Machine Learning with Adaptive Rate Processing for Power Quality Disturbances Identification

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 194 (2021) 272–279

Procedia Computer Science

www.elsevier.com/locate/procedia

18th International Learning & Technology Conference 2021

A Comparative Evaluation of Ensemble Classifiers for Malicious Webpage Detection

Abdulhamit Subasi[1], Mohammed Balfaqih[2], Zain Balfagih[1] Khaled Alfawwaz[3]

[1]*Computer Science Department, Effat University, Jeddah, Saudi Arabia.*

[2]*Department of Computer and Network Engineering, University of Jeddah, Jeddah, Saudi Arabia*

Department of Information Systems, King Abdulaziz University, Saudi Arabia

Abstract

Malicious webpage is developed or manipulated to be used as attack tool where it is considered as one of the main reasons of Internet criminal activities. Thus, it is essential to detect such webpages and prevent end users form accessing it. The conventional malicious webpages detection techniques are based on searching through a blacklist that contains a list of webpages classified as malicious from the perspective of users. However, these techniques have high false-negative rates especially with aforesaid sophisticated attacks due to technical and computational limitations. Hence, machine learning techniques have been employed to classify webpages by systemically analyzing set of features that reflect the characteristics of a malicious webpage. This paper compares the prediction accuracy of several machine learning classification algorithms and ensemble techniques. A data set of 5000 instances of URLs, with 189 different features are used in the comparative study. The results show that the most accurate classification technique in MultiBoost and Adaboost is Support Vector Machine (SVM), while K-Nearest Neighbor (k-NN) technique in bagging and random subspace.

Keywords: Biometrics, Face recognition, Face detection, Correlation Coefficient, Recognition rate, Mobile API vision, Normalized Features.

E-mail address: absubasi@effatuniversity.edu.sa [a]

10.1016/j.procs.2021.10.082

1. Introduction

Web security threats are increasing daily due to the open nature of Internet in which harmful webpages could be accessed as safe websites. The speed of Internet and its usage increased during the last two decades. Developers have increased the usage of JavaScript, images, and other elements to offer different convenient web services such as Web 2.0-based services, multimedia sharing, and albums. According to [1], the evolution of web to create interactive animated websites with the use of Dynamic HTML (DHTML) allows hackers to embed malicious HTML code in normal webpages to infect the victim (person or organization) while browsing the webpages through obfuscation or transformation.

Using antivirus to detect such attacks is insufficient since the blacklist method is the most common deployed technique to detect malicious webpages. Blacklists are based on a database of webpages that have been classified as malicious in the past. The detection process occurs in a binary executable based on pattern matching techniques and relying on the frequency of the available data in the signature database [2]. However, it is difficult to preserve an exhaustive list of malicious webpages especially due to their daily increment. In addition, attackers avoid the blacklist using creative techniques such as modifying the webpage address to appear as a benign webpage.

In the last decade, machine learning techniques have been proposed to address these issues. To detect malicious webpages, a set of webpage addresses are used as training set to learn classifying webpages into malicious or benign based on the statistical properties of webpage features. The quality of features representation of webpages plays a key role in the accuracy of machine learning classifiers. Unlike the blacklist method, machine learning generalizes

malicious detection solutions with any new webpage addresses. Several machine learning techniques could be used to classify webpages after training

This paper compares the accuracy of six classifiers on a malicious data set. The classifiers include K-Nearest Neighbor (k-NN), Support Vector Machines (SVM), Classification and Regression Trees (CART), C4.5 Tree, Logit boost Alternating Decision tree (LADTree), and Naive-BayesTree (NBTree). The utilized dataset consists of 189 different features and 5000 different instances, each instance is classified as benign or malicious, to indicate whether the given URL is malicious or not. The rest of the paper is organized as follows: Section 2 discusses the related work. In Section 3, the studied classification techniques are described. The data set construction, and the evaluation metrics are discussed in Section 4. The results and discussion are presented in Section 5. Finally, Section 6 concludes the work.

2. A literature Review

The most common approach that have been developed in the literature to protect web users when clicking URLs is called web filtering through blacklisting [3]. The idea is to use a third-party service to mark the bad websites according to factors including user feedback, web crawling and analyzing site content for example using classifiers (e.g., decision tree) [3]. Although such third-party systems have low query overhead, they can only offer modest accuracy because it is very hard to create a comprehensive and up-to-date blacklist. This means that a user may click an infected before finding it in the blacklist [4].

To address the limitations of blacklisting method, machine learning techniques have been proposed for detecting malicious webpages. The authors in [4] proposed a machine learning approach to detect malicious DHTML codes in webpages. The features that are considered in classification process are (1) DHTML document length (2) word length, (3) word count, (4) the link to remote source of scripts, (5) asymmetric HTML tags, and (6) Count native JavaScript functions used. NBTree and Decision Trees were employed and fed with features and training set as input to classify the webpages. Logistic regression was used to classify URLs in [5] over eighteen features including Google Page Rank, red-flag keywords, and the scores guidelines of Web quality.

In [6], SVM was utilized as a binary classification technique for high dimensional data, where it was trained with large data set to increase the percentage of true positive. The authors in [7], employed several algorithms for webpages classification including, (1) Perceptron that gives a linear classifier to update weight vector when wrong classification took place. The algorithm gives a simple rule to update the error. It can undercompensate or overcompensate for errors in many cases [8], (2) Logistic Regression with Stochastic Gradient Descent which uses gradient descent to optimize the objective function which represents the total sum of the sample individual objective functions, (3) nonlinear classification using online kernel-based Forgetron algorithm [9].

The main problem of detecting attacks in social networks is that sometimes the URLs used seems to be benign because it refers to blogs or websites that may not look malicious, so the features should be changed in the case of social networks. There are large set of feature that were used in [10] including (1) count of dashes in hostname is used as a lexical feature, since benign URLs rarely contains dashes [11], (2) Longest Domain Label in hostname is used as a lexical feature, since legitimate websites always use longer, short names that represents the brand name, while those of malicious websites are always longer (3) Google Domain rank was suggested as a strong feature, if the website is ranked high, it's benign otherwise, it may be malicious [12]. The proposed approach suggests the use of Bayesian

classification method described before. The training data set were collected from popular Facebook fan pages from distinct various geographical areas in the period from July 2010 to February 2011. The authors in [13] compared many different classifiers over a text corpus (large and structured set of texts stored and processed electronically) of phishing emails, using the frequency of the 43 most-popular words in the corpus as features. However, to the best of the authors knowledge, there is no performance comparative study on classification techniques in detecting malicious webpages.

3. 1. Machine Learning Classification Algorithms

This section provides a brief description about the different classification algorithms used in classifying the given training and test sets. This includes k-NN, SVM, C4.5, CART, LADTree, and NBTree.

K-Nearest Neighbor (k-NN)

Nearest-neighbor classifiers compares a given test tuple with similar training tuples that has n attributes. The training tuples are stored in an n-dimensional pattern space in which each tuple represents a point. Accordingly, for an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the k training tuple that are nearest neighbors of that tuple. The term nearest is measured with a distance metric such as Euclidean distance. The most common class among the k-nearest neighbors is assigned to the unknown tuple, for example, the closest class of the training tuple in pattern space is assigned to the unknow tuple when k=1. The best k value which is the number of neighbors is determined experimentally by estimating the error rate of the classifier if k=1 and repeating the process with k increment to check one more neighbor. The k value that produces the minimum error rate is selected. Generally, increasing the value of k increases the number of training tuples. The main limitation of Nearest-neighbor classifier is the poor accuracy especially with noise and irrelevant attributes because the distance-based comparisons are used which assigns equal weight to each attribute [14].

Support Vector Machine (SVM)

SVM algorithm uses a nonlinear mapping that transforms the original training data into a higher dimension and tries to find the linear optimal separating hyperplane using support vectors and margins. SVM is extremely slow but has high accuracy due to its ability to model complex nonlinear decision boundaries. The process of SVM with two classes that are linearly separable is labeling classes with one of two values either +1 or -1. However, if there is an infinite number of separating lines that could be drawn, the best line with minimum classification error must be found. In other word, if we generalize to n dimensions, the best hyperplane must be found. This occurs by searching the maximum marginal hyperplane which is the hyperplane with the largest margin [14].

C4.5 Decision Tree

The decision tree program C4.5 was devised by Ross Quinlan over a 20-year period beginning in the late 1970s. C4.5 decision eliminates tests for which almost all the training examples have the same outcome. Such tests are often of little use. Consequently, tests are not incorporated into the decision tree unless they have at least two outcomes that have at least a minimum number of instances. The default value for this minimum is 2, but it is controllable and should perhaps be increased for tasks that have a lot of noisy data. Another heuristic in C4.5 is that candidate splits on numeric attributes are only considered if they cut off a certain minimum number of instances. C4.5 includes Minimum description length (MDL)-based adjustment to the information gain for splits on numeric attributes. More specifically, if there are S candidate splits on a certain numeric attribute at the node currently considered for splitting, log2(S)/N is subtracted from the information gain, where N is the number of instances at the node. This heuristic is designed to prevent overfitting. The information gain may be negative after subtraction, and tree growing will stop if there are no attributes with positive information gain—a form of prepruning [15].

Classification and Regression Trees (CART)

CART is a very successful decision tree induction method [16]. It is a nonparametric method that builds binary trees from continuous and discrete features of the data. The analysis of continuous features considers all possible binary splits into intervals $(-\infty, a)$ and (a, ∞), while all possible splits of symbols set into two disjoint and complementary subsets is considered. Each tree node is labeled with the class label dominating in the node. The possible misclassification cost must be considered in the decisions; thus, surrogate splits technique is employed to handle missing data values. The data object is passed to another test if it does not describe a value necessary for the original test of a tree node. Consequently, the new test exploits another feature to generate a similar split to the original test.

Logit boost Alternating Decision tree (LADTree)

LADTree is a decision tree for multi-class classification using the LogitBoost strategy to induce alternating decision tree. The predicted values are vector representing the sum of the responses of ensemble classifiers on instance over the classes. Two-class symmetric logistic transformation is generalized and used to compute the estimation of class probability. At each iteration, a single attribute test is conducted as splitter node for the tree. The working response and weights on a per-class basis are stored with each training instance. To make the working response suited to the mean value of the instances, the least-squares value is minimized between them. Hence, tests are chosen to be added to the tree if it will maximize the gain by dropping the least squares calculation [17].

Naive-Bayes Tree (NBTree)

NBTree is same as the classical recursive partitioning schemes with the difference that the leaf nodes act as Naive-Bayes categorizers. NBTree algorithms have high classification accuracy on small database but not for some larger database. A split is considered significant if it has at least 30 instances in the node and the relative error reduction is greater than 5%. A split utility is the weighted sum of the nodes which is proportional to the number of instances of a node. The standard entropy minimization technique is used to choose the threshold for continuous attributes. A node utility is calculated based on data discretization and 5- Direct use of cross-validation computation. Naive-Bayes can be cross-validated in time when it is linear in the number of instances, number of attributes, and number of label values. This is because, if the data is discretized, the instances can be removed, the counters can be updated to classify them, and repeat the process for a different set of instances [18].

Ensemble Techniques

Ensemble techniques create multiple models and combine them to produce improved results. These techniques usually produce more accurate solutions than a single model would. Combining the decisions of different models means amalgamating the various outputs into a single prediction. This occurs by taking a vote (or weighted vote) for classification case and by calculating the average (or weighted average) for numeric prediction.

Bagging

Bagging combines the trees by having them vote on each test instance where a class is considered correct if it receives more votes than any other. In general, the more votes are taken the more reliable the predictions become. The trees are built for the new discovered training sets, and their predictions participate in the vote as well. In particular, the combined classifier will seldom be less accurate than a decision tree constructed from just one of the datasets. To neutralize the instability of learning methods, bagging alters the original training data by deleting some instances and replicating others instead of sampling a fresh, independent training dataset each time. Instances are randomly sampled, with replacement, from the original dataset to create a new one with the same size. This sampling procedure inevitably replicates some of the instances and deletes others. The datasets generated by resampling are different from one another but are certainly not independent because they are all based on one dataset. However, it turns out that bagging produces a combined model that often performs significantly better than the single model built from the original training data and is never substantially worse [19].

AdaBoost

Similar to bagging, boosting uses voting (for classification) or averaging (for numeric prediction) to combine the output of individual models. Although it combines models of the same type, it is iterative in which each new model is influenced by the performance of those built previously. Boosting make the new models to handle instances that are handled incorrectly by earlier ones by assigning greater weight to those instances. Moreover, boosting weights a

model's contribution by its confidence rather than giving equal weight to all models. AdaBoost is one of boosting methods that is designed specifically for classification. It can be applied to any classification learning algorithm. The method begins by assigning equal weight to all instances in the training data. It then calls the learning algorithm to form a classifier for this data and reweights each instance according to the classifier's output. The weight of correctly classified instances is decreased, and that of misclassified ones is increased. This produces a set of "easy" instances with low weight and a set of "hard" ones with high weight. In the next iteration—and all subsequent ones—a classifier is built for the reweighted data, which consequently focuses on classifying the hard instances correctly. Then the instances' weights are increased or decreased according to the output of this new classifier. As a result, some hard instances might become even harder and easier ones even easier; on the other hand, other hard instances might become easier, and easier ones harder—all possibilities can occur in practice. After each iteration, the weights reflect how often the instances have been misclassified by the classifiers produced so far. By maintaining a measure of "hardness" with each instance, this procedure provides an elegant way of generating a series of experts that complement one another [15].

MultiBoosting

MultiBoosting technique combines both Bagging and AdaBoost techniques to obtain greatest benefit since these techniques achieve great effect individually. Such benefit can be obtained even with quite small committees because most of the effect of each technique is obtained by the first few committee members. The techniques could be combined by developing two sub-committees, in which one contains the members formed by boosting and one contains the members formed by bagging. The default number of sub-committees for a single run, and the size of those sub-committees are set to be a single committee size T, and the number of sub-committees and the size of those sub-committees to $\sqrt{T}$. The target final sub-committee member index is also set by setting an index for each member of the final committee starting from one. This will allow the premature termination of boosting one sub-committee if the error is too great or too low. MultiBoost has computational advantage over Bagging and AdaBoost due to the parallel computation capability [20].

Random Subspace

The Random Subspace technique combines the classification techniques with a modification in the feature space of the training data. In the training sample set $\boldsymbol{X} = (\boldsymbol{X}1, \boldsymbol{X}2, \ldots, \boldsymbol{X}n)$, if each training object $\boldsymbol{X}i = (i = 1, \ldots, n)$ is a p-dimensional vector $\boldsymbol{X}i = (\boldsymbol{X}i1, \boldsymbol{X}i\ 2, \ldots, \boldsymbol{X}i\ p)$, described by p features, then rp features could be randomly selected from the p-dimensional data set X. Accordingly, r-dimensional random subspace can be obtained from the original p-dimensional feature space in which the modified training set will consist of r-dimensional training objects. Hence, the classifiers can be constructed in the random subspaces and combined by simple majority voting in the final decision rule. The Random Subspace technique uses random subspaces to enhance both constructing and aggregating the classifiers. The combined decision of classifiers with small sample size problem is superior to a single classifier in the complete feature space [21].

The comparative Evaluation of Ensemble Classifiers

The malicious database and evaluation metrics that are used to compare the performance of the discussed machine learning classification algorithms and ensemble techniques. The features that are used in the study are (1) the number of dots in the website URL, since most benign URLs do not contain dots, (2) WHOIS properties, which represents the date of expiration, registration, and update, and (3) Domain name properties (DNS), for example the time-to-live (TTL) record value for the DNS associated with the website hostname [7, 22]. The malicious database and the evaluation metrics used in the study are discussed.

4. Malicious Database

The utilized malicious database for training and testing the classifiers consists of 189 different features named F1 to F189 and 5000 different instances. Each instance is classified as benign or malicious to indicate whether the given URL is malicious or not. The given class is priori defined to ease the process of training the given set. WEKA 3.6 was used to classify the database. The classification performance is measured by comparing the prediction of the classifiers on a dataset S with the true class labels of the instances from this dataset. The true class is usually separated from the training set and called the validation set or test set [23]. In our evaluation, S refers to the arbitrary

dataset and Q to a validation/test set separate from the training set. To determine training and validation set pairs from a dataset S, it is divided randomly into K parts in which each part is also divided randomly into two parts for training and validation. K is usually set to 10 or 30. Since our dataset is small, the crossvalidation process is conducted by repeating use of the same data split differently to ensure that the overlap between different sets is small as possible and according the error

estimation is robust. Moreover, the stratification process is conducted to ensure that classes are represented in the right proportions without disturbing the class prior probabilities [24].

Evaluation Metrics

Several performance metrics could be used for classification performance evaluation. However, they mostly depend on the four prediction cases in the classification process: (i) true positive if the prediction and the example are positive,

(ii) false negative if the prediction was negative while the example is positive, (iii) true negative if the prediction and the example are negative, and (iv) false positive if the prediction is positive and the example is negative example as positive [24].

The most general method to compare between techniques is comparing the classification performance without focusing on a class. Accordingly, the most used performance metric is accuracy which does not distinguish between the number of correct labels of different classes [25]:

$$\text{Accuracv} = (t\_p+t\_n)/(tp + fp + fn + tn) \quad (1)$$

The performance metric F-score is also considered which is calculated as follows

$$F - measure=((B\text{^}2+1)*\text{precision}*\text{recall})/(B\text{^}2*\text{precision}*\text{recall}) \quad (2)$$

It is consistently stable when $\beta = 1$. It pretends accuracy if $\beta > 1$, and recall if $\beta < 1$. The precision represents the relation between true positives and false positives [25]

$$precision = t\_p/(t\_p+f\_p) \quad (3)$$

On the other hand, recall represents the relation between true positives and false negatives [25]

$$\text{recall}=t\_p/(t\_p+f\_p) \quad (4)$$

The third performance metric is ROC area which refers to the output of classification between most positive and most negative plotted by the ROC curve. It is an established measure in learning through imbalanced data sets and asymmetric cost functions according to the varied usage in cost/benefit decision study [26].

5. Results and Discussion

The results were obtained from the given training and testing data set and using 10- cross fold validation as a test option. The results are illustrated in Table 1. As you can notice, the best result was obtained by KNN algorithm using random subspace technique with accuracy rate 89.24%. In addition, the best result in case of bagging is also using k-NN with a rate of 88.57%. In MultiBoosting and AdaBoost, the best classification algorithm was obtained by SVM with accuracy rate 89.15% and 89.05%, respectively. In rotation forest case, the best result was obtained by C4.5 algorithm with accuracy rate of 88.57%.

Table 1. Results of MultiBoost, Adaboost, Bagging and Random Subspace methods

		K-NN	SVM	C4.5	CART	LADTree	NBTree
MultiBoost	**ROC Area**	0.902	0.868	0.884	0.907	0.881	0.921
	F-Measure	0.888	0.89	0.835	0.857	0.805	0.874

	Accuracy (%)	88.95	89.15	83.48	86.17	81.84	87.61
AdaBoost	**ROC Area**	0.882	0.862	0.895	0.903	0.894	0.929
	F-Measure	0.85	0.889	0.796	0.861	0.831	0.868
	Accuracy (%)	84.92	89.05	79.15	86.36	83.67	86.74
Bagging	**ROC Area**	0.929	0.873	0.853	0.868	0.872	0.914
	F-Measure	0.884	0.894	0.821	0.861	0.801	0.837
	Accuracy (%)	88.57	88.18	83.09	86.55	81.36	84.05
Random Subspace	**ROC Area**	0.928	0.887	0.858	0.89	0.865	0.909
	F-Measure	0.891	0.869	0.816	0.857	0.807	0.842
	Accuracy (%)	89.24	87.42	82.61	86.07	82.04	84.63

6. Conclusion

This paper presented a comparative evaluation of several machine learning algorithms and ensemble techniques for classification, to check malicious webpages. The classification algorithms are k-NN, SVM, C4.5, CART, LADTree, and NBTree, while the ensemble techniques are Bagging, AdaBoost, MultiBoosting and Random Subspace techniques. The techniques have been employed to classify webpages by systemically analyzing set of features that reflect the characteristics of a malicious webpage. A data set of 5000 instances of URLs, with 189 different features were used in the study. The paper considered the accuracy, F-score, and ROC area as performance evaluation metrics. The results illustrated that the most accurate classification technique in MultiBoost and Adaboost is SVM, while k-NN technique in bagging and random subspace.

8. Conclusion

This paper presented a comparative evaluation of several machine learning algorithms and ensemble techniques for classification, to check malicious webpages. The classification algorithms are k-NN, SVM, C4.5, CART, LADTree, and NBTree, while the ensemble techniques are Bagging, AdaBoost, MultiBoosting and Random Subspace techniques. The techniques have been employed to classify webpages by systemically analyzing set of features that reflect the characteristics of a malicious webpage. A data set of 5000 instances of URLs, with 189 different features were used in the study. The paper considered the accuracy, F-score, and ROC area as performance evaluation metrics. The results illustrated that the most accurate classification technique in MultiBoost and Adaboost is SVM, while k- NN technique in bagging and random subspace.

References

1) McGraw, G., & Morrisett, G. (2000). Attacking malicious code: A report to the Infosec Research Council. *IEEEsoftware*, *17*(5), 33.
2) Christodorescu, M., & Jha, S. (2004). Testing malware detectors. *ACM SIGSOFT Software Engineering Notes*, *29*(4), 34-44.
3) Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009, June). Identifying suspicious URLs: an application of large-scale online learning. In *Proceedings of the 26th annual international conference on machine learning* (pp.681-688). ACM.
4) Hou, Y. T., Chang, Y., Chen, T., Laih, C. S., & Chen, C. M. (2010). Malicious web content detection by machinelearning. *Expert Systems with Applications*, *37*(1), 55-60.
5) Garera, S., Provos, N., Chew, M., & Rubin, A. D. (2007, November). A framework for detection and measurementof phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode* (pp. 1-8). ACM.
6) Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

7) Fette, I., Sadeh, N., & Tomasic, A. (2007, May). Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web* (pp. 649-656). ACM.
8) Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.
9) Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research*, *7*(Mar), 551-585.
10) Chen, C. M., Guan, D. J., & Su, Q. K. (2014). Feature set identification for detecting suspicious URLs using Bayesian classification in social networks. *Information Sciences*, *289*, 133-147.
11) Zhang, Y., Hong, J. I., & Cranor, L. F. (2007, May). Cantina: a content-based approach to detecting phishing websites. In *Proceedings of the 16th international conference on World Wide Web* (pp. 639-648). ACM.
12) Guan, D. J., Chen, C. M., & Lin, J. B. (2009, August). Anomaly based malicious url detection in instant messaging. In *Proceedings of the joint workshop on information security (JWIS)*.
13) Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007, October). A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit* (pp. 60-69). ACM.
14) Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
15) Hall, M., Witten, I., & Frank, E. (2011). Data mining: Practical machine learning tools and techniques. *Kaufmann, Burlington*.
16) Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.
17) Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., & Hall, M. (2002). *Multiclass Alternating Decision Trees.Machine Learning: ECML 2002, 161–172.* doi:10.1007/3-540-36755-1_14
18) Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd* (Vol. 96,pp. 202-207).
19) Witten, I. H., & Frank, E. (2002). Data mining: practical machine learning tools and techniques with Javaimplementations. *Acm Sigmod Record*, *31*(1), 76-77.