	g	0	0	0	0	0	0	10	0	0	1
h=Petya	h	0	0	1	1	0	0	0	8	1	0
i=Sage	i	1	0	2	0	0	0	2	0	11	0
j=Torrent L.	j	1	0	1	0	0	0	0	0	0	18

(b)

Fig. 3. Confusion matrix of ransomware, (a) 27 behavioral attributes, (b) 12 behavioral attributes

From our 10 families, Petya and Wannacry had lower classification accuracy after reducing the number of behavioral attributes to 9. Therefore, we verified that reducing the number of attributes to less than 12, can lead to missing relevant information for ransomware families.

More work can be done for behavioral attributes selection for classification and detection purposes considering a combination of static and dynamic analysis reports. The results of the behavioral attributes selection and classification experiments can be used to improve existing malware classification systems with the use of machine learning classification algorithms. Future work includes further analysis on clustering and classification approaches using machine learning algorithms and considering a combination of static and dynamic analysis for malware classification and detection.

VI. Conclusion

In this paper, we studied the implementation of machine learning algorithms for malware classification based on the behavior of malware samples. Firstly, we collected behavioral analysis reports of ransomware samples from VirusTotal and extracted behavioral attributes used for classification. Then, to achieve better classification results, we performed behavioral attributes selection experiments. Using an iterative approach we determined the set of behavioral attributes which can be used for ransomware classification to achieve the optimal classification accuracy. Moreover, we evaluated classification accuracy of three machine learning algorithms available in WEKA. The J48 Decision Tree algorithm was evaluated to have the best performance for classification. Using machine learning, we identified modified variants of ransomware samples, confirming the new trend of malware in evading classification and detection systems by modifying their behavior. We identified ransomware samples from evolving families with a diverse behavior compared to their predecessors. The intention of creating malware variants with various behaviors might be to evade detection systems by presenting a rare behavior on new samples, or to mislead detection and classification systems by using a similar behavior to other ransomware families.

References

[1] "McAfee Labs Threat Report,"", Mcafee.com, 2018. [Online]. Available: https://www.mcafee.com/ca/resources/reports/rp-quarterly-threats-dec-2017.pdf. [Accessed: 03- Feb- 2018].

[2] A. Moser, C. Kruegel, E. Kirda, "Limits of static analysis for malware detection", Proceedings of the 23rd Annual Computer Security Application Conference (ACSAC), pp. 421-430, 2007.

[3] M. Baig, P. Zavarsky, R. Ruhl, and D. Lindskog, "The study of evasion of packed PE from static detection, " in World Congress on Internet Security (WorldCIS), Guelph, ON, 2012, pp. 99-104.

[4] E. Gandotra, D. Bansal, S. Sofat, "Malware analysis and classification: A survey", Journal of Information Security, vol. 5, no. 02, pp. 56, 2014.

[5] S. Nari and A. A. Ghorbani, Automated malware classification based on network behavior, in Computing, Networking and Communications (ICNC), 2013 International Conference on, 2013, pp. 642-647

[6] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda. Scalable, Behavior-Based Malware Clustering. In 16th Annual Network & Distributed System Security Symposium (NDSS), San Diego, CA, February 2009.

[7] Z. Pan, C. Feng and C. Tang, "Malware Classification Based on the Behavior Analysis and Back Propagation Neural Network", ITM Web of Conferences, vol. 7, p. 02001, 2016.

[8] A. Sung, J. Xu, P. Chavez and S. Mukkamala, "Static Analyzer of Vicious Executables (SAVE)", 20th Annual Computer Security Applications Conference.

[9] "Anubis", Anubis.iseclab.org, 2018. [Online]. Available: http://anubis.iseclab.org/. [Accessed: 18- Feb- 2018].

[10] "Cuckoo Sandbox - Automated Malware Analysis", Cuckoosandbox.org, 2018. [Online]. Available: https://cuckoosandbox.org/. [Accessed: 18- Feb- 2018].

[11] J. T. Juwono, C. Lim and A. Erwin, "A Comparative Study of Behavior Analysis Sandboxes in Malware Detection," in International Conference on New Media (CONMEDIA), 2015.

[12] K. Rieck, P. Trinius, C. Willems and T. Holz, "Automatic analysis of malware behavior using machine learning", Journal of Computer Security, vol. 19, no. 4, pp. 639-668, 2011.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. Witten, "The WEKA data mining software", ACM SIGKDD Explorations Newsletter, vol. 11, no. 1, p. 10, 2009.

[14] "VirusTotal", Virustotal.com, 2018. [Online]. Available: http://www.virustotal.com. [Accessed: 12- Nov- 2017].

[15] I. Guyon and A. Elisseeff, "An Introduction to Variable and Behavioral attributes Selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[16] G. Sison, "Cerber Starts Evading Machine Learning - TrendLabs Security Intelligence Blog", Blog.trendmicro.com, 2018. [Online]. Available: https://blog.trendmicro.com/trendlabs-security-intelligence/cerber-starts-evading-machine-learning/. [Accessed: 18- Feb- 2018]

[17] H. Zhang, "The optimality of naive Bayes", Proc. 7th Int. Florida Artif. Intell. Res. Soc. Conf., pp. 562-567, 2004.

[18] J. R. Quinlan, C4.5: Programs for Machine Learning. San Francisco, CA, USA: Morgan Kaufman, 1993.

[19] E. Peterson Leif, "K-nearest neighbor", Scholarpedia, vol. 4, no. 2, 2009