International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1S5, December 2019

Tuning the False Positive Rate / False Negative Rate with Phishing Detection Models

Sailee Dalvi, Gilad Gressel, Krishnashree Achuthan

Abstract*—Phishing attacks have risen by 209% in the last 10 years according to the Anti Phishing Working Group (APWG) statistics [19]. Machine learning is commonly used to detect phishing attacks. Researchers have traditionally judged phishing detection models with either accuracy or F1-scores, however in this paper we argue that a single metric alone will never correlate to a successful deployment of machine learning phishing detection model. This is because every machine learning model will have an inherent trade-off between it's False Positive Rate (FPR) and False Negative Rate (FNR). Tuning the trade-off is important since a higher or lower FPR/FNR will impact the user acceptance rate of any deployment of a phishing detection model. When models have high FPR, they tend to block users from accessing legitimate webpages, whereas a model with a high FNR will allow the users to inadvertently access phishing webpages. Either one of these extremes may cause a user base to either complain (due to blocked pages) or fall victim to phishing attacks. Depending on the security needs of a deployment (secure vs relaxed setting) phishing detection models should be tuned accordingly. In this paper, we demonstrate two effective techniques to tune the trade-off between FPR and FNR: varying the class distribution of the training data and adjusting the probabilistic prediction threshold. We demonstrate both techniques using a data set of 50,000 phishing and 50,000 legitimate sites to perform all experiments using three common machine learning algorithms for example, Random Forest, Logistic Regression, and Neural Networks. Using our techniques we are able to regulate a model's FPR/FNR. We observed that among the three algorithms we used, Neural Networks performed best; resulting in an higher F1-score of 0.98 with corresponding FPR/FNR values of 0.0003 and 0.0198 respectively.

***Index Terms*—Machine Learning, Phishing Detection, Model Tuning, Cyber-security**

I. INTRODUCTION

Phishing attacks have risen 209% in the last 10 years [19]. Phishing is a masquerading cybercrime in which victims are lured into visiting malicious websites, which appear to be legitimate, in order to steal the victims credentials [23].

Manuscript published on 30 December 2019.
***** Correspondence Author (s)
Sailee Dalvi, Department of Cybersecurity, Systems and Networks, Amrita Vishwa Vidyapeetham, Amritapuri, India. saileenarayandalvi@gmail.com
Gilad Gressel, Georgia Institute of Technology, Atlanta, USA, ggressel3@gatech.edu
Dr. Krishnashree Achuthan, Department of Cybersecurity Systems and Networks Amrita Vishwa Vidyapeetham Amritapuri, India. krishna@am.amrita.edu

[20]. Perhaps the most common solution to prevent phishing attacks is to train humans to detect the attacks. However, it is difficult even for a well trained human to differentiate between a phishing page and legitimate page, especially when they are in a hurry to perform a task [17]. Thus, Machine learning is well suited to solve the task of phishing detection because it is automated, re-trainable (as trends change), and easy to deploy.

Previous work has shown that machine learning models are able to distinguish whether or not a site (or URL) is a phish with greater than 95% accuracy [2] [10] [14] [16].

When implementing any detection model it is very important to examine the False Positive Rates (FPR) and False Negative Rates (FNR) of that model. With any given model there is an inherent trade-off between the FPR and FNR, this is due to the nature of binary classification. A model will find a decision boundary in order to classify any given data point as positive or negative. An increase in classification of the positive class leads to a higher FPR and lower FNR, the reverse is also true [3].

As we move the decision boundary, it can be seen that an increase in FPR will cause a decrease in FNR and vice-versa. In the case of phishing detection a higher FPR will classify more benign pages as phish while a high FNR will classify more phish as benign.

In this paper we argue that, in order to deploy a phishing detection model successfully we must consider the risk assessment of the user-base. Once we understand the risk of phishing for the deployment site, we should then adjust the FPR and FNR rates accordingly. If a model has a high FPR, users will find legitimate pages classified as phish, leading to frustration. We call this type of user frustration false positive frustration, which will lead to users seeking workarounds to circumvent the phishing detection model. In this situation we argue that the phishing detection model should be deployed with a low FPR, in order to avoid the false positive frustration, thus maintaining confidence in the detection model. On the contrary, a business with a high risk user-base (in perhaps a more secure setting) would require emphasis on low FNR. A low FNR represents increased precision in the detection of benign pages, it will naturally increase the amount of false positives (benign pages detected as phish). If there is a high risk impact of a successful phishing attack which would compromise company secrets, then we argue that the model should maintain a low FNR in order to avoid negative business impact due to phishing. There will be some associated false positive frustration but this is acceptable and tolerated in a secure setting, because users will be trained and aware of the security risks.

Phishing detection using machine learning has been studied extensively in the last 10 years [6]. However, all research for phishing detection have used a single metric, either the F1-score or Accuracy to judge the model's performance [1] [2] [14] [17] [18]. Using a single metric is an efficient method to select a generally high performing model, but in order to gain the maximum performance from a given phishing detection model we suggest that all models should have FPR/FNR trade-off tuned to the particular risk assessment of the deployment site. In this work we present two effective methods for tuning a model's FPR/FNR trade-off. We experimented with (a) the class distribution in the training set and (b) the actual decision boundary of a model. There has always been an argument about what class distribution of Phishing:Legitimate should exist within a training set [14]. Further we adjust the decision boundary of three common models in order to empirically test our hypothesis. Our aim was to discover how much we can affect the FPR/FNR trade-off without sacrificing the overall F1-score of a trained model. The result is that we could maintain a F1-score of 0.98 after tuning the FPR/FNR to 0.0003 and 0.0198 respectively. In this paper we see the following as our contribution, As to best of our knowledge we are the first work focused solely on tuning the FPR/FNR of a phishing detection model We performed 27 experiments with different class distri-butions of the training data in order to explore FPR/FNR.
We performed 27 experiments with different decision boundaries in order to explore the FPR/FNR.

II. RELATED WORK

Phishing web pages cannot be easily recognized due to their similar appearance, similar URLs, or source codes. Some classification techniques depend on URL lexical attributes [12] [14], and website page facilitating related highlights [7], to classify the originality of a webpage. Phishing discovery is dependent on visual comparability, assuming that a potentially targeted website is known earlier [7] [12] [14].

To increase the performance of phishing detection techniques, a few guidelines put forth were: dataset selection, unbiased comparison, system designs, temporal resilience. An unbiased real world data set is necessary for effective phishing detection and currently most data sets publicly available are highly biased towards popular English websites [1].

The frequently used features extracted from the URL and source code used to classify pages as phishing and legitimate. Their system used features such as: URL features, consistency in term usage, starting and main level domain (mld) use, Registered domain name (RDN) usage, webpage content, etc. which they described while expanding the feature set considerably [2]. In our work, we extracted 31 features to create a model. The Phishers do not have control over the hyperlinks present on a webpage which get redirected outside the phishing webpages. The Phishers can change most parts of the webpage to make a look-a-like of the target page, but they have limited access to domain names [4]. Hence in our project we focused
on the domain name features because the Phishers try to mimic the domain name to fool the users, for eg. facebook will be written as faceb00k and so on.

One solution for detecting phishing webpages was attempted by maintaining blacklists. Blacklists were enclosed in toolbars which would give feedback to the user. Most of the work has only used URL based features, which were taken from phishtank, majestic million and Alexa. The URL based phishing detection methods generated a lower accuracy with false alarms [13] [2] [14]. We have taken into consideration both URL as well as source code and a few other attributes of a webpage for our project. The well known phishing detection approaches can effi-ciently identify phishing webpages with 99% accuracy, while producing low false positive rates up to 0.1% [7] [2] [10] [14] Our model further reduces the wrongly classified rate by 0.0982% while maintaining a F1-score of 0.98.

III. APPROACH

A. Dataset

The dataset was created at Amrita Vishwa Vidyapeetham Centre for Cybersecurity Systems and Networks. The dataset was collected on daily basis using a web scraping tool (Selenium) [20] and the Google Chrome Webdriver [24]. The URLs for scraping phishing web pages were taken from phish-tank.com and URLs for scraping the legitimate web pages were taken from majestic million [21] [22]. The collected data is as shown in Table I

B. Features

The raw data shown in Table I was used to extract features, it comprised of Date and Time stamp, URL, title, source code, redirection chain, request history, header information, certificate information, screen shot of the webpage, etc. as they are commonly used features in phishing detection study. [2]. Refer Table II for list of extracted features.

C. Algorithms

We used three algorithms to build our classifiers to arrive at a conclusion of which algorithm resulted in most efficient FPR/FNR trade-off. The Algorithms used were:

Random Forest Classifier
Logistic Regression Classifier
Neural Network Classifier

Confusion Matrix

Binary classification deals with two classes, our models are also binary classification as we used two classes, 1 for Phishing and 0 for Legitimate. Table III shows that there are four possible outputs which represent the elements of a 2x2 confusion matrix.

True Positive: If a positive value is correctly classified, it is considered to be a true positive value (TP).

True Negative: If the negative value is correctly classified, it is considered to be a true negative value (TN).

TABLE I
EXTRACTED DATA INFORMATION

Data	Examples
Date-time stamp	24-04-2019 23:29:55
URL	https://theeternalgroup.com/irs/identity.php
Title	Contact Support
Source-code	<!DOC.. html... ”-//W3C//DTD HTML 4.01 Transitional//EN”> <html xmlns=”http://www.w3.org/xhtml”> <head> <title>Contact.....</title..> <meta=”Content-Type” content=”text/html; charset=utf-8” /> </head> <bod-marginw=”0” marginh=”0” leftmargin=”0” top-margin=”0”> <iframe width=”100%” height=”90%” frameborder=”0” scrolling=”auto” marginw=”0” src=”http://fwdssp.com/?dn= referer detect& [redacted] pid=5POL4F2O4”> </iframe> </body> </html>
Redirection chains	[’https://theeternalgroup.com/irs/identity.php’, ’https://theeternalgroup.com/cgi-sys/suspendedpage.cgi’]
Request history	[<Response [302]>]
Header info	f’Type’: ’html’, ’Encoding’: ’chunk’, ’Server info’: ’nginx/1.14.1’, ’Content-Encoding’: ’gzip’, ’Date’: ’Wed, 24 Apr 2019 18:01:24 GMT’, ’Connection’: ’keep-alive’g
Certificate Information	f ”OCSP”: [”http://ocsp.int-x3.letsencrypt.org”], ”caIssuers”: [”http://cert.int-x3.letsencrypt.org/”], ”issuer”: [[[”countryName”, ”US”]], [[”organizationName”, ”Let’s Encrypt”]], [[”commonName”,”Authority...”]]], ”notAfter”: ”Jul 7 21:37:23 2019 GMT”, ”notBefore”: ”Apr 8 21:37:23 2019 GMT”, ”serialNumber”: ”03B44C1CF121F7D36950341 C04C20B5B60D8”, ”subject”: [[[”commonName”, ”theeternalgroup.com”]]], ”subjectAltName”: [[”DNS”,”cpanel.theeternalgroup.com”], [”DNS”,”mail.theeternalgroup.com”], [”DNS”,”theeternalgroup.com”], [”DNS”,”webmail.theeternalgroup.com”], [”DNS”,”webdisk.theeternalgroup.com”], [”DNS”,”www.theeternalgroup.com”]], ”version”: 3 g
Screen captures	2019_04_24_23_29_51.png

TABLE II
FEATURES

Data	Extracted Features
URL	URL length Vowel count Consonant count Vowels/Consonant Ratio Digits count Digits to Letter ratio Special symbols count Dots count HTTPs count Alphabets count Subdomains count Domain length Count of - symbols Non-alphabetic character count Count of = symbols Count of $ symbols Count of ’?s
Source Code	Title present or absent Length of title Count of javascript tags Count of Forms Count of CSS tags Count of href links Count of iframes tag Count of image tags Count of src tags Length of text
Redirection Chain	Count of redirection
Request History	Count of 301 redirection Count of 302 redirection
Certificate Information	Present or Absent

TABLE III
CONFUSION MATRIX

		Predicted Values	
Actual Values		Legitimate (0)	Phishing (1)
	Legitimate (0)	TN (True Negative)	FP (False Positive)
	Phishing (1)	FN (False Negative)	TP (True Positive)

False Positive: If a positive value is incorrectly classified, it considered as a false positive (FP).
False Negative: If a negative value is incorrectly classi-fied, it is considered to be a false negative value(FN).
False Positive Rate (FPR): The ratio between the incor-rectly classified negative samples to the total number of negative samples. Formula: FPR = FP/(TN+FP)
False Negative Rate (FNR): The ratio of positive sam-ples that were incorrectly classified. Formula: FNR = FN/(TP+FN)

E. Training Process

To create the training data from our raw data, we extracted the features as shown in Table II. We then split the data into the ratio of 80 % for training and 20% for testing [2]. In order for the model to learn we gave more training samples and tested it on a lower number of testing samples. In experiment one, we ran three different machine learning algorithms and nine different distributions i.e, 1:1, 1:2, 1:3, 1:5, 1:10, 2:1, 3:1, 5:1,
10:1, Phishing:Legitimate and measured various performance matrix such as Precision, Recall, F1-score, Accuracy, TPR, FPR, TNR and FNR. In experiment two, we choose the best resultant distribution and algorithm obtained in experiment one, i.e. 5:1, Phishing:Legitimate with Neural Networks. We varied the decision threshold value which is set to 0.5 by default for any model, to move either towards 0.1 or towards 0.9. We limited the use of the performance metrics to the F1-score, FPR and FNR for our work.

F. Testing Process

The ratio of Phishing:Legitimate in testing is a represen-tation of the real world scenarios which is approximately 1:100, Phishing:Legitimate [2]. For our testing set, we fixed

1:10, Phishing:Legitimate distribution. To test the model's performance, total 54 tests were run, 27 tests where the training distribution was varied and 27 tests where the decision threshold was varied, with the distribution of testing dataset being constant.

IV. RESULTS AND DISCUSSION

A. Experiment 1

1) Random Forest - In this experiment we observed that the change in the distribution affects the FPR/FNR; for Random Forest distributions such as: 1:2, 2:1, 3:1, 5:1 which obtained better results as compared to other distributions. As illustrated in Fig 2. we obtained a FPR/FNR score of 0.0006 and 0.0682 while maintaining the F1-score of 0.9614 in Fig 1.

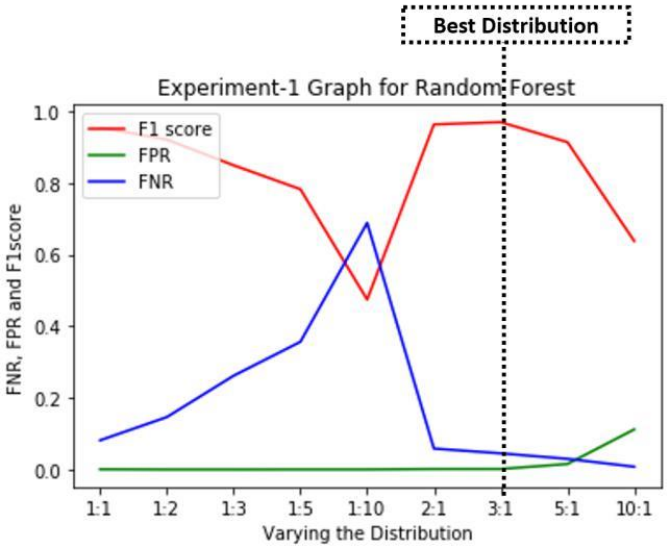

Fig. 1. Random Forest- F1-score, FPR, FNR

2) Logistic Regression - In this experiment we observed that the change in the distribution affects the FPR/FNR; for Random Forest distributions such as: 3:1, 5:1 which better results as compared to other distributions. As seen in Fig 4. we obtained a FPR/FNR score of 0.0078 and 0.0451 while maintaining the F1-score of 0.9393 in Fig 3.

3) Neural Network - In this experiment we observed that the change in the distribution affects the FPR/FNR; for Random

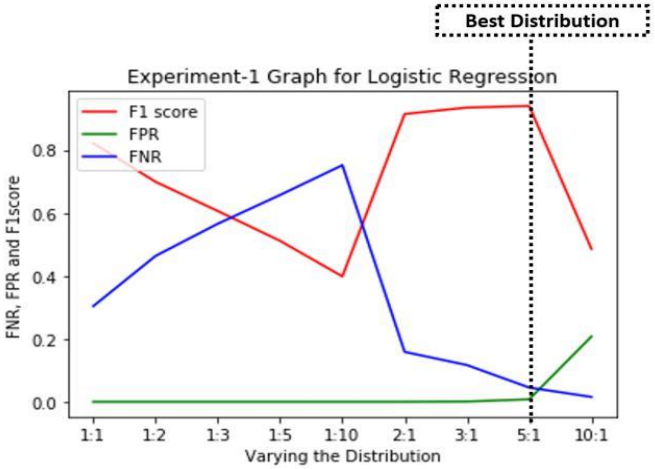

Fig. 2. Logistic Regression- F1-score, FPR, FNR

Forest distributions such as: 2:1, 3:1, 5:1, 10:1 which gave better results as compared to other distributions. We can observe in Fig 6. that we obtained a FPR and FNR score of 0.0018 and 0.0198 while maintaining the F1-score of 0.9807 in Fig 5.

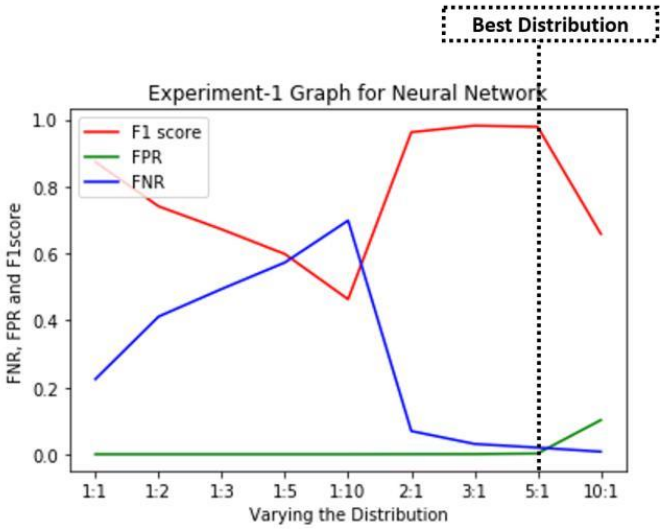

Fig. 3. Neural Network- F1-score, FPR, FNR

Among these three algorithms, Neural Network performed better than Random Forest and Logistic Regression, resulting in a higher F1-score of 0.9807 with corresponding FPR/FNR values of 0.0018 and 0.0198 respectively.

B. **Experiment** 2

1) Random Forest -

In this experiment we observed that changing the threshold value affects the FPR/FNR values. We tuned the threshold val-ues such that FPR/FNR values are further lowered maintaining the F1-score of 0.98. We observed that in Fig 4. we could adjust the balance of FPR from 0.0027 to 0.0014 by varying
the threshold from 0.40 to 0.60. Similarly we observed in Fig that we could adjust the balance of FPR from 0.0495 to 0.0418 by varying the threshold from 0.60 to 0.40.

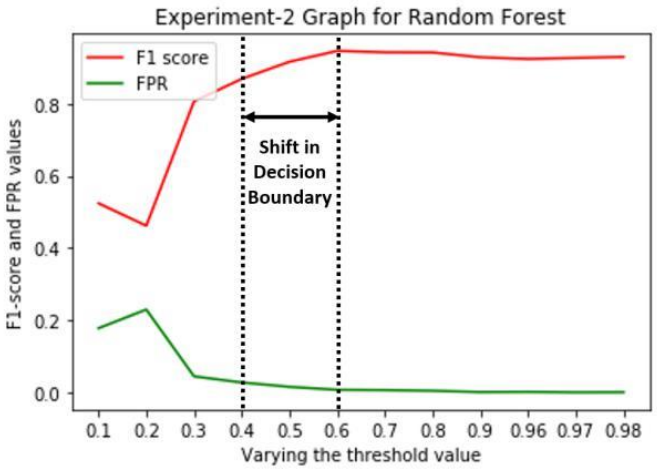

Fig. 4. Varying the Threshold values for FPR

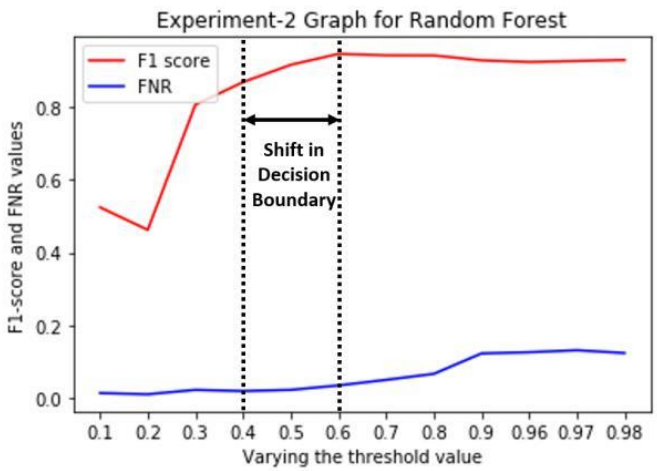

Fig. 5. Varying the Threshold values for FNR

2) Logistic Regression -

In this experiment we observed that changing the threshold value affects the FPR/FNR values. We tuned the threshold values such that FPR/FNR values are further lowered maintaining the F1-score of 0.95. We observed in Fig 6. that we could adjust the balance of FPR from 0.0042 to 0.0018 by varying the threshold from 0.50 (default) to 0.80. Similarly we observed in Fig 7. that we did not adjust the balance of FPR as it is at its best fit of 0.0605 at the default threshold of 0.50 if we further tune it, we might affect the F1-score.

3) Neural Network -

In this experiment we observed that changing the threshold value affects the FPR/FNR values. We tuned the threshold values such that FPR/FNR values are further lowered maintaining the F1-score of 0.98. We observed that in Fig 8. we could adjust the balance of FPR from 0.0028 to 0.0003 by varying the threshold from 0.50 (default) to 0.99. Similarly in Fig 9. we could adjust the balance of FPR from 0.0209 to 0.0198 by varying the threshold from 0.50 (default) to 0.80.

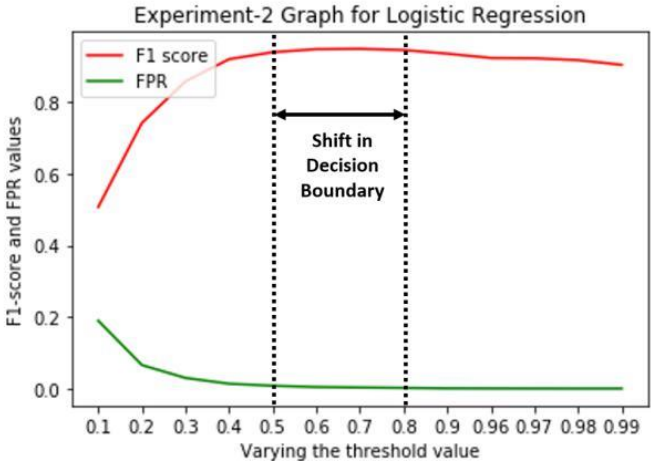

Fig. 6. Varying the Threshold values for FPR

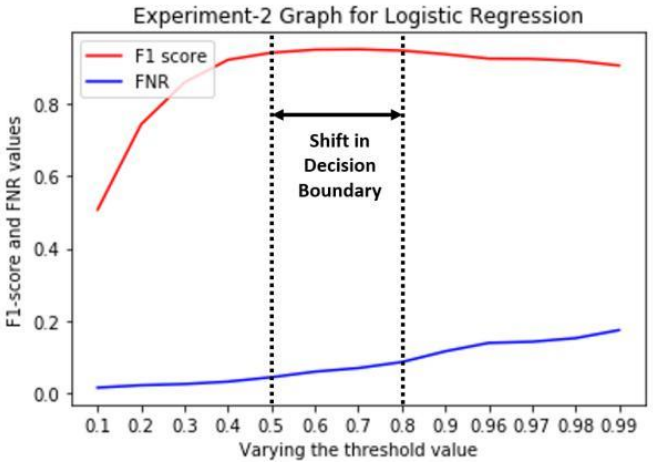

Fig. 7. Varying the Threshold values for FNR

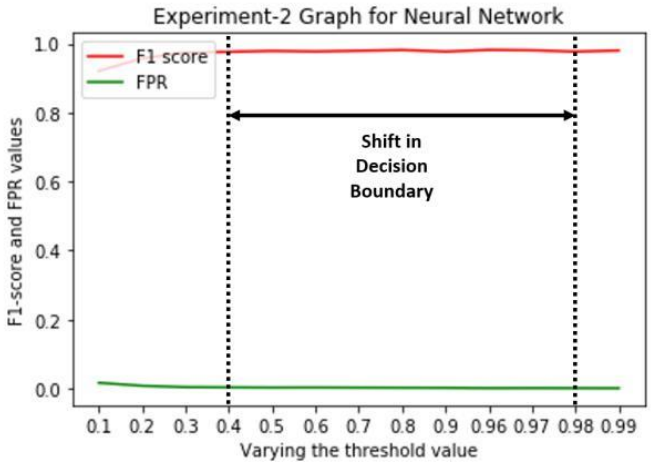

Fig. 8. Varying the Threshold values for FPR

Among these three algorithms, Neural Network performed better than Random Forest and Logistic Regression, resulting in a higher F1-score of 0.98 with corresponding FPR/FNR values of 0.0003 and 0.0198 respectively.

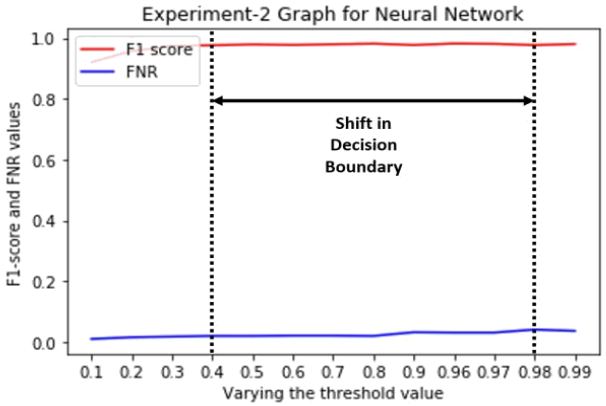

Fig. 9. Varying the Threshold values for FNR

V. CONCLUSION AND FUTURE WORK

In our paper we argue that F1-score does not give the entire picture, we demonstrate how to adjust the FPR/FNR trade-off, so the users can tune their models based on their deployment scenarios. We proposed two techniques to achieve this, in the first technique we learned how to adjust the FPR/FNR trade-off by varying the distribution and in the second technique, we learned how much we can adjust the FPR/FNR trade-off while maintaining a high F1-score by varying the threshold values. Our model generated remarkably lower False Positives and False Negatives after tuning. While we have outlined several combinations that we used to obtain better results based on our dataset, the other researchers can use the same techniques and tune the model the way they require. Identifying the Phishing webpages by performing image processing, using Domain name server (DNS) look-up information, etc. is something we will address in our future work.

REFERENCES

1. Marchal, S. and Asokan, N., 2018. On designing and evaluating phishing webpage detection techniques for the real world. In 11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18).
2. Marchal, S., Saari, K., Singh, N. and Asokan, N., 2016, June. Know your phish: Novel techniques for detecting phishing sites and their targets. In 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS) (pp. 323-333). IEEE.
3. Tharwat, A., 2018. Classification assessment methods. Applied Computing and Informatics.
4. Zhang, X.D., 2010. An effective method for controlling false discovery and false nondiscovery rates in genome-scale RNAi screens. Journal of biomolecular screening, 15(9), pp.1116-1122.
5. De Smet, F., Moreau, Y., Engelen, K., Timmerman, D., Vergote, I. and De Moor, B., 2004. Balancing false positives and false negatives for the detection of differential expression in malignancies. British Journal of Cancer, 91(6), p.1160.
6. Vazhayil, A., Harikrishnan, N.B., Vinayakumar, R., Soman, K.P. and Verma, A.D.R., 2018. PED-ML: Phishing email detection using classical machine learning techniques. In Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA) (pp. 1-8). Tempe, AZ, USA.

7. Whittaker, C., Ryner, B. and Nazif, M., 2010. Large-scale automatic classification of phishing pages.
8. Baader, G. and Krcmar, H., 2018. Reducing false positives in fraud detection: Combining the red flag approach with process mining. International Journal of Accounting Information Systems, 31, pp.1-16.
9. Miyamoto, D., Hazeyama, H. and Kadobayashi, Y., 2008, November. An evaluation of machine learning-based methods for detection of phishing sites. In International Conference on Neural Information Processing (pp. 539-546). Springer, Berlin, Heidelberg.
10. Ma, J., Saul, L.K., Savage, S. and Voelker, G.M., 2009, June. 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 (pp. 1245-1254). ACM.
11. Fette, I., Sadeh, N. and Tomasic, A., 2007, May. Learning to detect phishing emails. In Proceedings of the 16th international conference on World Wide Web (pp. 649-656). ACM.
12.] Darling, M., Heileman, G., Gressel, G., Ashok, A. and Poornachandran, P., 2015, July. A lexical approach for classifying malicious URLs. In 2015 international conference on high performance computing simula-tion (HPCS) (pp. 195-202). IEEE.
13. Le, A., Markopoulou, A. and Faloutsos, M., 2011, April. Phishdef: Url names say it all. In 2011 Proceedings IEEE INFOCOM (pp. 191-195). IEEE.
14. Whittaker, C., Ryner, B. and Nazif, M., 2010. Large-scale automatic classification of phishing pages.
15. Ra, V., HBa, B.G., Ma, A.K., KPa, S., Poornachandran, P. and Verma, A.D.R., 2018. DeepAnti-PhishNet: Applying deep neural networks for phishing email detection. In Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA) (pp. 1-11). Tempe, AZ, USA.
16. Zhang, N. and Yuan, Y., 2013. Phishing detection using neu-ral network. Department of Computer Science, Department of Statistics, Stanford University, CA, available at: http://cs229. stan-ford. edu/proj2012/Zhang Yuan-Phishing Detection Using Neural Network. pdf (accessed April 23, 2016).[Google Scholar].
17. Xiang, G., Hong, J., Rose, C.P. and Cranor, L., 2011. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security (TISSEC), 14(2), p.21.
18. De Smet, F., Moreau, Y., Engelen, K., Timmerman, D., Vergote, I. and De Moor, B., 2004. Balancing false positives and false negatives for the detection of differential expression in malignancies. British Journal of Cancer, 91(6), p.1160.
19. APWG, G.A. and Manning, R., 2008-2018. APWG Phishing Reports.
20. Seleniumhq.org. (2019). Selenium WebDriver. [online] Available at: https://www.seleniumhq.org/projects/webdriver/ [Accessed 7 Jul. 2019].
21. Phishtank.com. (2019). PhishTank — Join the fight against phishing. [online] Available at: http://phishtank.com/ [Accessed 7 Jul. 2019].
22. Majestic.com. (2019). Majestic Million - Majestic. [online] Available at: https://majestic.com/reports/majestic-million [Accessed 7 Jul. 2019].
23. Dr. Gowtham R., Gupta, J., and Gamya, P. G., Identification of phishing webpages and its target domains by analyzing the feign relationship, Journal of Information Security and Applications (Elsevier) (SNIP: 1.186) , vol. 35, pp. 75-84, 2017.
24. Chrome Driver (2019). Google Chrome WebDriver. [online] Available at: http://chromedriver.chromium.org/downloads [Accessed 7 Jul. 2019].