Machine Learning Classifier to Detect Malicious Websites

Gurunatha Srinivasa Bhatta[1], Sathisha G [2]
Department of Computer Science Engineering
[1,2] Atria Institute of Technology

Abstract- *A risk that exists in web is that the access of websites with malicious content, as a result of they might be open doors for cyber-crimes or be the mechanism to transfer files so as to have an effect on organizations, persons and therefore the setting. What is additional, the attack registers through websites are a part of cyberattacks reports throughout the last years; this info includes attacks created by the presently risks found in new technologies, like the IoT. Due the pc security quality, studies have been operating in to use machine learning algorithms to spot net malicious content. On the net, users typically visit unknown websites. However, malicious websites area unit a big threat to the Internet users. Malicious websites implant malwares into users computers while not their data, through drive-by downloads technology. A naive user may simply fall victim of such attack. With enhanced use of net browsing, net security is a vital issue and a vital analysis topic. The motivation of this study is to classify malicious web-sites from benign ones from their URL options. If it might be finished high exactness, particularly with low false settle for rate, automatic blocking of suspicious URL at the user web site are going to be doable. We collect URL knowledge for an outsized variety of glorious benign as well as malicious websites. This article explores the appliance of an information analysis method through a framework that features dynamic, static analysis, updated websites and an occasional interaction shopper protea in order to classify an internet site. what is more, it evaluates the capability of the classification of four machine learning through the knowledge analyzed.*

Keywords- Random forest, Classification, Internet, Machine Learning

I. INTRODUCTION

Number of internet sites is growing exponentially as a result of a massive growth in net applications and services, such as social networking, blogs, and e-commerce. the web has already become a vital a part of our lifestyle. Because of high-speed net association and wireless hot-spots available everyplace, the recognition of the web has naturally attracted miscreants. World Wide net has become a platform. to support a good vary of net criminal activities such as spam-advertising, money fraud, and malware implanting [3] [8] [9] [13] [15]. They started varied forms of malicious websites to bait their victims. although motivation and activities are completely different, the common target is to draw in careless users to these pretend websites which might be accessed via links through email, net search result, or links from websites redirection. The advent of recent communication technologies has had tremendous impact within the growth and promotion of companies spanning across several applications as well as online-banking, e-commerce, and social networking. In fact, in today's age it is nearly obligatory to own an internet presence to run a successful venture. As a result, the importance of the globe Wide net has ceaselessly been increasing. sadly, the technological advancements come back as well as new refined techniques to attack and scam users. Such attacks include scallywag websites that sell counterfeit product, financial fraud by tricking users into revealing sensitive data which eventually cause stealing of cash or identity, or even installing malware within the user's system.

Several studies within the literature tackle this drawback from a Machine Learning stand. That is, they compile a listing of URLs that are classified as either malicious or benign and characterize every URL via a collection of attributes. Classification algorithms area unit then expected to be told the boundary between the decision categories. De las Cuevas et. al. [3] reported classification rates regarding 96% that climbed up to ninety seven with a rough-set-based feature selection preprocessing step that reduced the first twelve features to nine. The labeling of every URL was done once a collection of security rules settled by the Chief Security Officer (CSO) in a company. This resulted in associate unbalanced classification problem that was treated via under sampling. In total, 57,000 URL instances were thought-about once removing duplicates. The authors detected associate improvement over the results earned in their previous work [4].

Kan and Thi [5] classified web content not by their content but victimization their URLs, that is far quicker as no delays are incurred in taking the page content or parsing the text. The uniform resource locator was metameric into multiple tokens from that classification options were extracted. The options shapely sequential dependencies between tokens. The authors pointed to the actual fact that the

mixture of high-quality uniform resource locator segmentation and have extraction improved the classification rate over many baseline techniques. Baykan et. al. [6] pursue a similar objective: topic classification from URLs. They trained separate binary classifiers for every topic (student, faculty, course and project) and were able to improve over the simplest rumored F-measure. Ma et. al. [7] brought forth associate degree approach to find malicious websites from the lexical and host-based options of their URLs in light-weight of the wealth of knowledge they carry concerning the website's nature. Their system is ready to sift through tens of thousands of options and determine the vital uniform resource locator components and information while not requiring significant domain expertise. The approach was evaluated with up to thirty,000 instances and yielded promising results, particularly a really high classification rate (95%-99%) and a coffee false positive rate. The same authors in [8] resorted to on-line algorithms so as to handle innumerable URLs whose options evolve over time. A system was developed to assemble period uniform resource locator options. The authors paired it with a period feed of labelled URLs from an oversized internet mail supplier. A classification rate of ninety nine is rumored on a balanced dataset victimization confidence-weighted learning. Zhao and Hoi [9] avoid the (typical) category imbalance within the malicious uniform resource locator detection downside and also the would like for an oversized amount of coaching information through their Cost-Sensitive on-line Active Learning (CSOAL) framework. CSOAL queries solely a small fraction of the on the market information (about zero.5% out of one million instances) and directly optimizes 2 cost-sensitive measures to handle the class-imbalance issue. The empirical proof indicated that their theme achieved higher or highly comparable classification performance when put next to the progressive cost-insensitive and cost-sensitive on-line classification algorithms employing a Brobdingnagian quantity of labelled information.

II. RELATED WORKS

Drive-by-download attack [3] [5] is, by which, malicious websites inject malware onto users computers once users visit these websites. Provos et.al. [15] known four major types of the attack: advertising, third-party gadget, web security application, and user contributed content. In drive-by download attack, once a user browses the landing web site, he will be directed to a drive-by-download server, sometimes known as hop point. The hop purpose can determine the vulnerabilities of the user system and choose the weakest one to launch Associate in Nursing attack.

The attack can command the browser to transfer malware from the malware distribution web site. Finally, the malware is put in and dead mechanically while not user noticing it. Because the attack grows apace, any effective thanks to stop the attack is to develop detection mechanism, before it's activated. The best approach is to spot and refrain from connecting to malicious sites.

Blacklisting could be a standard and wide used technology. Google blacklists roughly 9500 to ten thousand websites per day [11]. However, although blacklisting prevents variant malicious attacks, it's not effective to safeguard once the assaultive websites area unit unknown. Crawler primarily based looking and detection malicious websites throught the entire net is not possible. Sandboxing (testing on a unique platform that may not have an effect on the main system) is a good thanks to notice a malicious website [21]. nonetheless it takes a minimum of tens of seconds to verify a single web site. Blacklisting will be combined with different technologies for higher security. we have a tendency to propose to use machine learning to classify malicious websites in period of time before progressing to access an unknown web site [11].

Ma et al. [8] used four datasets and valid the chance of distinguishing malicious websites by victimization 3 machine learning models: Naive Bayes, Support Vector Machine with an RBF kernel Associate in Nursingd regularized supply regression. Kazemian and Ahmed [6] compared many machine learning models including 3 supervised classifiers: K-Nearest Neighbor, SVM, and Naive Bayes; and 3 unsupervised techniques: Mini Batch K-Means, Affinity Propagation and K-Means. Supervised techniques might reach a classification accuracy of 85-97%. Darling et. al. [12] developed a classification systems supported lexical analysis. They collected their datasets by configuring their crawler to gather from six sources and used eighty seven options for his or her call tree primarily based system. However, the main disadvantage was the very fact that they used huge number of options to attain their results, creating the method slow to coach the choice tree. Finally, we are going to compare our results thereupon obtained by Darling.

III. DATA COLLECTION AND FEATURE

URL Dataset

This is an important topic and one of the most difficult thing to process, according to other articles and another open resource, we used three black list: + machinelearning.inginf.units.it/data-andtools/hidden-fraudulent-urls-dataset + malwaredomainlist.com + zeuztacker.abuse.ch. From them we got around 185181 URLs, we supposed that they were malicious according to their

information, we recommend in a next research step to verity them though another security tool, such as, VirusTotal.

We got the benign URLs (345000) from https://github.com/faizann24/Using-machinelearning-to-detect-malicious-URLs.git, similar to the previous step, a verification process is also recommended through other security systems.

1. For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

2. For papers with less than six authors: To change the default, adjust the template as follows.

    a. Selection: Highlight all author and affiliation lines.

    b. Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

    c. Deletion: Delete the author and affiliation lines for the extra authors.

Feature Generator

During the research process we found that one way to study a malicious website was the analysis of features from its application layer and network layer, in order to get them, the idea is to apply the dynamic and static analysis.

In the dynamic analysis some articles used web application honeypots kind high interaction, but these resources have not been updated in the last months, so maybe some important vulnerabilities were not mapped.

Data Description

Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

Table 1 Data Description

URL	It is the anonimous identification of the URL analyzed in the study
URL_LENGTH	it is the number of characters in the URL
NUMBER_SPECIAL_CHARACTERS	it is number of special characters identified in the URL, such as, “/”, “%”, “#”, “&”, “. “, “=”
CHARSET	it is a categorical value and its meaning is the character encoding standard (also called character set)
SERVER	it is a categorical value and its meaning is the operative system of the server got from the packet response.
CONTENT_LENGTH	it represents the content size of the HTTP header
WHOIS_COUNTRY	it is a categorical variable, its values are the countries we got from the server response (specifically, our script used the API of Whois).
WHOIS_STATEPRO	it is a categorical variable, its values are the states we got from the server response (specifically, our script used the API of Whois).
WHOIS_REGDATE	Whois provides the server registration date, so, this variable has date values with format DD/MM/YYY HH:MM
WHOIS_UPDATED_DATE	Through the Whois we got the last update date from the server analyzed
TCP_CONVERSATION_EXCHANGE	This variable is the number of TCP packets exchanged between the server and our honeypot client
DIST_REMOTE_TCP_PORT	it is the number of the ports detected and different to TCP
REMOTE_IPS	his variable has the total number of IPs connected to the honeypot
APP_BYTES	this is the number of bytes transfered
SOURCE_APP_PACKETS	packets sent from the honeypot to the server
REMOTE_APP_PACKETS	packets received from the server
APP_PACKETS	this is the total number of IP packets generated during the communication between the honeypot and the server
DNS_QUERY	this is the number of DNS

_TIMES	packets generated during the communication between the honeypot and the server
TYPE	this is a categorical variable, its values represent the type of web page analyzed, specifically, 1 is for malicious websites and 0 is for benign websites

Sample of a Table footnote. (Table footnote)

Example of a figure caption. (figure caption)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

IV. CLASSIFICATION TECHNIQUES

Random Forest (RF) [13] may be a well-known ensemble learning technique for supervised classification or regression. This machine learning technique operates by building associate ensemble of random call trees at coaching time and outputting the class that's the mode of the categories (classification) or mean prediction (regression) of the individual trees. so a RF may be a classifier consisting in a very assortment of tree structured classifiers that uses random choice in 2 moments. In a first step, the rule selects many (e.g. 500) bootstrap samples from the historical knowledge. for every bootstrap choice *k*, the dimensions of the chosen knowledge is roughly 2/3rd of the full training knowledge (exactly sixty three.2%). Cases area unit designated willy-nilly with replacement from the initial knowledge and observations in the original knowledge set that don't occur in a very bootstrap sample are known as out-of-bag (OOB) observation. in a very second step, a classification tree is trained victimization every bootstrap sample, but solely a little range of willy-nilly designated variables (commonly the root of the quantity of variables) area unit used for partitioning the tree. The OOB error rate is computed for each tree, victimization the remainder (36.8%) of historical knowledge.

```
# Train a Random Forest Regressor
from sklearn.ensemble import RandomForestClassifier

# n_estimators is the number of random forests to use
# n_jobs says to use all processors available
rf = RandomForestClassifier(n_estimators=100, n_jobs=-1, max_depth=30, criterion = 'entropy')
rf.fit(X_train, y_train)

print('Training Accuracy Score: {}'.format(rf.score(X_train, y_train)))
```

```
Training Accuracy Score: 1.0
```

Figure 1 Random Forest Implementation using Python

The overall OOB error rate is then mass, observe that RF does not need a split sampling technique to assess accuracy of the model. the ultimate output of the model is that the mode (or mean) of the predictions from every individual tree. Random Forest comes at the expense of a some loss of interpretability, but generally greatly boosts the performance of the ultimate model, becoming one in every of the foremost seemingly to be the most effective acting classifier in real-world classification issues [11] [12].

V. RESULT

First, there's a lot of SERVER types. To reduce our dimensionality, we'll make that assumption that having a rare server type is more interesting then the specific server type for a decision tree classification. It's possible we could extract interesting features of a server type (such as old versions, the "base" type such as nginx, the specific version type, etc.) which would be an interesting way to extend the research.

We've reduced our number of unique values in our column sets quite a bit.

SERVER (web server types) are now 98 vs. the original 240

WHOIS_REGDATE is now down to 30 vs. 891
WHOIS_UPDATED_DATE is now down to 10 vs. 594

```
Test results:

Accuracy Score: 0.9551

Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97       462
           1       0.98      0.68      0.81        73

   micro avg       0.96      0.96      0.96       535
   macro avg       0.97      0.84      0.89       535
weighted avg       0.96      0.96      0.95       535


Confusion Matrix:
[[461   1]
 [ 23  50]]
```

Figure 2 Final Result

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks". Machine Learning No. 20, pp.273-297 (1995).

[2] C. M. Chen, J. J. Huang, Y. H. Ou, "Efficient suspicious URL filtering based on reputation". Journal of Information Security and Applications Vol. 20, pp.26-36 (2015)

[3] P. de las Cuevas, Z. Chelly, A. Mora, J. Merelo, and A. EsparciaAlcazar, "An improved decision system for URL accesses based on a ´ rough feature selection technique," in Recent Advances in Computational Intelligence in Defense and Security. Springer, 2016, pp. 139–167.

[4] A. Mora, P. De las Cuevas, and J. Merelo, "Going a step beyond the black and white lists for URL accesses in the enterprise by means of categorical classifiers," ECTA, pp. 125–134, 2014.

[5] M.-Y. Kan and H. O. N. Thi, "Fast webpage classification using url features," in Proceedings of the 14th ACM international conference on Information and knowledge management. ACM, 2005, pp. 325–326.

[6] E. Baykan, M. Henzinger, L. Marian, and I. Weber, "Purely URL-based topic classification," in Proceedings of the 18th international conference on World wide web. ACM, 2009, pp. 1109–1110.

[7] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009, pp. 1245–1254.

[8] "Learning to detect malicious URLs," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, p. 30, 2011.

[9] P. Zhao and S. C. Hoi, "Cost-sensitive online active learning with application to malicious URL detection," in Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013, pp. 919–927.

[10] M. Aldwairi and R. Alsalman, "MALURLS: A Lightweight Malicious Website Classification Based on URL Features", Journal of Emerging Technologies in Web Intelligence (JETWI) Vol. 4, pp.128-133 (2012)

[11] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do ´ we Need Hundreds of Classifiers to Solve Real World Classification Problems?" Journal of Machine Learning Research, vol. 15, pp. 3133–3181, 2014.

[12] M. Wainberg, B. Alipanahi, and B. J. Frey, "Are random forests truly the best classifiers?" Journal of Machine Learning Research, vol. 17, no. 110, pp. 1–5, 2016.

[13] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[14] Zhou, Q., Song, et.al., Efficient Lasso training from a geometrical perspective, Neurocomputing, Vol. 168, pp. 234-239 (2015).

[15] Ourston D, Matzner S, Stump W, Hopkins B., "Application of hidden Markov models to detecting multi-stage network attacks," In System Sciences, Proceedings of the 36th. Annual Hawaii Intl. Conf., 2003.

[16] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 12, pp. 1553–1566, 2004.

[17] Symantec, "2016 Internet security threat report," https://www.symantec. com/security-center/threat-report, 2016, [Online; accessed 11-Aug2016]

[18] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: predictive blacklisting to detect phishing attacks," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–5.

[19] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in Proceedings of the 23rd USENIX Security Symposium, 2014, pp. 625–640.

[20] R. Tibshirani, Regression Shrinkage and Selection via the lasso, Journal of the Royal Statistical Society Vol.58, pp.267-288 (1996).

[21] P. Zhao and S. C. Hoi, "Cost-sensitive online active learning with application to malicious URL detection," in Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013, pp. 919–927.