TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 3
ISSN: 2054 - 7390

Unsupervised Machine Learning Techniques for Detecting Malware Applications in Wireless Devices

[*]Jackson Akpojaro[1], Princewill Aigbe[1], Ugochukwu Onwudebelu[2]

[1]Department of Mathematics and Computer Science, Western Delta University, Oghara, Delta Stat, Nigeria
[2]Department of Mathematics & Computer Science, Federal University, Ndufu, Alike Ikwo, Abakiliki, Ebonyi State,Nigeria

* jakpojaro@yahoo.com

ABSTRACT

It is no doubt that we are in the era of 'big data', and different machines and tools are being developed every day to enable users to effectively access, manipulate and process data to provide timely information needed for decision making. The situation has led to increasingly use of wireless devices including smartphones, tablets, pacemakers, etc., with different platforms. As professionals including doctors, engineers, scientists, artists, etc., use these devices in accessing, process and disseminating information services are available, so also malware attackers are strategizing. Hence the last one decade has witnessed constant literatures in the design and development of both supervised and unsupervised machine learning algorithms to checkmate malware applications in wireless devices. In this paper, we study the properties of unsupervised learning algorithms; in particular, we quantify the performance of these algorithms under two scenarios; using data sets from unknown attackers and data sets from known attackers. Our findings show that the recently γ-algorithm appears superior to the other unsupervised algorithms investigated.

Keywords: big data, wireless devices, malware, supervised algorithms, unsupervised algorithms.

1 INTRODUCTION

The use of wireless devices such as smartphones, tablets, pacemakers, etc. have become very popular among professionals because they provide convenience and easy access to timely information. As the functionalities and capabilities of these devices are increasing rapidly within

DOI: 10.14738/tmlai.23.206
Publication Date: 9th June 2014
URL: http://dx.doi.org/10.14738/tmlai.23.206

a short space of time with every new model, health experts and other users are beginning to rely on them to conduct diagnoses, businesses, interact with families and friends, play games, shopping, etc. Medical scientists have keyed in into this technology, using smartphones and wireless pacemakers for diagnoses, early testing, and electronic medication alerts with the aim of reducing prescribing errors [20]. A pacemaker is a small device that is placed in the chest or abdomen to help control abnormal heart rhythms, while the recently developed mobile phone application could help make monitoring conditions such as diabetes, kidney disease, and urinary tract infections much clearer and easier for both patients and health professionals, and could be used to slow or limit the spread of pandemics in the developing world [20].

As the technology is developing rapidly with increasing applications, so also security threats that target these applications are on the increase. In fact, malicious users and hackers are taking advantage of lack of standard security mechanisms to design mobile-specific malware that can access sensitive data, steal users' phone credit, or deny users' access to key functionalities in the device [18]. In the Juniper networks report on mobile threats, malware attacks have increased by 155 % across all platforms. In particular, devices with android platform had the highest malware growth rate [19].

To mitigate these security threats, the last one decade has witnessed a constant stream of literature on design and development of machine learning algorithms to detect malware in wireless devices. In this paper, we evaluate the performance of some the proposed and currently used unsupervised algorithms. In particular, we study their properties and characterize their performance under two scenarios: data sets from known attack and data sets from unknown attacks.

Summarizing, our main findings in this paper are:

- We study the properties of some unsupervised learning algorithms.
- We create different data sets and run the algorithms to produce experimental results.
- We find that the recently proposed γ-algorithm demonstrates some significant performance difference in both data sets with known attacks and data sets with unknown attacks.
- γ-algorithm is shown to be more promising than other unsupervised algorithms evaluated.

The reminder of this paper is organized as follows: Section 2 reviews some relevant background work. Section 3 discusses types of machine learning, while Section 4 describes the unsupervised algorithms we have evaluated. Section 5 presents our experimental results, while the results are discussed in Section 6. The paper is concluded in Section 7 with proposed research direction to formalize probabilistic models to quantify currently used supervised and unsupervised algorithms in static and dynamic environments with a view to determining allocation of scarce resources to promising algorithms at early design stage.

2 RELATED WORK

It is no doubt that we are in the era of 'big data', and different machines and tools are being developed to ensure that users have access to timely information to make decisions wherever they are. Many professionals have keyed in, doctors and other health experts use smartphones and other wireless devices to conduct medical diagnoses and tests. As these wireless devices with different operating platforms are increasing, developers of malware are strategizing. This has intensified and motivated research in machine learning to checkmate malware in different platforms. More heuristic methods have been proposed in this field to tackle specific problems. For instance, neural network models [21] have been inspired by the support vector classifiers [22, 23, 24]. Weston et al. [25] focused on the study of outliers from the perspective of the classification problem.

In the last decade, the field of semi-definite programming (SDP) has opened windows of opportunities for designing promising machine learning techniques. The consistency of researchers in this field has yielded a viable technology with efficient characteristics similar to quadratic programming [26]. Lanckreit et al. [27] demonstrated how SDP is used to optimize the kernel matrix for a supervised support vector machine (SVM). Xu et al. [28], De Bie et al. [29] developed new unsupervised and semi-supervised training techniques for SVMs based on SDP.

Several machine learning techniques have been applied for classifying applications with focus on detecting malware [30, 31]. Their goal is to classify applications into two main categories; malware or goodware. In [32, 33], the authors tried to classify applications by specifying the malware class (e.g., worms, Trojan, virus, etc.).

As the number of malware samples is exponentially increasing, particularly with Android platform, several techniques have been proposed to tackle the surge. Shabtai et al. [34] trained machine learning models, e.g., parsing *apk* which contains *xml* and counting xml elements, attributes or namespaces. They evaluated their model using information gained, fisher score, and Chi-square. They obtained 89% of accuracy classifying applications into two categories: tools and games. Recently, the γ-algorithm was proposed [11]. It is a graph-based outlier which assigns to every example the γ-score, which is the mean distance to the example k-nearest neighbors. Our experimental results show that this algorithm appears superior to other unsupervised algorithms in detecting malware in data sets involving known and unknown attackers.

The surveyed works provide the background for this paper. We study the properties of some unsupervised learning algorithms. In particular, we evaluate their performance under two scenarios: we create data sets with known attackers and data sets with unknown attackers. We

run the algorithms under these two situations and find that the γ-algorithm is more promising in detecting malware than the other algorithms investigated.

3 MACHINE LEARNING

Machine learning is a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict futures data, or to perform other kinds of decision making under uncertainty, for example, planning how to collect more data. Machine learning has been an active research area for more than a decade with focus on design and development of new algorithms that allow the computers to think and decide based on data [1].

Machine learning usually distinguishes three cases: supervised, unsupervised, and reinforcement learning. In supervised learning approach, the goal is to learn a mapping from input χ to output y , given a labeled set of input-output pairs [2], which is defined by a learning function,

$$\partial = \{(x_i, y_i)\}_{i=1}^{n} \quad (1)$$

where ∂ is the training data set, and n is the number of training examples. In its simplest form, each training inputs x_i could be features, attributes or covariates. More generally however, x_i could be a complex structured object, e.g., an image, email message, segment of application, sentence, etc. Variants of supervised learning algorithms include Bayesian Networks [3], Decision Trees [4], k-Nearest Neighbor (KNN) [5], and Support Vector Machine (SVM) [6].

In unsupervised learning problems, we have unlabeled inputs with a learning function defined as,

$$\partial = \{x_i\}_{i=1}^{n} \quad (2)$$

The aim is to find (or discover) interesting patterns or structures in the data set that can help to make informed decisions. In a purely unsupervised learning problem, agent cannot learn what to do because it has no specific output information as to what constitutes a correct action or a desirable state [7].

In reinforcement learning, rather than being told what to do, a reinforcement agent learns how to act or behave when given occasional reward or punishment signals. It is the most general of the three categories. The following subsection reviews unsupervised machine learning algorithms, explore how unlabeled data are clustered with a view to revealing some hidden structures to detect malware applications in wireless devices.

4 UNSUPERVISED ALGORITHMS

In unsupervised malware detection problems, we receive a large data set (e.g., emails) which contains both normal and buried malicious data within the data set [8]. Unsupervised algorithms have general features of able to process unlabeled data to detect malicious data that otherwise could not have been detected. In particular, some of these algorithms can automate the manual audit of data in forensic analysis by assisting analysts to focus on the suspicious elements in the data.

Unsupervised malware detection algorithms make two specific assumptions about the received data set: first, the number of normal instances outnumbers the number of malware instances. Secondly, the malicious instances are qualitatively different from the normal instances. Since the malware instances are both different from the normal instances and rare, they will appear as outliers in data set, which can be detected. In the light of this, we discuss the following unsupervised algorithms we have implemented in this work.

k-Means Clustering: The k-mean clustering algorithm is a variant of the partition clustering technique. It is a classical algorithm [9]. Its methodology is that after an initial random assignment to example k clusters, the centers of clusters are computed and examples are assigned to the clusters with the closest centers. This followed with several iterations until the cluster centers do not significantly change. Once the cluster assignment is fixed, the mean distance of an example to clusters is used as the score. There are simple approximations that speed up this algorithm considerably. For instance, one can project the data set and make cuts along selected axes, instead of using the arbitrary hyperplane divisions that are implied by choosing the nearest cluster center [10]. Details of how to speed up things are found in [10].

γ-Algorithm: The γ-algorithm [11] proposed recently is a graph-based outlier which assigns to every example the γ-score, which is the mean distance to the example k-nearest neighbors. It ignores the distances to the closer neighbors. More formally, a refined index that takes the distances to all k nearest neighbors is given thus [11];

$$\gamma(x) = \frac{1}{k}\sum_{j=1}^{k} \| x - z_j(x) \| \qquad (3)$$

where $\gamma(x)$ is x's average distance to its k nearest neighbors, $z_1(x),\ldots, z_j(x) \in \{x_{1,\ldots}, x_j\} \subset \Re^d$ (where $\Re^d$ refers to d-dimensional Euclidean space).

Divisive Hierarchical Clustering (DHC) - top-down: The divisive hierarchical clustering [12, 13] starts with one cluster of data set and each iteration split the most appropriate cluster until a stopping criterion such as a requested number k of clusters is achieved. Its implementation is described in [14].

Agglomerative Hierarchical Clustering (AHC) - bottom-up: An alternative to the top-down method for forming a hierarchical structure of clusters is the bottom-up approach called agglomerative clustering. This idea was proposed many years ago and has recently enjoyed a resurgence in popularity [10]. It starts with each data set in a separate cluster and at each iteration it merges the most similar clusters until the stopping criterion is met. Agglomerative clustering algorithms are categorized as single-linkage, complete-linkage, and average-linkage algorithms depending on the method each defines inter-cluster similarity.

The single-linkage algorithm defines the minimum distance between two clusters – the distance between their two closest numbers [10]. That is, it defines the similarity of two clusters C_i and C_j as the similarity of the least similar data $D_i \in C_i$ and $D_j \in C_j$ as;

$$S_{sk}(C_i, C_j) = \min_{D_i \in C_i, D_j \in C_j} |\cos(D_i, D_j| \quad (4)$$

where S refers to similarity and sk is single-linkage. Since this measure takes into account only the two closest members of a pair of clusters, the procedure is sensitive to outliers; the addition of a single new instance can radically alter the entire clustering structure.

The complete-linkage algorithm measures the maximum distance between the clusters. Two clusters are considered close only if all instances in their union are relatively similar. More formally, it defines the similarity of two clusters C_i and C_j as the similarity of the two most similar data $D_i \in Ci$ and $D_j \in C_j$ as;

$$S_{ck}(C_i, C_j) = \max_{D_i \in C_i, D_j \in C_j} |\cos(D_i, D_j| \quad (5)$$

where ck refers to complete-linkage. This measure which is also sensitive to outliers seeks compact clusters with small diameters. However, some instances may end up much closer to other clusters than they are to the rest of their own cluster.

The average-linkage algorithm is a measure which tries to avoid the problem inherent in centroid-linkage method since centroids are not instances and the similarity between them may be impossible to define. The average-linkage method defines the similarity of two data C_i and C_j as the average of pairwise similarities of the data from each cluster as;

$$S_{ak}(C_i, C_j) = \frac{\sum_{D_i \in C_i, D_j \in C_j} |\cos(D_i, D_j)|}{n_i n_j} \quad (6)$$

where ak is average-linkage, n_i and n_j are sizes of clusters C_i and C_j respectively.

Quarter-sphere Support Vector Machine (QSSVM): The quarter-sphere SVM [15] detects malicious data based on the idea of fitting a sphere onto the center of mass of data. An anomaly score is defined by the distance of a data point from the center of the sphere.

Choosing a threshold for the attack scores determines the radius of the sphere enclosing normal data points.

5 EXPERIMENTAL RESULTS

In this section, we evaluate the tradeoffs of the unsupervised learning algorithms briefly reviewed in Section 4. We evaluate the algorithms under two scenarios; first, we assume that the training and test data come from unknown attacks. Under the second scenario, we violated this assumption by taking data sets in which attacks unseen in training data are present in test data. Based on these, we created 6 data sets 200, 300, 400, 500, 1000, 2000 android applications (see Table 1). First we extract the necessary features from the applications to identify known malware (e.g., Adware, worm, Trojan, virus, rootkit, etc.), while in the second case, we pretend that the data sets contained malicious and normal data without classification.

Table 1: Datasets

Data Set #	No. of Samples	No of Features
1	200	120
2	300	145
3	400	148
4	500	175
5	1000	250
6	2000	318

We find that as the number of the samples increases, the performance difference of the algorithms becomes slightly significant. Hence we chose to provide the experimental results of the data set with 2000 samples (see Table 3 and Table 4). The evaluation metrics, true positive ratio (TPR), false positive ration (FPR), accuracy, and area under the ROC curve (AUC) are formalized and discussed in [16]. We use these formulae to obtain our experimental results as shown in Table 2 and Table 3 respectively.

Table 2: Obtained result for known attacks.

Algorithm	TPR	FPR	AUC	Accuracy (%)
γ -algorithm	0.96	0.10	0.98	98.33%
k -Means Clustering	0.90	0.08	0.86	91.12%
DHC	0.91	0.11	0.92	91.01%
AHC (single-linkage)	0.93	0.09	0.93	93.04%
QSSVM	0.89	0.19	0.85	91.11%

Table 3: Obtained result for unknown attacks

Algorithm	TPR	FPR	AUC	Accuracy (%)
γ -algorithm	0.98	0.08	0.99	99.54%
k -Means Clustering	0.89	0.10	0.86	91.11%
DHC	0.91	0.11	0.92	91.12%
AHC (single-linkage)	0.93	0.11	0.93	93.04%
QSSVM	0.90	0.19	0.85	91.11%

6 DISCUSSIONS

As presented in Table 2 and 3, the algorithms exhibit no significant difference in performance between known and unknown attacks except the γ-algorithm. This is because the two data sets differ merely in the set of attacks contained in them. However, only the γ-algorithm is shown to be promising in both data sets. It is 98% (FPR) better in detecting malware for unknown attacks as against 96% (FPR) for data sets containing known attacks. More generally, the γ-algorithm is not only significant in performance (accuracy (%)) in both data sets, but it also better than the other algorithms tested. The k-means clustering has the least TPR of 0.89 (see Table 3), but compares favorably with the QSSVM algorithm. Our results corroborate the work of Borja Sanz et al., [16], Pavel Laskov et al. [17], and Stafan Harmeling et al. [11].

The limitation of our results is that they are based on 2000 samples. We believe more significant performance differences among the algorithms could be revealed in larger samples (e.g., between 100,000 and more) that require more computational time and other resources. For brevity, we leave this investigation to others.

7 CONCLUSIONS AND FUTURE WORK

We have presented an experimental framework in which the unsupervised learning algorithms are evaluated in detection of malware in wireless devices. Our experimental results demonstrate no major significant performance difference in both unknown and known data sets except the γ-algorithm, which is not only superior to the other algorithms but also exhibits performance difference in both data sets. We find that as the data sets get larger, all the algorithms exhibit some performance differences; hence we chose to present the results for 2000 samples. We believe that with larger samples, e.g., 100,000 or more data sets, the algorithms would exhibit more significant results that could be further used to characterize them.

In our future research, we plan to formalize analytic model to quantify both supervised and unsupervised learning methods using common metric(s). In particular, we will analyze and evaluate these algorithms in both static and dynamic environments. In the dynamic scenario, we plan to introduce probabilistic models to enable us determine in real time the relative performance of these algorithms in detecting malware and also measure what new (or upgraded) algorithms claimed to be contributing at development stage. In doing so, scarce resources could be channeled to only new algorithms that demonstrate promising contribution to the current state of the art in machine learning.

REFERENCES

[1]. Bishop, C., Pattern recognition and machine learning, Springer New York, 2006.

[2]. Murphy, K., Machine learning: A Probabilistic perspective, MIT Press, Cambridge, MA, 2012.

[3]. Pearl, J., Reverend Bayes on inference engines: A distributed hierarchical approach, In Proceedings of National Conference on Artificial Intelligence, 1982, pp. 133-136.

[4]. Quinlan, J., Induction of decision trees, Machine learning 1(1), 1986, pp. 81-106.

[5]. Fix, E., Hodges, J. L., Discriminatory analysis: Nonparametric discrimination: Small sample Performance, Technical Report Project 21-49-004, Report number 11, 1952.

[6]. Vapnik, V., The nature of statistical learning theory, Springer, 2000.

[7]. Russell, S. and Norving, P., Artificial Intelligence: A Modern approach, 2nd Edition, Prentice Hall, Upper Saddle River, New Jersey 07458, 2003.

[8]. Leonid, P., Leazar, E., and Salvatore, J., Instruction Detection with unlabeled Data using Clustering. In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA 2001) Philadelphia, PA, 2001.

[9]. Duda, R., Hart, P., and Stork, D., Pattern Classification, Second Edition, John Willey & Sons, 2001.

[10]. Ian, H., Eibe, F., and Hall, M., Data Mining: Practical Machine Learning Tools and Techniques, Third Edition, Morgan Kaufman Publishers, Burlington, MA 01803, USA, 2011.

[11]. Harmeling, S., Dornhege, G., Tax, D., Meinecke, F., and Miller, K., From Outliers to Prototypes: Ordering Data, Neurocomputing Vol. 69, pp. 1608-1618, 2006.

[12]. Jain, A., Murty, M., and Flynn, P., Data clustering: A review, ACM Computing Surveys, Vol. 31, No. 3, pp. 264–323, September 1999.

[13]. Berkhin, P., Survey of clustering data mining techniques, Research paper, Accrue Software, http://www.accrue.com/products/researchpapers.html, 2002.

[14]. Kaufman, L. and Rousseeuw, P., Finding groups in data, Wiley, New York, NY, 1990.

[15]. Laskov, P., Schafer, C., and Kotenko, I., Intrusion Detection in Unlabeled Data with Quarter-sphere Support Vector Machines. In proceedings DIMVA, pp. 71-82, 2004.

[16]. Borja, S., Igor, S., Javier, N., Carlos, L., Inigo, A., and Pablo, G., MADS: Malicious Android Applications Detection through String Analysis. Lecture Notes in Computer Science, Vol. 7873, pp. 178-191, 2013.

[17]. Laskov, P., Diissel, P., Schafer, C., and Rieck, K., Learning Instruction Detection: Supervised or Unsupervised?. Fraunhofer-FIRST IDA, 12489 Berlin, Germany, 2006.

[18]. Zami, A., and Zawi, W., Permission-Based Android Malware Detection. International Journal of Scientific and Technology Research, Vol. 2, Issue 3, pp. 228-234, 2013.

[19]. Juniper Networks: 2011 Mobile threats report, February 2012.

[20]. Muanya, C., Smartphones, wireless pacemakers, turned into portable medical devices. The Guardian, p.31, Thursday, March 27, 2014.

[21]. Marshland, S., Online novelty detection through self-organization with application to inspection robots. Ph.D. Thesis, University of Manchester, 2001.

[22]. Scho, B., J. Shawe-Taylor, P., Smola, A., and Williamson, R., Estimating the support of a high-dimensional distribution, Neural Computation Vol. 13 Issue 7, pp. 1443-1471, 2001.

[23]. Campbell, C., and Bennett, K., A linear programming approach to novelty detection. Advances in Neural Information Processing Systems, Vol. 13, MIT Press, Cambridge, MA, pp. 395-401, 2001.

[24]. Tax, D., and Duin, R., Uniform object generation for optimizing one-class classifiers, J. Mach. Learn. Research, pp. 155-173, 2001.

[25]. Weston, J., Chapelle, O., and Guyon, I., Data cleaning algorithms with applications to micro-array experiments, Technical Report, BIOwulf Technologies, 2001.

[26]. Boyd, S., and Vandenberghe, L., Convex Optimization, Cambridge, U. Press, 2004.

[27]. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., and Jordan, M., Learning the kernel matrix with semidefinite programming, Journal of Machine Learning Research, 2004.

[28]. Xu, L.; Neufeld, J.; Larson, B.; and Schuurmans, D., Maximum margin clustering. In Advances in Neural Information Processing Systems 17 (NIPS-04), 2004.

[29]. De Bie, T., and Cristianini, N., Convex methods for transduction, In Advances in Neural Information Processing, 16 (NIPS-03), 2003.

[30]. Santos, I., Laorden, C., and Bringas, P., Collective classification for unknown malware detection, In Proceedings of the 6th International Conference on Security and cryptography (SECRYPT), 2011.

[31]. Y. Ye, Y., Wang, D., Li, T., and Ye, D., IMDS: Intelligent malware detection system, In Proceedings of the 13th ACM SIGKDD International conference on Knowledge discovery and data mining, ACM, pp. 1043-1047, 2007.

[32]. Rieck, K., Holz, T., Willems, C., Dussel, P., and Laskov, P., Learning and classification of malware behavior, In Proceedings of the 2008 Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), pp. 108-125, 2008.

[33]. Tian, R., Batten, L., Islam, R., and Versteeg, S., An automated classification system based on the strings of trojan and virus families, In Malicious and Unwanted Software MALWARE), 2009 4th International Conference on IEEE, pp. 23-30, 2009.

[34]. Shabtai, A., Fledel, Y., and Elovici, Y., Automated Static Code analysis for classifying Android applications using machine learning," 2010 International Conference on Computational Intelligence and Security, pp. 329–333, 2010.

