Journal of Information Technology and Digital World (ISSN: 2582-418X)
www.irojournals.com/itdw/

Inhibiting Webshell Attacks by Random Forest Ensembles with XGBoost

D. Sasikala[1*], D. Chandrakanth[2], C. Sai Pranathi Reddy[3], J. Jitendra Teja[4]

[1]Professor, Department of Computer Science & Engineering, JB Institute of Engineering & Technology, Moinabad, Hyderabad
[2,3,4]UG Scholars, Department of Computer Science & Engineering, JB Institute of Engineering & Technology, Moinabad, Hyderabad

E-mail: [1]godnnature@gmail.com

Abstract

Malign websites effectively endorse the evolution of web illicit events and force the progression of Web services. As an efficient outcome, there is powerful enthusiasm to create systemic resolutions in inhibiting the client from the call onto such Websites. Knowledge-centered Random Forest outfits with XGBoost tactic is recommended for categorizing Websites into 3 categories: Benign, Spam and Malicious. This practice evaluates the Uniform Resource Locator in the situation deprived of accessing the matter of Websites. Thus, it wipes out the run-time expectation and the likelihood of uncovering clients to the browser aimed susceptibilities. As a consequence of involving Random Forest Ensembles with XGBoost, it realizes superior enactment on expansive view and publicity correlated with blacklisting amenity. Preprocessing is performed in order to improve the quality of the data subsequently, analyze certain algorithms, thereby explore the best model are the facts discussed in this research. Work also continues to probe how well this chosen archetypal will operate in the future ahead.

Keywords: Webshell attacks, benign, spam and malicious websites, uniform resource locator, browser-based vulnerabilities

1. Introduction

Web shells are malevolent scripts that facilitate threat actors to give and take web servers and lift-off further attacks. These treat actors initially pierce into a network or system and then set up a web shell [1-3]. From this instant onwards, they custom them as an

DOI: https://doi.org/10.36548/jitdw.2022.3.003
Received: 30.06.2022, received in revised form: 24.07.2022, accepted: 10.08.2022, published: 20.08.2022

everlasting backdoor into the aimed web purposes and any allied schemata. Web shell attackers utilize them for the subsequent purposes:

1. Exfiltrating and ingathering delicate facts and credentials.
2. Transmitting malevolent that will hypothetically build a gushing hovel for promoting contaminating and perusing of other losses.
3. Vandalizing websites by transforming or accumulating files and many more.

In the course of a web shell attack, a cybercriminal inoculates a malevolent file into an objective web server's directory and then implements malevolent activities from their web browser for that file.

Just the Websites have acquired distinctive cosiness to ample individuals for governing their reserves and assets. It is likewise made available with openings for guiding scam on an enormous scale using trivial price to the cheaters [4, 5]. Swindlers influence customers as a standby of hardware/software, where hurdles to high-tech conciliation amplify much. Phishing is a unique, utmost extensively proficient web scams. It is highlighted on the holdup of delicate individual facts, including credit card and passwords details. Phishing spasms yield two variations:

- Determinations to delude preys to source them to uncover their ambiguities by playacting to be reliable individuals by a tangible requisite for such facts.
- Efforts to achieve enigmas by instilling malware on top of preys' kit.

The precise malware castoff in phishing outbreaks is the topic of a probe by the virus and malicious software group that is not studied in this proposal. Phishing outbreaks that ensue by unreliable customers are the study emphasis of this work and the word 'phishing' will be castoff to denote this category of the outbreak. The focal intent of this work is to spot the Benign, and Malicious web shell attacks with the use of ML (machine learning). The motive of this system is to yield securities to inhibit customers from these detrimental sites [6, 7]. It will mark individuals cognizant besides constructing robust security means that are capable to spot and inhibit phishing web shell from realization of the customer.

2. Literature Survey

Analysis of [9], well-defined the design & enactment aspects of a scalable ML classifier - online gradient descent Logistic Regression progressive to spot phishing sites utilized to sustain Google's phishing blacklist naturally. Once training ends in many weeks,

yet with noises, it perfectly categorized beyond 90% of phishing pages. Review of [10], explored SMOTE Boost plus, a unique modest, rapid, efficient and excellent process RUS Boost for easing the inspiring skewed training data, progressing its class imbalance data that is a smaller amount of cost ensuing in appreciably smaller archetypal training times. Valuation of investigation [11] appraised enactment of an extensive set of pattern recognition & ML processes such as, Multilayer perceptron, K-means clustering, Gaussian classifier, Incremental radial basis function, Nearest cluster algorithm, Leader algorithm, Fuzzy ARTMAP, Hypersphere algorithm, C4.5 decision tree on four outbreak classes comprising of Probe, DoS, U2R (User to Root attack), R2L (Root to Local attacks) as set up in the KDD 1999 Cup ID (intrusion detection) dataset. Consequences of simulation study realized to that outcome specified that guaranteed categorization processes, implement better for assured outbreaks. Exploration is that definite processes execute better for certain outbreak classes and so, if a multi-expert classifier design furnish the preferred enactment measure is of great concern.

Inspection of research work [12] conferred that spammers entail bloomed image spam, and emails that encircle the text of the spam in a human understandable imagery in its place of the spam body, creating recognition by custom content filters are hard. Innovative systems are vital to filter these spams. The objective was to inevitably categorize an image openly as being spam or ham. The aspects that accent on modest possessions of the image are given constructing categorisation as swiftly as viable. This appraisal precisely categorized spam images in surplus of 90% and up to 99% on actual data. Still, a fresh facet selection process was hosted that prefers facets for categorization centred on their rapidity besides forecasting power. This system created a precise scheme that tracks in a tiny portion of the time. To conclude, Justin Time (JIT) facet extraction that crafts facets at a categorization time as vital by the classifier that is hosted. JIT mining is customary by JIT choice that promotes escalation scheme swiftness.

The script created hands-on imagery spam categorization by imparting together great accuracy aspects and a way to study the high-speed classifiers. Appraisal of this work [13] narrated on the similarities and dissimilarities of the verb-subject and verb-object in their custom amid legitimate and phishing emails. The fortitude of this study was to realize if the syntactic structures, subjects and objects of verbs will be unique facets for phishing recognition. For realizing this intent, two chains of research, the syntactic match for sentences, the verb-subject and verb-object appraisal were steered specifying the

significances that is together with the facets that will be castoff for various verbs, then further work has to be completed for others.

The study of work [14] etiquette spotted phishing on email stage reasonably than in impersonated websites that is vital to avert the prey from selecting any damaging linkages in the email. The execution named Phish Net-NLP, functions amid of a customer's mail user agent (MUA), mail transfer agent (MTA) and scheme search inward email for phishing outbreaks prior to receiving these in the inbox. Probing [15] conversed on the user-friendly automated anti-phishing tools that deteriorate due to false positives, false negatives, and numerous real-world sprints. Thereby iTrust Page – anti-phishing tool trusts on customer input and external warehouses of facts to avert clients from filling out phishing Web forms aids in resolving if or not it is legitimate. As iTrust Page is customer-supported, escapes the false positives and the false negatives correlated to automatic phishing recognition. iTrust page was realized as a downloadable extension to FireFox, then Mozilla website. Centered on the study of this tool's effectiveness and ease of custom-built inspection of usage logs compiled from the 2,050 iTrust Page customers for more than two weeks it was found that iTrust page disorders customers on less than 2% of the pages they pop in, and the extent of interferences drops over time.

A survey of research works [8] and [9] was on [16] and [17] that surveyed the clustered works on the exposing of phishing outbreaks. Examination of [18] familiarizes freshly, the adversarial data mining i.e., categorization issue is experienced as a game tool amid adversary and intelligent adaptive classifier. Above few eons, phishing fraud over malevolent emails was a severe menace that has an abundant impact on universal security plus cheap, where out-dated spam filtering scheme was vain. An excellent scheme was recommended to dynamic games of partial facts, a game hypothetical data mining configuration was advocated to create an adversary-aware classifier for phishing scam recognition by an online anticipated Weighted Margin SVMs using a game hypothetical erstwhile knowledge function.

Summary of the research issues and problems of the erstwhile attack, revealing techniques is that, attackers custom an extensive choice of web request susceptibilities and abuses to distribute web shells, comprising cross-site scripting (XSS) and SQL injection (SQLi) along with invincibility in applications and services, wide-open admin edges, file processing susceptibilities, over and above remote file inclusion (RFI) and local file inclusion (LFI) liabilities.

3. System Analysis

3.1 Prevailing Systems

An unreliably regulated NN model will root itself to underfit the guiding dataset. As an option, amplification in the reformation of the scheme to outfit each single constituent in the operational dataset will root the scheme to be overfitted. A probable resolution is to evade the overfitting issue is reshuffling the NN archetypal via tuning precise factors, totalling novel neurons to the concealed level or at times accumulation of an innovative level to the web. A NN by a trivial number of concealed neurons may not involve an apt depictive authority to process the intricacy and multiplicity inborn in the data. Conversely, webs with too much concealed neurons will overfit the data. Yet, at definite step the archetypal will have no elasticity to be amended, then, the constituting practice is to be ended. From now, an ample fault rate is stated when engendering any NN archetypal that it is swotted as an issue as it is hard to define the sufficient fault rate a priori. For illustration, the archetypal designer will fix this satisfactory fault rate to a value explicitly inaccessible that roots the archetypal to glue in local minima or now and then the archetypal designer can fix the adequate fault rate to a value that will as well be enhanced with RF archetypal and Adaboost prototypical and their hybrid exemplary.

3.1.1 Disadvantages

1. This one will procure time to put in the entire dataset.
2. The procedure is not precise.
3. Analysis is gradual.

3.2 Suggested System

Lexical facets are customary on the surveillance that the URLs of several illicit sites act dissimilar, matched by authentic sites. Investigating lexical facets aids to confining the stuff for categorization drives. First, the two portions of a URL are extricated, the host name as well as the path, from the bag-of-words (strings delineated by ‘/’, ‘?’, ‘.’, ‘=’, ‘-’ then ‘’) endure mined. It is located that the phishing site opts to take lengthier URL, extra stages (restricted by dot), additional tokens in a field track lengthier token. Farther, malicious and phishing sites will make believe to be benevolent by comprising prevalent trademark forenames as tokens other than those in the next-stage field. The malicious and phishing sites utilize IP address straight to shield the apprehensive URL, very infrequent in benign instance.

On the contrary, phishing URLs are set up to hold some redolent word tokens (account, banking, confirm, secure, sign in, login, webscr, ebayisapi), the reality of these safety subtle words is ensured with the binary quantity in these facets. Naturally, malevolent locations are eternally not as much prevalent as the benign ones. For this intent, site reputation is deliberated as a vital aspect. Traffic rank facet is attained from Alexa.com. Host-centered facets set upright are customary on the remark that malevolent locations are perpetually recorded in not as much of trustworthy hosting centres or constituencies.

3.2.1 Advantages

1. Intact URLs in these datasets are categorized.
2. Random forest algorithm (supervised learning) is utilized to train using scikit-learn library, and experimentations on support vector machine are in progress.

4. Methodology

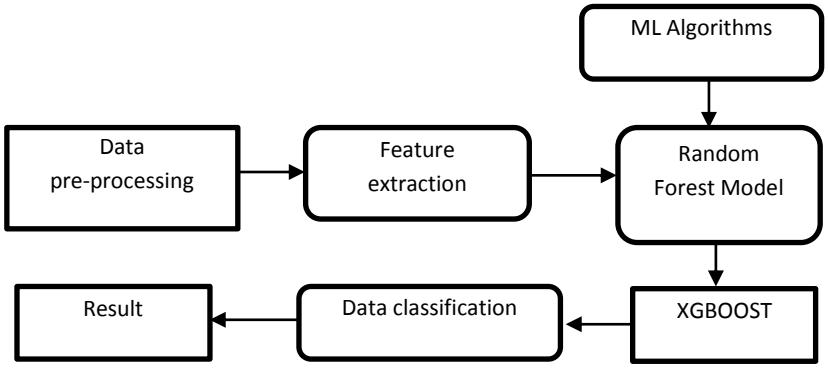

Figure 1. System Architecture

4.1 Modules

- Data pool
- Data pre-handling
- Feature extracting
- Appraisal model

4.1.1 Data Pool

Data castoff in this work is customary annals of PhishTank website. The dataset involves about 30 facets. There are nearly 11,000 illustration websites. This step involves with opting for the subsets of all accessible data that is at work. ML issues begin with data,

and plenty of facts (instances or interpretations) for that is already well-known for the target response termed *labelled data*.

4.1.2 Data Pre-handling

The consolidate data are processed with three steps as cleaning, formatting, and sampling.

Formatting: The data that have been designated may not be in a format that is apt to work with. The facts may be in a relational database and will prefer it in a flat file, or in a proprietary file format & will opt for it now as a relational databank or a text file.

Cleaning: It is the elimination or setting of mislaid/lost data. There may be data illustrations that are incomplete and do not transmit the data needed to be addressed in this issue. These instances may entail to be detached. Besides, at hand will be subtle facts in definite facets and those must be anonymized or detached on or after the data wholly.

Sampling: There may be far more selected data available than needed to work with. More data will considerably cause extensive realizing periods for processes and superior memory and computational necessities. Before taking the whole data, a small piece of data is considered for fastening the exploration and prototyping steps.

4.1.3 Feature Extracting

The next thing is to do Feature extraction and it is an attribute extension that create more columns from URL's including having IP Address, URL Length, shortening service, having @ symbol, etc. To end with, these models are trained by the RF algorithm, XGBOOST algorithm and Random Forest Ensembles with Xgboost using the labelled dataset gathered from PhishTank website. The rest of the labelled data is utilized to evaluate the archetypes.

4.1.4 Appraisal Model

Archetypal estimate is a vital segment of the archetypal expansion practice. It helps to find the best model that are these facts and how well the chosen model will work in the future. To evade overfitting individually, these techniques are done via a test customary (not realized by the archetypal) to appraise prototypical enactment. Enactment of each classification model is assessed centered on its average. The outcome will be in the envisaged

form. Depiction of categorized facts is in the form of graphs. **Accuracy, Recall, Precision and F1 Score** are the metrics for appraisal.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1\ Score = \frac{2\ xPrecision\ x\ Recall}{Precision + Recall}$$

where, TP – True Positive, TN – True Negative, FP – False Positive and FN – False Negative

4.2 TensorFlow

Tensorflow structural enterprise procedures have three segments: Pre-processing the data, Construct the archetypal and Coaching and assessment of the archetypal entitled as Tensorflow as it ensues with its input as a multi-dimensional array above and beyond, well-known as tensors.

It is categorized into *Development Phase*: This is when coaching the method is regularly completed on Desktop or Laptop. *Run Phase or Inference Phase*: After coaching is completed, Tensorflow will be tracked on numerous dissimilar stages with Desktop executing Windows, macOS or Linux, Cloud as web amenity besides Mobile devices similar to iOS and Android.

These will be coached on manifold technologies at that time will realize it on a dissimilar device, formerly has been ready as a proficient archetypal logged on and regulated by further lingos primarily, Python. As a final point, a vital facet of TensorFlow is the TensorBoard that is empowered to observe realistically and visually whatever activities that TensorFlow is undertaking.

Noticeable process reinforced by TensorFlow is the Boosted tree categorization, tf.estimator. Boosted Trees classifier is extended to Boosted Random Forest Categorization that encompasses self-governing decision trees.

4.3 Algorithms

4.3.1 Random Forest

It is a supervised ML process customary on ensemble learning i.e., a sort of learning to seam with dissimilar categories of processes, or identical process multiple spells custom a further influential forecast archetypal. The random forest process pools, multiple process of the identical category i.e., multiple decision *trees*, follow-on as a *forest of trees*, so the term "Random Forest" and that process are castoff for regression tasks too.

a) Active Random Forest

The modest phases expounded in execution of the random forest process are subsequent:

1. Prefer and pick out N arbitrary archives from the given dataset.
2. Construct a decision tree centred on these N annals.
3. Decide on the quantity of trees crucial in this process and recap phases 1 and 2.
4. Intended for categorization issue. Every tree in the forest forecasts the class to which the novel annals fit. To end with, the naive archives are allotted to the class that earns the bulk vote.

b) Benefits of Manipulating Random Forest

1. The process of random forest algorithm is non subjective, as, there are many trees, and every tree is proficient with a subcategory of data. Ultimately, the random forest process depends over the power of "the mass"; thus, the global part of the process is lowered.
2. This process is very steady. Even if an innovative data point is hosted in the dataset the complete process is not affected much, as naive data will influence unique tree, then it is very hostile for it to influence entire trees.
3. The process works fine when numerical and categorical facets are composed.
4. The random forest process too works excellently when data has lost/misplaced values or it does not be extant sound scaled.

4.3.2 XGBoost

XGBoost is a customized and refined algorithm of a Gradient Boosting approach for giving a better energizing performance and computational speed with higher accuracy. The utmost vital issue after the realization of XGBoost is its scalability in the complete scenarios,

and illustrations in distributed or memory-limited settings. The scalability of XGBoost is owed to numerous vital algorithmic optimizations. These novelties embrace a unique tree, realizing process for dealing with sparse data; a hypothetically justified, weighted quantile delineate practice empowers supervision of instance loads in imprecise tree realizing. Parallel and distributed computing makes realizing faster that empowers more rapid model exploration. Further notably, XGBoost achieves out-of-core computation and empowers data scientists to practice hundreds of millions of cases on a desktop. To end with, it is uniformly stimulating to chain these systems to mark an endwise scheme that scales to superior level data using the minimum quantity of cluster assets.

a) Active XGBoost

Distinct to former boosting processes, wherever loads of misclassified outlets are improved, in Gradient Boosted processes the deficit function is enhanced. XGBoost is a cutting-edge feat of gradient boosting besides specific regularization facets. XGBoost will custom the Lasso and Ridge Regression regularization as one to rectify the immensely multifaceted archetypal. Parallelization and Cache block: XGboost cannot coach multiple trees in parallel, since it will engender the diverse nodes of tree in parallel, for that, data prerequisites are to be arranged in rank.

b) Benefits of Deploying XGBoost

1. Highly flexible.
2. Utilizes the power of parallel processing.
3. Faster than Gradient Boosting.
4. Supports regularization.
5. Designed to handle missing facts with its in-build facets.
6. The customer will run cross-validations ensuing towards every iteration.
7. Works well in small to medium dataset

4.3.3 Random Forest Ensembles with XGBoost

The XGBoost library lets the prototypes to be coached in a mode that repurposes and fixes the computational proficiencies realized in the reference library for working out random forest prototypes.

1. XGBoost tracks will be set up to coach random forest ensembles.
2. To custom the XGBoost, API trains and appraises random forest ensemble prototypes for categorization.

3. To tune the hyper-factors of the XGBoost, random forest ensemble archetypal is used.

a) Working of Random Forest Ensembles with XGBoost

XGBoost is routinely castoff to coach gradient-boosted decision trees and additional gradient boosted prototypes. Random Forests custom the identical archetypal depiction and insinuation, as gradient-boosted decision trees, but then is a dissimilar training process. The individual will custom XGBoost to coach a separate random forest or practiced random forest as a source archetypal for gradient boosting. Now the emphasis is on coaching separate random forest. Native APIs for coaching random forests are accessible since the early days, and a new Scikit-Learn wrapper after 0.82 (not incorporated in 0.82). The novel Scikit-Learn wrapper is yet new that signifies to amend the edge every time.

b) Advantages of Using Random Forest Ensembles with XGBoost

1. The core benefit of using the XGBoost library to train random forest ensembles is speed. It is awaited to be much quicker to custom than additional executions, with the inborn scikit-learn execution.
2. Intensely accelerates the training of the model and will be parallelized across clusters.
3. Frequently end in better overall performance of the model.
4. Give more preferences to hyper-factors to optimize the model.
5. Reduce the cost of the model.

5. Datasets and Results

From the public GITHUB repositories, the webshell detection dataset is extracted. PhishTank is an open community spot where someone will defer to, validate, track, and distribute phishing data. It is at liberty to one and all, together, the website and the facts functioned by Cisco Talos Intelligence Group (Talos) (through the API). It’s not fortification and is a data clearinghouse that aids to transfer brightness on a few of the gloomy alleyways of the Internet, also given that precise illegal facts of any individual to ascertain ruthless actors, if it’s for others or for their own (i.e., constructing security tools). There are about 15 emails and if that may be phishing sites, they are cast-off by attackers, they snip our private facts done over this deceitful effort.

Training data of 80 % of an overall dataset along with 20% of testing set investigation was performed in Python with Scikit-Learn, Pandas, Numpy, Keras, Tensorflow packages using both PhishTank and GITHUB datasets. Basically, it consists of attributes which are commonly used in the identification of webshell attack in a web server. The main goal is to find the presence of the webshell by categorizing into Benign, Spam or Malicious. Websites are categorised only by analyses of the URL itself without gaining access to the content of Web sites. Consequently, it excludes the run-time latency and the option of revealing manipulators to the browser centered susceptibilities. By retaining the algorithm, Random Forest Ensembles with Xgboost, realizes better enactment on generality and coverage associated with blacklisting service.

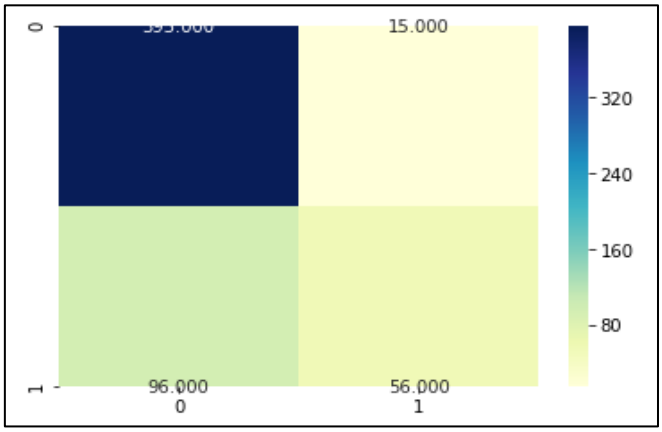

Figure 2. Confusion Matrix

Table 1. Categorization Outcomes

Categorizer	Train Time(s)	Test Time(s)	Accuracy	Recall	Precision	F1 Score
Random Forest	0.513417	0.025628	0.965849	0.972133	0.953571	0.962916
XGBoost	0.577532	0.007841	0.985683	0.977922	0.975195	0.979721
Random Forest Ensembles with XGBoost	0.536218	0.004782	0.989417	0.985736	0.9874431	0.988293

6. Conclusion And Future Scope

In this research work, the large-scale scheme is designated for inevitably categorizing phishing pages that sustains a FPR lower than 0.1%. This categorization process inspects

millions of probable phishing pages day-to-day in a little time of a physical analysis procedure. By spontaneously filling in the blacklist by the classifier, the extent of time and cost are curtailed that phishing pages will endure active afore to protect these customers from them. Moreover, with a flawless classifier and a robust scheme, it is spotted that this blacklist tactic retains always by a pace after the phishers. It is merely ascertained by phishing and customary URLs by RF ensembles with XGBOOST process. The outcome is attained in terms of metrics that comprise accuracy as well. Best accurate results are obtained from this created model, and executing in a very less time. By means of phishing websites escalations, outbreaks too are observed gradually, and specific facets will be encompassed or swapped with novel ones to spot these outbreaks and beware of these.

References

[1] G. Aaron and R. Rasmussen, “Global phishing survey: Trends and domain name use in 2016,” 2016.

[2] Gupta, Brij B., Aakanksha Tewari, Ankit Kumar Jain, and Dharma P. Agrawal. "Fighting against phishing attacks: state of the art and future challenges." *Neural Computing and Applications* 28, no. 12 (2017): 3629-3654.

[3] Aleroud, Ahmed, and Lina Zhou. "Phishing environments, techniques, and countermeasures: A survey." *Computers & Security* 68 (2017): 160-196.

[4] Aaron, G., and R. Rasmussen. "Phishing Activity Trends Report, 4th Quarter 2015." (2016).

[5] Verma, Rakesh, Narasimha Shashidhar, and Nabil Hossain. "Detecting phishing emails the natural language way." In *European Symposium on Research in Computer Security*, pp. 824-841. Springer, Berlin, Heidelberg, 2012.

[6] Khonji, Mahmoud, Youssef Iraqi, and Andrew Jones. "Phishing detection: a literature survey." *IEEE Communications Surveys & Tutorials* 15, no. 4 (2013): 2091-2121.

[7] Park, Gilchan, and Julia M. Taylor. "Using syntactic features for phishing detection." *arXiv preprint arXiv:1506.00037* (2015).

[8] Dazeley, Richard, John L. Yearwood, Byeong H. Kang, and Andrei V. Kelarev. "Consensus clustering and supervised classification for profiling phishing emails in internet commerce security." In *Pacific Rim Knowledge Acquisition Workshop*, pp. 235-246. Springer, Berlin, Heidelberg, 2010.

[9] Whittaker, Colin, Brian Ryner, and Marria Nazif. "Large-scale automatic classification of phishing pages." (2010).

[10] Seiffert, Chris, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. "RUSBoost: Improving classification performance when training data is skewed." In *2008 19th international conference on pattern recognition*, pp. 1-4. IEEE, 2008.

[11] Sabhnani, Maheshkumar, and Gürsel Serpen. "Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context." In *MLMTA*, pp. 209-215. 2003.

[12] Dredze, Mark, Reuven Gevaryahu, and Ari Elias-Bachrach. "Learning fast classifiers for image spam." In *CEAS*, pp. 2007-487. 2007.

[13] Park, Gilchan, and Julia M. Taylor. "Using syntactic features for phishing detection." *arXiv preprint arXiv:1506.00037* (2015).

[14] Verma, Rakesh, Narasimha Shashidhar, and Nabil Hossain. "Detecting phishing emails the natural language way." In *European Symposium on Research in Computer Security*, pp. 824-841. Springer, Berlin, Heidelberg, 2012.

[15] Ronda, Troy, Stefan Saroiu, and Alec Wolman. "Itrustpage: a user-assisted anti-phishing tool." *ACM SIGOPS Operating Systems Review* 42, no. 4 (2008): 261-272.

[16] Ahmed Aleroud, and Lina Zhou, "Phishing Environments, Techniques, and Countermeasures: A Survey", Computers & Security, Vol. 68, No.9, pp. 1-44, April 2017.

[17] Khonji, Mahmoud, Youssef Iraqi, and Andrew Jones. "Phishing detection: a literature survey." *IEEE Communications Surveys & Tutorials* 15, no. 4 (2013): 2091-2121.

[18] L'Huillier, Gaston, Richard Weber, and Nicolas Figueroa. "Online phishing classification using adversarial data mining and signaling games." In *Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics*, pp. 33-42. 2009.