International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6S3, September 2019

Feature Selection for Phishing Detection with Machine Learning

Anierudh Sundararajan, Gilad Gressel, Krishnashree Achuthan

Abstract*—In the last 10 years machine learning has been widely used to combat phishing attacks. The most common approach was to build a classification model that would be able to detect whether or not a given URL or website is a phishing attack. In order to effectively detect a phishing page with machine learning we must find an effective method to represent websites (both phish and benign) as features which can be fed into a machine learning model. One of the challenges faced by these approaches was to find a good set of features to represent the phishing and benign sites. Within the last 10 years hundreds of different features had been proposed and used to great success [6] [7] [9]. However, due to the curse of dimensionality, use of all available features will exponentially increase the sparsity of the dataset, lowering the odds of successful classification. In this work we extract 31 features that had been commonly used in the literature and perform an in depth feature ranking analysis in order to find the most effective features for phishing detection. Using both filter and wrapping methods we were able to find 23 effective features for phishing detection. The F1-score for all 31 features was 0.88 and time taken to train the multilevel perceptron model was 45.49 seconds and the size of the data used is 100k. Using these 23 features we were able to train a model that has 0.99 F1-score and which was comparable with all previous work while reducing the overall dimensionality of the data and time taken to train the model was 43.71 seconds.

***Index Terms*—Phishing, Cyber Security, Feature Selection, Machine Learning**

I. INTRODUCTION

In 2018 the Anti Phishing Working Group (APWG) reported 65,493 Phishing attacks which cost the global economy 13 billion dollars [17]. Phishing is a masquerading attack in which attackers attempt to steal victims credentials by pretending to be legitimate websites. As smartphones proliferate our lives, maintaining a secure login to manage sensitive material such as banking, email, social media, company sites, and health records has become paramount. The common citizen is required to memorize numerous passwords and login ids, which often leads to password reuse. Attackers leverage this situation by performing phishing attacks, pretending to be legitimate websites, they lure victims into typing their credentials into fraudulent (phishing) sites. Once an attacker has phished a victim they will either sell the username/password combination or attempt to break into other sites using the username/password provided. In order to detect and prevent phishing attacks there have been numerous studies which implement machine learning models to detect phishing [1] [8] [11] [15] [16]. Machine learning is well suited to solve phishing attacks because it identifies the patterns in the data (collected sites) and is then able to automatically predict whether a site will be a phish or benign.

In order to create phishing detection models with machine learning we need to collect a dataset, extract features from the dataset, and then train a machine learning model. In order to collect a dataset for phishing, researchers have collected URLs and source code from known phishing and legitimate sites. Once a dataset is collected, researchers then *extract* features from the raw data. This step is highly subjective and often referred to as “black magic” in the machine learning community [18]. The process of extracting features can be simple natural language processing (NLP) statistics or complex language models. In the case of phishing detection there have been a set of standard features which are commonly used, these are generally different types of ‘count’ features, counting the occurrence of certain symbols within the URL or source code, NLP features such as Term Frequency - Inverse Document Frequency (TF-IDF), and binary features such as the existence of certain symbols or blacklisted words. While all these types of features are intuitively useful, without rigorous testing we cannot be sure of the efficacy of a feature. The curse of dimensionality refers to the sparsity that grows as the number of features we utilize increases. When we use more features to represent a datapoint, the amount of data we need to cover all situations increases. Simply put, the more features we use, the more dimensions we have and thus the more data we require in order for a model to find better decision boundaries [21]. Thus, the curse dimensionality is directly at odds with the desire to represent our data with more features. Intuitively more features represent the data to the model better, but the curse tells us that the more features we choose the sparser the data will be and thus the models may perform more poorly. In order to combat the curse, we need to be selective with the features that we choose. Due to the curse of dimensionality it is in fact quite possible to use fewer features and obtain better results.In this paper we perform careful feature selection on the common features used in the literature. We use two types of feature selection, filtering and wrapping. Filtering methods make use of statistical tests in order to find a correlation be- tween a feature and a target variable, while wrapping methods train machine learning models with different combinations of features searching for the best results.

Manuscript published on 30 September 2019.
* Correspondence Author (s)
Anierudh Sundararajan, Cyber Security Systems and Networks, Amrita Vishwa Vidyapeetham Amritapuri, India,
Gilad Gressel, Georgia Institute of Technology Atlanta, USA.
Krishnashree Achuthan, Cyber Security Systems and Networks, Amrita Vishwa Vidyapeetham Amritapuri, India,

We see the following as our contribution in this work:

- We gather a unique dataset of 100k URLs and associated source code (50:50, phish:benign) and extract the most commonly used 31 features from the dataset.
- We perform 3 separate filtering tests on the 31 features, chi-square, mutual information, and the pearson correlation.
- We perform a wrapper method of recursive feature elimination with three different machine learning models (random forest, logistic regression, and neural networks).
- We are able to conclusively select 23 features from the pool.
- These features yield an increase of f1-score from 0.88 to 0.99.

II. RELATED WORK

Mohammad, R.M., Thabtah, F. and McCluskey, L. extracted all the possible features from the website using a tool, instead of human interaction and judging the importance of each feature. They were also able to provide a group of features which are effective in detect phishing based on precise rules. They collected phishing URLs from phishtank and used javascript and PHP script program was used to extract different kind of features such as abnormal based, domain based, HTML and javascript based and address bar based. They collected 17 features that distinguish between phishing and legitimate websites, but the dimensionality of the data was larger [20].

Basnet, R.B., Sung, A.H. and Liu, Q. proposed how machine learning helps to improve phishing detection by selecting relevant features using feature selection algorithm to reduce the dimensionality of the data. Their dataset is taken from phishtank for phishing URLs and DMOZ, Yahoo directory for legitimate URLs. They extracted 138 commonly used features, out of which 100 are URL based and the remaining are source code based. The feature selection algorithm used was a correlation based genetic search (42 features) and a wrapper method of greedy forward search (12 features). To find the performance of the selected features they used classifiers such as logistic regression, random forest, multilayered perceptron, and naive bayes. The wrapper method performed well compared to correlation-based feature selection by obtaining FPR 0.2 and FNR 0.5. In this work, we use URL and source code based features [1].

Thabtah, F. and Abdelhamid, N. wanted to compare different feature selection techniques in websites to determine the minimal set of features for detecting phishing. The dataset they used is the Yahoo directory for legitimate URLs, phishtank for phishing URLs. Feature selection methods used are correlation based, chi-square, information gain, wrapper method. Classifiers used are PART, RIPPER, and decision trees. They used
30 features from URL and web-based content. Decision tree (error rate 24.44%) and PART (error rate 23.42%) gave good results [7].

Ubing, A.A., Jasmi, S.K.B., Abdullah, A., Jhanjhi, N.Z. and Supramaniam, M. research focused on evaluating phishing and legitimate webpages based on accuracy by integrating feature selection algorithm with ensemble methods. The dataset is taken from UCI machine learning repository. Feature selection method used is random forest regressor. Classifiers used are the random forest, logistic regression, prediction models (ensemble learning). URL based features and source code features are taken. By combining all classifiers they are able to make a predictive model performed better compared to individual models [8].

Hall, M.A. and Smith, L.A. compare features selected from a filter method correlation based approach and wrapper method on two machine learning models. 12 standard datasets were used from the UCL repository. Feature selection algorithm used is correlation based, wrapper method using cross- validation. Classifiers used are naive bayes and decision trees. Correlation-based feature selection method gave good results than wrappers in most of the scenarios [9].

Lee, C. and Lee, G.G. research focus on reducing redundancy between the features using information gain and divergence based feature selection method for text categorization. The dataset used for text categorization is Reuters-21578 corpus and world wide web knowledge base consist of HTML web pages. Feature selection methods used are information gain, Maximal Marginal Relevance (MMR), greedy feature selection method and classifiers used are naive bayes, SVM, K-nearest neighbors, probabilistic indexing. MMR feature selection gave good result [10].

Cai, J., Luo, J., Wang, S. and Yang, S. surveyed how to reduce high dimensional data by using feature selection methods from supervised, unsupervised, and semi supervised learning. They used two sets of data from gene expression, colon, and CNS. The feature selection algorithms used were information gain, maximum relevance minimum redundancy, JMI, ReliefF, and SVM-RFE. They ran on classifiers: naive bayes, KNN, SVM, random forest, K-means, RF-NN, BP- NN, and hierarchical clustering. To reduce high dimensional problem, researchers use heuristic method with polynomial time complexity. Some challenges still remain: extreme data for machine learning, online feature selection, and ensemble feature selection [6].

Venkatesh, B. and Anuradha, J. research focus on combining mutual information filter method and recursive feature elimination wrapper method and measuring performance on three benchmark dataset from UCI repository ionosphere, libras movement, and clean.

They ran on random forest classifier and hybrid model. Based on the analysis of the results it is evident that hybrid model gave good result [25].

Reddy, N.S.C., Nee, S.S., Min, L.Z. and Ying, C.X. wanted to study which selected features play a key role in heart disease prediction on cleveland, switzerland, hungarian, v.a. medical and statlog project heart dataset from UCI repository. Combined all the five datasets into a single dataset and used here. Three different train and test split 60%-40%, 70%-30%, and 80%-20% were used respectively. The feature selection algorithms used were RFE, correlation, variable importance. Used commonly selected eight and six features by ranking them.

They ran on classifiers: naive bayes, KNN, SVM, random forest, and neural network. The random forest classifier on eight features with 80%-20% split gave an accuracy of 95%, sensitivity of 0.94, and specificity of 0.97 [26].

In this work, we took data from phishtank and majestic million, used filter and wrapper feature selection techniques. We used filter method with statistical tests chi-square, information gain, and pearson correlation coefficient, and wrapper method recursive feature elimination. We took features from URL, source code, redirection chain, certificate and title of the webpage. Algorithms used are logistic regression, random forest, and Multilevel perceptron to analyze the performance and find a good set of features for phishing detection.

III. APPROACH

A. Dataset and Feature Extraction

The data was created at Amrita Vishwa Vidyapeetham Centre for Cybersecurity Systems and Networks. We collected legitimate URLs from majestic million. Majestic million is a repository which contains 1 million popularly used URLs [19]. Phishing data was collected from phishtank, which is a community based site that maintains phishing URLs [22]. The URLs from phishtank are updated daily based on human reporting. This phishtank is a commonly used repository for phishing data. We used web scraping tool selenium, with google chrome web driver to scrape the data from each URL [28]. We split the majestic million data into 1000 or 2000 and gave as input to the crawler to scrape majestic data. Phishtank is updated on a daily basis, each day we collected the updated list of 500 to 1000 URL's and used that data to scrape phishing data. From each URL we extracted raw data, comprised of date and time stamp, URL, title, source code, redirection chain, request History, header Information, certificate Information, screen shot of the webpage, etc. These attributes cover the whole webpage. Date and time stamp gives information on what date and time the particular URL was crawled, title gives the title of the webpage, redirection chain gives information on redirection URL's, request history gives whether the redirected URL is 302 or 301, header information gives response information of the webpage, certificate information the attributes of the certificate and screenshot takes the picture of the webpage. The extra information regarding the dataset can be seen in Table I. We chose features from URL, source code, redirection chain, request history, certificate, and title. Most of the literature papers use URL and source code based features [11] [12] [15] [16]. We extracted a total of 31 features from the raw data can be seen in Table II.

TABLE I
ADDITIONAL INFORMATION ON THE DATASET

Parameter	Majestic Data	Phishing Data
Total count of data	50,000	50,000
Overall file size (raw data)	4.18GB	1.62GB
File size (after extracting features)	7.88MB	7.30MB
Data collected from	Majestic Million	Phish-tank
Total number of days we ran the crawler (daily basis)	32 days	45 days
Data collected	May 1st 2019 - June 11th 2019	May 1st 2019 - June 30th 2019
System configuration we used to run the crawler	14 GB RAM 500GB hard disk, ubuntu 16.04 server	
Number of extracted features from the raw data	31	

TABLE II
THE 31 STANDARD FEATURES FROM THE LITERATURE

Count features	*Count features*	*Binary features*
Equal to	Number of 301 Redirection	Certicate present or not
Vowels and Consonants Ratio	Number of 302 Redirection	HTTPS
Digit/Letter Ratio	Number of Redirections	Special Symbols
Digits	Source code Href	***Statistic feature***
Question Mark	Source code Iframes	Domain Length
Non Alphabetic	Vowels Count	URL Length
Number of Dots	Source code Images	Title Length
Dash	Source code Links	Source code Length
Alphabets	Source code Java Script	Source code title
Sub Domains	Source code forms	
Dollar	Source code CSS	
Consonant Count	Source code src	

B. Training the Model

The training data was used in the ratio of 1:1, Phishing:Legit and testing in the ratio of 1:100, Phishing:Legit. We choose 1:1 for training because we need equal proportion of data for the machine learning models to learn the patterns and 1:100 for testing as that is actual ratio based on today's scenario the number of legitimate web pages is higher than phishing ones [13].

C. Filter Method

In the filter method, feature selection is independent of the machine learning algorithm used. The features are ranked using statistical tests. Filter methods are faster to compute and less accurate [9]. They are less accurate because filter methods rank the features by finding the relationship between feature and the target variable and this does not find any interrelationship between features. One of the forms of filter method is Univariate feature selection. This method examines each feature individually and identifies the relevance of the feature with respect to the target variable. It is then possible to rank the features using their scores. Some of the commonly used statistical tests are: mutual information, chi-square, information gain, pearson correlation coefficient, and variance threshold. Mutual information ranks the features based on amount of information exchanged with the target variable [23]. Chi-square ranks the features based on the difference between expected value and obtained value when compared with the target variable [14]. Information gain ranks features based on entropy. It gives the amount of information provided from each feature. In text categorization based feature selection, information gain is commonly used [10] [27]. Pearson correlation coefficient identifies the linear correlation between two entities (a feature and target variable).

It gives values between -1 to 1, where -1 means negative correlation (where one variable increases and the other decreases), 0 means no correlation, 1 means positive correlation (there is some correlation between the two values). The drawback is it finds only a linear correlation between the two variables. If there are non-linear features it won't consider them. In variance threshold, by giving a threshold value, this method calculates the variance of each feature and gives the feature whose value is more than that threshold value [3].

1) *Experiment 1:* The module used from sklearn is SelectKbest. SelectKbest is a feature selection method that ranks the features according to k highest scores [3]. Using SelectKbest, we use statistical methods: pearson correlation coefficient, mutual information, and chi-square to rank all the features. We took 100,000 data points (50k each) and split the data into 80% training and 20% testing. We took 40,000 legitimate and 40,000 phishing data points as training data and 10,000 phishing and 10,000 legitimate data points as testing data for all experiments. The testing data is of the ratio 1:100, Phishing:Legit (99:9900). We ran three algorithms: Random Forest, Logistic Regression and Multilayer Perceptron evaluating with performance metrics such as accuracy, F1-score, FPR, and FNR. In order to choose the number of features from each statistical test, we use an elbow graph. Elbow graph is a method, which helps to find the drop-in value in the graph plotted to choose the number of features to be extracted [24]. If there is no clear elbow, we can select a upper 75% quartile. Then we select the upper 75% quartile of the scores, and whichever features are associated with the upper 75% quartile.

We obtained the scores of all the features using chi-square. From chi-square we selected two features after that there was a drop-in value, refer Fig 1. They are source code href and source code length. Refer to Table III under column CS for selected features.

We obtained the scores of all the features using mutual information. From mutual information we selected 23 features from upper 75% quartile as there is no proper elbow graph formed, refer Fig 2. Refer to table III under column MI for the selected features.

We obtained the scores of all the features using pearson correlation coefficient. From pearson correlation coefficient we selected 23 features from upper 75%quartile, refer to Fig 3. Refer to Table III under column PCC for the selected features.

We have selected three sets of features one from each statistical test. To find the number of features to be used, we found the union set of features from the three statistical tests. We did this to know how the selected features performs with the machine learning models. This number can be used in *Experiment 2* to find the corresponding features. The count of features used was 26. Refer to Table III.

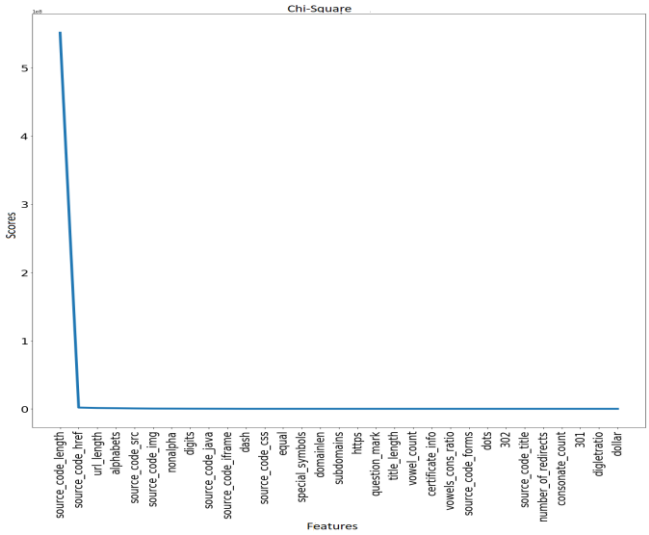

Fig. 1. Chi-Square Scores Found Using SelectKbest

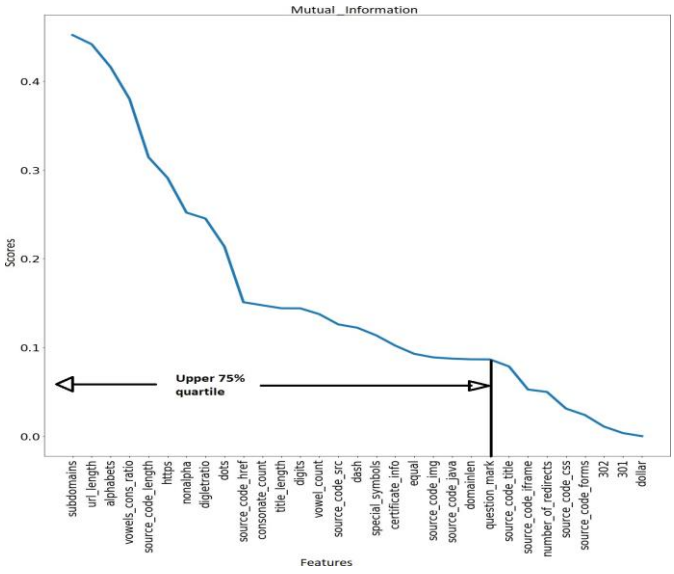

Fig. 2. Mutual Information Scores Found Using SelectKbest

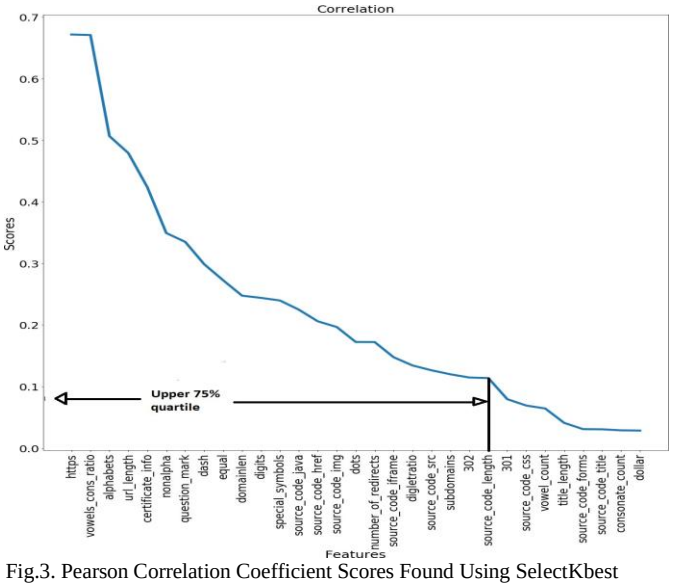

Fig.3. Pearson Correlation Coefficient Scores Found Using SelectKbest

D.Wrapper Method

With the wrapper method, a feature subset was selected and trained with the classification algorithm. Based on classifier performance the features can be added or eliminated. As it is done iteratively it takes more computational power than filtering. It gives accurate results compared to filter methods [4]. Some of the commonly used wrapper methods are: recursive feature elimination, forward selection, backward elimination. Recursive feature elimination (RFE) is a greedy search algorithm, where a subset of features are taken, the model is ran and performance is measured, likewise all combinations of the subsets is taken and performance is measured. The features are ranked based on feature importance using either random forest classifier or support vector machine. In this paper, we used RFE with random forest classifier as it helps to find the nonlinear relationship between the features and the target variable in high dimensional data. Whichever combination has best performance is considered as the set of features. RFE is a recursive iteration process. The drawback of RFE is if the dimensionality of data is large, this method becomes expensive in terms of time, and as it needs to check all subset of features we need more computational power [25]. Forward selection starts with 1 feature and runs a model, and another feature and does the same. This process continues until the performance of the model doesn't increase and becomes stable. By this way, it selects the features [2]. Backward elimination starts with all features and in each round, it eliminates the least significant feature and performance is measured. This process continues until the performance of the model doesn't increase and becomes stable. By this way, it selects the features [2].

Experiment 2: In order to give RFE a target number of features to select from, we chose to use the union set of features selected from experiment 1 which was a total of 26 features. we ran RFE with a target of 26 features and got the top 26 features from it. Refer to Table III under column RFE for selected features.

IV. RESULTS DISCUSSION AND PERFORMANCE ANALYSIS

In total we obtained five sets of features and ran 15 models. In Experiment-1, the 23 mutual information features ran on multilayer perceptron gave an F1-score of 0.99 with FPR 0.01 and FNR 0.000. Interestingly, the 26 features from the union of the three statistical tests and 23 features from the pearson correlation coefficient ran on multilayer perceptron gave an similar F1-score of 0.99 with FPR 0.019 and FNR 0.000. The performance of the 23 mutual information features is the highest compared to other models as seen in Table IV. The features selected by these three high performing models share 95% similarity of features as seen in Table III. The difference is the features not used by Pearson correlation coefficient are title length, vowel count and consonant count and the features not used by mutual information are the number of redirections, source code iframe, and the number of 302 redirections. Number of 301 redirects, dollar sign, and source code title are not selected as features by any of the methods as the rank value obtained is low in all the methods.

In Experiment-2, we use RFE and ranked the top 26 features. Most of the features obtained by RFE were also found in the union set of features from the three statistical tests, pearson correlation, and mutual information as seen in Table III. Source code css, source code forms features are not selected in experiment-1. The model which ran on logistic regression gave an F1-score of 0.91 with FPR 0.162 and FNR 0.0001, is the highest compared to the other two models as seen in Table V. Overall the mutual information based features ran on multi- level perceptron had the best performance (F1-score 0.99 FPR 0.01 FNR 0.000) compared to other models.

TABLE IV
EXPERIMENT-1 UNIVARIATE FEATURE SELECTION-PERFORMANCE ANALYSIS

Statistical tests	Models	F1-score	FPR	FNR
Chi-Square	Random Forest	0.10	0.944	0.002
	Logistic Regression	0.03	0.984	0.003
	Multilayer Perceptron	0.06	0.966	0.004
Mutual Information	Random Forest	0.88	0.22	0.000
	Logistic Regression	0.90	0.169	0.0001
	Multilayer Perceptron	**0.99**	**0.01**	**0.000**
Pearson Correlation	Random Forest	0.90	0.181	0.000
	Logistic Regression	0.91	0.155	0.0001
	Multilayer Perceptron	0.99	0.019	0.000
Union set of features from the three statistical tests	Random Forest	0.88	0.208	0.000
	Logistic Regression	0.90	0.176	0.0001
	Multilayer Perceptron	0.99	0.019	0.000

TABLE V
EXPERIMENT-2 RFE-PERFORMANCE ANALYSIS

	Models	F1-Score	FPR	FNR
RFE	Random Forest	0.88	0.208	0.000
	Logistic Regression	**0.91**	**0.162**	**0.0001**
	Multilayer Perceptron	0.90	0.188	0.000

TABLE III
EXPERIMENT 1 AND 2 SELECTED FEATURES

CS- Chi-square, MI- Mutual Information, PCC- Pearson Correlation Coefficient, RFE- Recursive Feature Elimination, Y – Particular feature is selected

Features	Filter Method			Wrapper Method
	CS	MI	PCC	RFE
Dash		Y	Y	Y
Equal to		Y	Y	
Alphabets		Y	Y	Y
Dots		Y	Y	Y
Number of Redirects			Y	Y

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Vowels Count		Y		Y
Title Length		Y		Y
Consonant Count		Y		Y
Digits		Y	Y	Y
Certificate Information		Y	Y	Y
Https		Y	Y	Y
URL Length		Y	Y	Y
Domain Length		Y	Y	Y
Sub Domains		Y	Y	Y
Digit Letter Ratio		Y	Y	Y
Non Alphabetic		Y	Y	Y
Vowels Consonant Ratio		Y	Y	Y
Question Mark		Y	Y	Y
Number of 302 redirects			Y	Y
Number of 301 redirects				
Special Symbols		Y	Y	
Source code Javascript		Y	Y	Y
Source code Forms				Y
Source code CSS				Y
Source code Href	Y	Y	Y	Y
Source code Title				
Source code Src		Y	Y	Y
Source code Length	Y	Y	Y	Y
Source code Iframe			Y	Y
Source code Image		Y	Y	Y
Dollar sign				

TABLE VI
BEST PERFORMED FEATURE SET

Features	Mutual Information
Dash	Y
Equal to	Y
Alphabets	Y
Dots	Y
Vowels Count	Y
Title Length	Y
Consonant Count	Y
Digits	Y
Certificate Information	Y
Https	Y
URL Length	Y
Domain Length	Y
Sub Domains	Y
Digit Letter Ratio	Y
Non Alphabetic	Y
Vowels Consonant Ratio	Y
Question Mark	Y
Special Symbols	Y
Source code Javascript	Y
Source code Href	Y
Source code Src	Y
Source code Length	Y
Source code Image	Y

Y – Particular feature is selected from that statistical test

V. CONCLUSION

From the results, we can conclude that the 23 mutual information features run on multilayer perceptron obtained the best performance metrics (F1-score 0.99, FPR 0.01 and FNR 0.000) compared to other methods. The best set of features selected by mutual information can be seen in Table VI. These features were the most impactful in phishing detection. In this work, we found the 23 best features for phishing detection using a standard set of feature selection algorithms.

REFERENCES

1. Basnet, R.B., Sung, A.H. and Liu, Q., 2012, June. Feature selection for improved phishing detection. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 252-261). Springer, Berlin, Heidelberg.
2. Venkatesh, B. and Anuradha, J., 2019. A Review of Feature Selection and Its Methods. Cybernetics and Information Technologies, 19(1), pp.3-26.
3. Scikit-learn.org. (2019). 1.13. Feature selection scikit-learn 0.21.2 documentation. [online] Available at: https://scikit-learn.org/stable/modules/feature selection.html [Accessed 20 Jun. 2019].
4. Maldonado, S. and Weber, R., 2009. A wrapper method for feature selection using support vector machines. Information Sciences, 179(13), pp.2208-2217.
5. Investigations, E., Program, W. and Survey, I. (2019). APWG — Unifying The Global Response To Cybercrime. [online] Apwg.org. Available at: https://apwg.org/ [Accessed 24 Jun. 2019].
6. Cai, J., Luo, J., Wang, S. and Yang, S., 2018. Feature selection in machine learning: A new perspective. Neurocomputing, 300, pp.70-79.
7. Thabtah, F. and Abdelhamid, N., 2016. Deriving correlated sets of website features for phishing detection: A computational intelligence approach. Journal of Information and Knowledge Management, 15(04), p.1650042.
8. Ubing, A.A., Jasmi, S.K.B., Abdullah, A., Jhanjhi, N.Z. and Supramaniam, M., 2019. Phishing Website Detection: An Improved Accuracy through Feature Selection and Ensemble Learning. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICA- TIONS, 10(1), pp.252-257.
9. Hall, M.A. and Smith, L.A., 1999, May. Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In FLAIRS conference (Vol. 1999, pp. 235-239).
10. Lee, C. and Lee, G.G., 2006. Information gain and divergence-based fea- ture selection for machine learning-based text categorization. Information processing and management, 42(1), pp.155-165.
11. 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 and simulation (HPCS) (pp. 195-202). IEEE.
12. 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.

13. 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).
14. Jin, X., Xu, A., Bie, R. and Guo, P., 2006, April. Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. In International Workshop on Data Mining for Biomedical Applications (pp. 106-115). Springer, Berlin, Heidelberg.
15. Marchal, S., Armano, G., Grndahl, T., Saari, K., Singh, N. and Asokan, N., 2017. Off-the-Hook: an efficient and usable client-side phishing prevention application. IEEE Transactions on Computers, 66(10), pp.1717- 1733.
16. Liu, C., Wang, L., Lang, B. and Zhou, Y., 2018, January. Finding effective classifier for malicious URL detection. In Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences (pp. 240-244). ACM.
17. apwg trends reports q3 2018. (2018). [ebook] Available at: https://docs.apwg.org//reports/apwg trends report q3 2018.pdf [Accessed 14 Jul. 2019].
18. Luo, W., Phung, D., Tran, T., Gupta, S., Rana, S., Karmakar, C., Shilton, A., Yearwood, J., Dimitrova, N., Ho, T.B. and Venkatesh, S., 2016. Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view. Journal of medical Internet research, 18(12), p.e323.
19. Majestic.com. (2019). Majestic Million - Majestic. [online] Available at: https://majestic.com/reports/majestic-million [Accessed 11 Jul. 2019].
20. Mohammad, R.M., Thabtah, F. and McCluskey, L., 2012, December. An assessment of features related to phishing websites using an automated technique. In 2012 International Conference for Internet Technology and Secured Transactions (pp. 492-497). IEEE.
21. Hua, J., Tembe, W.D. and Dougherty, E.R., 2009. Performance of feature-selection methods in the classification of high-dimension data. Pattern Recognition, 42(3), pp.409-424.
22. Phishtank.com. (2019). PhishTank — Join the fight against phishing. [online] Available at: https://www.phishtank.com/index.php [Accessed 13 Jul. 2019].
23. Kwak, N. and Choi, C.H., 2002. Input feature selection by mutual information based on Parzen window. IEEE Transactions on Pattern Analysis & Machine Intelligence, (12), pp.1667-1671.
24. Moradi, H. and Lee, S., 2005, August. Joint limit analysis and elbow movement minimization for redundant manipulators using closed form method. In International Conference on Intelligent Computing (pp. 423-432). Springer, Berlin, Heidelberg.
25. Venkatesh, B. and Anuradha, J., 2019. A Hybrid Feature Selection Approach for Handling a High-Dimensional Data. In Innovations in Computer Science and Engineering (pp. 365-373). Springer, Singapore.
26. Reddy, N.S.C., Nee, S.S., Min, L.Z. and Ying, C.X., 2019. Classification and Feature Selection Approaches by Machine Learning Techniques: Heart Disease Prediction. International Journal of Innovative Computing, 9(1).
27. Pragadeesh, C., Jeyaraj, R., Siranjeevi, K., Abishek, R. and Jeyakumar, G., 2019. Hybrid feature selection using micro genetic algorithm on microarray gene expression data. Journal of Intelligent Fuzzy Systems, 36(3), pp.2241-2246.
28. Seleniumhq.org. (2019). Selenium WebDriver. [online] Available at: https://www.seleniumhq.org/projects/webdriver/ [Accessed 7 Jul. 2019].

AUTHORS PROFILE

Mr Anierudh Sundararajan holds an M.Tech degree in cyber security systems and networks from Amrita Vishwa Vidyapeetham. His areas of interest are machine learning with security, cryptography, social media security, and cyber forensics.

Mr Gilad Gressel is currently a graduate student in Georgia Institute of Technology, working towards a degree in Computer Science with a specialization in machine learning. He is a consultant and researcher in the field of machine learning, developing and deploying machine learning models, and has two prior publications.

Dr Krishnashree Achuthan is a PhD graduate from Clarkson University, NY, USA. She is currently the dean of post graduate programs at Amrita Vishwa Vidyapeetham and heads the cyber security systems and networks graduate program and Amrita technology business incubator (Amrita TBI) at Amrita Vishwa Vidyapeetham. Her areas of interests are primarily cybersecurity policy and IoT security. She holds over 30 US patents and published over 50 research articles in various journals and conferences.