International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Phishing Detection System through Hybrid Machine Learning Based on URL

Prof. S.A Gaikwad
Assistant Professor
Department of Computer Science And Engineering
TPCTs COE Dharashiv, Maharashtra, India
sujatagaikwad414@gmail.com

Ram Vijay Thombare
Student
Department of Computer Science And Engineering
TPCTs COE Dharashiv, Maharashtra, India
ramthombare.cs@gmail.com

Abstract: Phishing attacks on the internet using a comprehensive dataset based on phishing URLs. The study utilizes various ML approaches like DT, LR, RF, NB, GBC, SVC and an innovative hybrid LSD model for enhancing cyber threat detection. In continuation, we have used a hybrid approach of hybrid multiple models, among which Stacking Classifier, an ensemble learning technique, has been used to merge the RF Classifier and MLP Classifier (as base classifiers). It is a LGBM Classifier based meta-estimator for the final prediction and therefore the extendability of the project to improve classification performance is enhanced. The effectiveness of the model is evaluated using evaluation metrics such as precision, accuracy, recall and F1-score. Based on the results, the hybrid LSD model is effective in combating phishing attacks and can provide a complete security solution for new cyber threats. The findings of this research can assist in strengthening cybersecurity measures and demonstrate the potential of ML in boosting internet security.

"Index Terms - *Phishing attacks, Machine learning algorithms, Cyber threat detection, Hybrid LSD model, Cyber security measures."*

I. INTRODUCTION

Phishing is a sophisticated method of fraud on the Internet that attempts to trick individuals into disclosing their personal information like credit card numbers, passwords or other information by masquerading as a legitimate website or organization, like a financial institution. It is extremely crucial to recognize phishing attempts to prevent any crucial information from being compromised and to prevent monetary damages. A type of AI known as machine learning is extremely useful in the fight against phishing. It does so by analyzing vast amounts of data, learning from it and using the knowledge gained to identify phishing emails. The upside is that ML systems can evolve to new and emerging phishing schemes, and are very powerful. One way to detect phishing is to scrutinize addresses of a website or URLs. One of the most common mistakes phishers make is misspelling the domain name, or adding too many subdomains. ML models are pretty accurate at catching such minute changes. Easy to integrate into a variety of internet applications, including web browsers, email client and business networks, are the key elements of effective phishing detection systems. These integrated solutions continuously capture incoming data and detect phishing attempts and immediately block users.

The Internet has become a part of everyday life in this technologically advanced world. So it provides us with many useful experiences in our communication, entertainment, education, commerce and so on in our lives. The internet has become a place for thieves to take the "real world" robbery into a virtual environment. The Internet offers conveniences in many things but has its disadvantages as for example the anonymity that the Internet provides to its users.[7] The number of Internet users is growing exponentially and the number of cybercrimes is proportionally growing rapidly. People and businesses are losing millions of dollars each day (Hong, 2012; Ragucci and Robila, 2006; University of Portsmouth, 2016). Phishing is one of the basic cybercrimes, which is geometrically expanding day by day.[12] With the growth of the internet era the bad actors are also increasing in number. With the age of websites as a daily phenomenon began the tendency of phishing assaults. Exploiting human weaknesses allows it to serve as an easier vehicle for victimizing consumers. Phishing websites are designed to resemble legitimate or other well known websites in order to fool the victims of the scam into succumbing to it.

The rogue website occasionally can be indistinguishable from the legal source. Most users of the Internet will not be able to tell the difference between the two. That resulted in the creation of blacklists of phishers. Phishing blacklists are databases of software that are maintained by specialties. They enable nonprofessional users to be alerted to possible phishing sites visited by them.

II. RELATED WORK

Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, Y. Si, F. Zhang, and J. In this paper, S. Dong et al. present a new system for identifying phishing, called "Phishpedia", which is extremely accurate and has a very small run-time overhead, using logos. This novel DL approach is more accurate than the existing ones for phishing identification, particularly in terms of detecting and matching logos. Not only does it outperform the existing strategies, but it also discovers new phishing sites and boosts the protection against phishing attempts. Phishpedia is an amazing and advantageous device to enhance cybersecurity. Cons: The more logos available and accessible on web pages, the better Phishpedia will perform. Continuous updating and maintenance are important to react to the changing techniques of phishing.[1]

Shirazi, Haynes and Raya present a new mobile-friendly phishing detection system using ANNs which performs the analysis of URL and HTML properties. They use state-of-the-art deep transformers like BERT, ELECTRA, RoBERTa, and MobileBERT to efficiently learn from URL text. The proposed system is rapidly trained, easy to maintain and can be deployed in real-time on mobile devices, thus offering effective solution to the mobile security. This guarantees competitive performance, giving a strong protection against phishing threats, and optimizing resources for better cybersecurity on mobile platforms. Cons: URL detection is only limited to detecting complicated phishing in real pages. Uses pre-trained transformers, some of which are not available and some of which are not of good quality. [2]

A. T. Akanchha examines the SSL certificate domain on phishing web sites, conducts analysis of attacker properties and suggests an automatic detection mechanism based on SSL certificate attributes. The research introduces an innovative and transparent way to detect phishing using SSL certificates, with high accuracy and user friendly Web API, using a DT [4] ML-based detection system, that is transparent and efficient. The study emphasizes the need for ongoing updates and changes to address the changing phishing landscape and provide a comprehensive approach to cybersecurity needs. Cons: Dependence on SSL certificates properties and system effectiveness is limited by the existence of new ways for attackers to fake real SSL certificates. The study of the scalability of the method to process a large number of domains is not done in depth.[3]

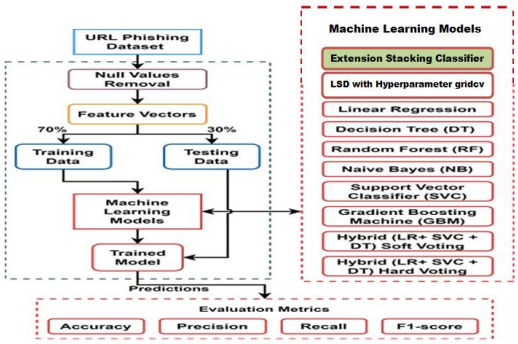
In the joint work of H. Shahriar, S. Nimmagadda's chapter discusses Network IDS using ML techniques as Gaussian Naive Bayes, logistic regression, DT [4] and neural networks. The study aims to distinguish the normal and abnormal network activities particularly on the TCP/IP level. The authors point out the need for testing in real-life scenarios and the scalability to assess the accuracy and efficiency of the approach for any real network intrusion detection, but the DT [4] performs well on public data sets. Cons: Assessments may not be accurate and may not be real-life situations or growing assaults. There are other algorithms and may be different results among different techniques.[4]

A. Rereading this, K. Dutta [4] had proposed a new approach of building the advanced phishing website identification system through the implementation of supervised ML technique called random forest. The process involves an exhaustive analysis and recognition of relevant features which clearly identify phishing websites. The solution is presented as a smart browser extension, equipped with an astonishing 98.8% accuracy in phishing site detection, which helps to overcome human weaknesses in online security. Although it is highly responsible for strengthening the security measures in the internet and providing good protection against future cyber threats, it also causes false alarms sometimes. Cons: The quality of features affect the adaptability to new phishing tactics. Users have no faith in the system if they think there is a possibility for wrong results. [5]

III. MATERIALS AND METHODS

The proposed system employs a novel hybrid ML technique for the phishing attack detection based on URL features. It uses a variety of ML methods to enhance defenses against threats and protect consumers. The use of cross-fold validation and grid search hyper parameter optimization brings a significant improvement of the predicted accuracy. The project extension brings a hybrid model into the picture by adding a Stacking Classifier to take it a notch higher. Here, their ensemble method is a mixture of predictions of two basic classifiers namely, Random Forest [4] Classifier and MLP Classifier. The final prediction is enhanced by LGBM Classifier as a meta-estimator, thus improving the project's classification. This all-or-round approach offers robust, reliable and effective

security against the attacks of Phishing, a significant advancement in the cybersecurity realm.



"Fig.1 Proposed Architecture"

A) Dataset Collection:

The collection is the "URL-based phishing dataset," it provides data for the study and development of systems to detect and discriminate phishing and authentic URLs. The data was collected from Kaggle, a popular data science competition and data set site.

The following is a general description of the data set:

Name: Phishing Data Set based on URL Details:

Source: Kaggle

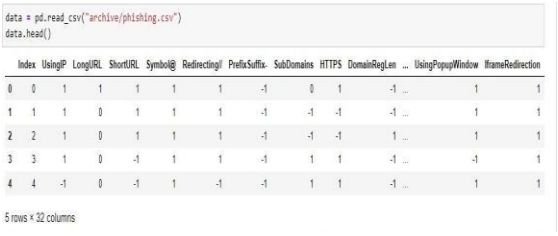
Purpose: T o facilitate phishing detection system research and development.

Size: Includes over 11,000 Web sites.

Format: In a vector form, where each URL will most likely be represented as a set of characteristics or attributes.

The data points (or instances) in the dataset are likely associated with a URL, and the features for each URL are associated with information that can be used by ML models to classify the URL as phishing or genuine.

Typical attributes for a phishing dataset would be things such as URL length, the presence of certain keywords, whether it uses HTTPS, domain age etc. They are significant variables to consider when developing a ML algorithm to detect patterns that distinguish real and phishing URLs.



```
data = pd.read_csv("archive/phishing.csv")
data.head()
```

	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	...	UsingPopupWindow	IframeRedirection
0	0	1	1	1	1	1	-1	-1	0	1	...	1	1
1	1	1	1	0	1	1	1	-1	-1	-1	...	1	1
2	2	1	1	0	1	1	1	-1	-1	-1	...	1	1
3	3	1	1	0	-1	1	1	-1	1	1	-1	...	-1	1
4	4	-1	1	0	-1	1	-1	-1	1	1	-1	...	1	1

5 rows × 32 columns

"Fig.2 Dataset Collection"

B) Processing:

Using Pandas Data frame: The next step is to clean, transform and prepare the data with Pandas, a powerful data manipulation library in Python. This can include handling missing data, data type conversions, and data formatting for further analysis or modeling.

Visualization with Seaborn & Matplotlib: We use Seaborn and Matplotlib to produce visualizations like charts and graphs to get insights on the features of the dataset. In this way we can be able to see the patterns, correlations and distributions of data, and make informed decisions for subsequent analysis.

Label Processing: Here, we use a preprocessing approach called label encoder which encodes the categorical labels into numerical values. Machine learning models are typically trained using numerical data, and it is crucial to ensure that the data is accurate and free from errors. The label processing enables the models to correctly comprehend and learn category information in the data set.

Feature Selection: In this step we find and choose the most relevant features from the dataset. The selection of the most useful variables, as well as the reduction of noise, are important aspects of improving the performance of the model. T echniques such as statistical testing, correlation analysis, or ML algorithms can be used to find the elements that significantly increase the model's predictive potential.

Training & Testing: For our first ML model, we built our first model (Model 9) to assess and better understand the preprocessed data. In the extension phase, we attempted to increase the accuracy of the model predictions by building a hybrid model based on the predictions of various models. The idea behind this new technique is to take advantage of the strengths of varied models to obtain higher overall accuracy in our forecasts. Concurrently, we designed a user-friendly web front-end in Flask, with user authentication features, that allowed the models to be more easily interacted with by users. The frontend is the easy-to-use interface that the user can use to enter data and receive predictions, making it practical and user-friendly. The main part of our project is to train the ML models indicated above on the preprocessed dataset, so that they can recognize complex patterns and relationships in the data. Once trained, rigorous testing is performed with a different test set. These algorithms are carefully tested based on the metrics such as accuracy, precision, recall and F1 score to check the performance of the algorithm in detecting the phishing URLs. This comprehensive evaluation process is crucial in ensuring the accuracy and reliability of the models while also validating their practical application. The research aims to deliver novel and dependable solutions in the area of

phishing URL Identification with this comprehensive technique.

C) Algorithms:

Stacking Classifier: The research is employing an ensemble method that uses the combination of RF [4] Classifier with MLP Classifier as basis classifiers by Stacking Classifier. By employing LGBM Classifier as a meta-estimator of the final prediction, the project's capabilities to make classification is improved.

LSD: The hybrid classification model of Logistic Regression, SVM and DT [4] with Hyperparameter GridCV is a combination of these three methods to enhance the accuracy and efficiency of the model. The organization of searching through hyperparameters combinations is done with GridCV to find the best model performance. It is of great value in numerous classification problems.

Hybrid LSD (Hard): The Hybrid LSD (Hard) model uses LR, SVM and DT [4] algorithms with a hard voting approach for classification judgments. Each base casts his vote and the winning vote is decided by majority. This enhances the accuracy and robustness in several classification problems.

Hybrid LSD (Soft): The Hybrid LSD (Soft) model classifies data using the soft voting of Logistic Regression, Support Vector Machine and Decision Tree [4]. It leverages the strengths of each model for predictions, and can process a variety of data types and boost accuracy for categorisation tasks.

Gradient Boosting: Gradient Boosting is an ensemble ML technique that sequentially builds a prediction model by combining the best properties of many weak learners typically Decision Trees [4]s. It does this by focusing on the errors of the previous models, and adjusting the predictions so as to reduce the errors. This results in a powerful and accurate predictive model with the ability to perform well in a variety of tasks ranging from regression to classification .

Random Forest: RF [4] is an ensemble learning technique that employs many Decision Trees [4] to make a prediction. It does this by using multiple Decision Tree [4]s trained on randomly selected portions of data with their predictions averaged. The ensemble method enhances the accuracy, reduces the overfitting problem and provides good performance on classification and regression tasks.

Decision Tree: Recursively partitioning data into a series of subsets using the most salient feature in order to make judgments to classify or predict results is a Decision Tree [4] ML model. It constructs a tree, with each node representing a feature and each branch representing a potential decision, and it is interpretable and helpful in many applications.

Now that you know how much each outcome costs and the likelihood that it will occur, you can determine the expected value of each outcome, using this formula:

“Expected value (EV) = (First possible outcome x Likelihood of outcome) + (Second possible outcome x Likelihood of outcome) - Cost. (1)”

Support Vector Classifier: A SVC is a ML model that seeks to find the best border (hyperplane) to separate various sets of data, and to maximize the distance between the sets. It is a selection of critical support vectors for good classification. It seems to be very effective for both binary and multi-class classification problems.

The equation of the linear hyperplane can be expressed in the form

“wTx+b=0$wTx+b$=0 (2)”

Where:

- The normal vector to the hyperplane, i.e. orthogonal direction (in this case, the direction of ww).
- The distance of the hyperplane from the origin (along the normal vector ww) is represented by bb, the offset or bias term.

Logistic Regression: LR is a classification algorithm which is used to predict the probability of a particular class for which the input belongs to. Uses a sigmoid function to transform the input features into a probability score ranging from 0 to 1. A threshold is then set and the distribution of probabilities is used to classify the input into 2 or more categories. During training, the model adjusts the coefficients to maximize the likelihood of correct classifications, meeting the best fit to the data.It learns the coefficients during training in order to best fit with the data and to get the correct classifications.

The basic linear regression is the regression line.The regression line in basic linear regression.

“^y=a+bx y ^ = a + b x (3)”

can be described as:

^y is the projected value of y , an is the intercept , and predicts where the regression line will cross the y -axis , b predicts the change in y for each unit change in x .

Naive Bayes: Naive Bayes is a statistical classification method based on the naive Bayes approach. It calculates the probability of a data point being in a class given the probabilities of the different attributes of a data point. Naive Bayes is particularly quick in the text classification, spam detection problems and other scenarios where the features are roughly independent.

In generalized notation we have:

"p(A,B|A) = p(A) * p(B|A) (4)"

This is interpreted as: "probability of A and B if A is known or has occurred is equal to the probability of A times the probability of B if A is known or has occurred". This is referred to as conditional probability or, more accurately, a joint conditional probability, since the occurrence of one event or condition will be considered given the occurrence of another event or condition.

IV. RESULTS AND DISCUSSION

Accuracy: If a test can differentiate people with the disease from healthy people, then it is accurate. The proportion of TP and TN for the total no of cases assessed is determined. This is the test accuracy criteria. Mathematically this is:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} (1)$$

Precision: Precision is the number of true/positive samples classified as positive. The precision is calculated as:

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} (2)$$

Recall: ML recall refers to the accuracy of the model in its ability to recognize all the examples of a given class. It is a measure of the ability of a model to predict positive instances, in terms of the ratio of correctly predicted positive instances to all positive instances.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} (3)$$

F1-Score: F1 score is a measure of accuracy for ML models. Model precision + recall scores. The accuracy statistic quantifies how well a model works over the entire data set.

$$F1\ Score = 2 * \frac{Recall\ \text{X}\ Precision}{Recall + Precision} * 100 (4)$$

Specificity: The ability of the model to correctly detect negative examples. It is given by:

Specificity= TN / (TN + FP). (5)

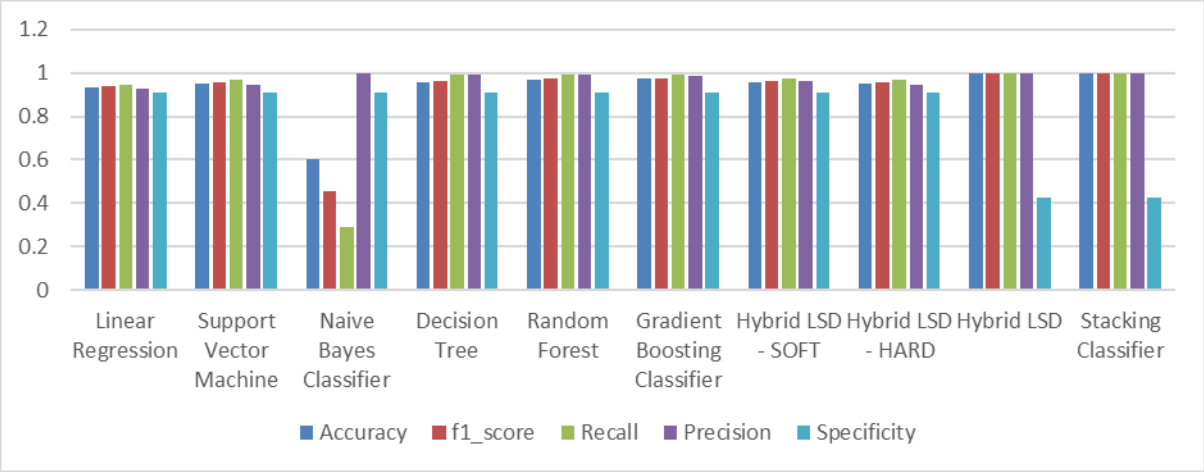
Specificity is the % of negative tests that are correctly identified.

Table (1) shows the performance metrics (Accuracy, precision, recall and F1-score) for each method. The extension stacking classifier is better than other algorithms. The tables also display a comparison of the metrics used for the various methods.

"Table.1 Performance Evaluation Table"

	ML Model	Accuracy	f1_score	Recall	Precision	Specificity
0	Linear Regression	0.934	0.941	0.943	0.927	0.909
1	Support Vector Machine	0.951	0.957	0.969	0.947	0.909
2	Naive Bayes Classifier	0.605	0.454	0.292	0.997	0.909
3	Decision Tree	0.957	0.962	0.991	0.993	0.909
4	Random Forest	0.969	0.972	0.993	0.990	0.909
5	Gradient Boosting Classifier	0.974	0.977	0.994	0.986	0.909
6	Hybrid LSD - SOFT	0.959	0.964	0.977	0.965	0.909
7	Hybrid LSD - HARD	0.950	0.956	0.967	0.945	0.909
8	Hybrid LSD	1.000	1.000	1.000	1.000	0.426
9	Stacking Classifier	1.000	1.000	1.000	1.000	0.426

Graph.1 Comparison Graphs



Accuracy in blue, F1 – Score in red, recall in green and precision in purple, Specificity sky blue Graph (1). Extension stacking classifier outperforms the rest of the models in all metrics and gives the best values as compared to the other models. The results are shown graphically in the above graphs.

V. CONCLUSION

Finally, the research was successful in using a combination of ML methodology in which the characteristics of the URL are given more importance in determining the phishing. Significant gains in accuracy and efficiency were achieved by the system using different models such Decision Tree [4]s, Random Forest [4]s, support vector classifiers and an LSD-based stacking classifier. This remarkable accuracy in classification and F-score, along with an overall improvement in the performance of the phishing detection system, was made possible by the choice of an extension stacking classifier. T his comprehensive strategy provides a robust solution to this significant cybersecurity issue, providing effective protection against serious phishing attacks. The integration of the various ML models added diversity to the capabilities of the system, and also offered a higher degree of flexibility to the new phishing approaches. The success of the project in improving accuracy and efficiency highlights its potential to contribute to strengthening cybersecurity measures, providing a useful contribution to ongoing efforts to battle cyber threats. In the light of the growing sophistication of phishing attacks, the developed system could be seen as a robust protection measure and it demonstrates the potential to safeguard sensitive information and mitigate the threat of phishing attacks in real-world applications.

This project's future goal is to continually improve and adjust to new phishing methods. Future work may study the combination of DL methods, behavioral analytics and real-time threat feeds to boost the system's ability to proactively defend. In addition, working with cybersecurity specialists and industry players can help to create a more comprehensive and robust solution. The easy to use interfaces and its use on cloud-based platforms and IoT devices will add to its usability. As the threat landscape evolves, the model will be continually updated, ensuring its relevance and effectiveness, and making it a cutting-edge solution in the dynamic field of cybersecurity.

REFERENCES

[1] Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, Y. Si, F. Zhang, and J. S. Dong, ''Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages,'' in Proc. 30th USENIX Secur. Symp. (USENIX Security), 2021, pp. 3793–3810.
[2] H. Shirazia, K. Haynesb, and I. Raya, ''Towards performance of NLP transformers on URL-based phishing detection for mobile devices,'' Int. Assoc. Sharing Knowl. Sustainability (IASKS), Tech. Rep., 2022.
[3] A. Akanchha, ''Exploring a robust machine learning classifier for detecting phishing domains using SSL certificates,'' Fac. Comput. Sci., Dalhousie Univ., Halifax, NS, Canada, Tech. Rep. 10222/78875, 2020.
[4] H. Shahriar and S. Nimmagadda, ''Network intrusion detection for TCP/IP packets with machine learning techniques,'' in Machine Intelligence and Big Data Analytics for Cybersecurity Applications. Cham, Switzerland: Springer, 2020, pp. 231–247.
[5] A. K. Dutta, ''Detecting phishing websites using machine learning technique,'' PLoS ONE, vol. 16, no. 10, Oct. 2021, Art. no. e0258361.
[6] A. K. Murthy and Suresha, ''XML URL classification based on their semantic structure orientation for web mining applications,'' Proc. Comput. Sci., vol. 46, pp. 143–150, Jan. 2015.
[7] A. A. Ubing, S. Kamilia, A. Abdullah, N. Jhanjhi, and M. Supramaniam, ''Phishing website detection: An improved accuracy through feature selection and ensemble learning,'' Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 252–257, 2019.
[8] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, ''PhishAri: Automatic realtime phishing detection on Twitter,'' in Proc. eCrime Res. Summit, Oct. 2012, pp. 1–12.
[9] S. N. Foley, D. Gollmann, and E. Snekkenes, Computer Security— ESORICS 2017, vol. 10492. Oslo, Norway: Springer, Sep. 2017.
[10] P. George and P. Vinod, ''Composite email features for spam identification,'' in Cyber Security. Singapore: Springer, 2018, pp. 281–289.
[11] H. S. Hota, A. K. Shrivas, and R. Hota, ''An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique,'' Proc. Comput. Sci., vol. 132, pp. 900–907, Jan. 2018.

[12] G. Sonowal and K. S. Kuppusamy, ''PhiDMA—A phishing detection model with multi-filter approach,'' J. King Saud Univ., Comput. Inf. Sci., vol. 32, no. 1, pp. 99–112, Jan. 2020.
[13] M. Zouina and B. Outtaj, ''A novel lightweight URL phishing detection system using SVM and similarity index,'' Hum.-Centric Comput. Inf. Sci., vol. 7, no. 1, p. 17, Jun. 2017.
[14] R. Ø. Skotnes, ''Management commitment and awareness creation—ICT safety and security in electric power supply network companies,'' Inf. Comput. Secur., vol. 23, no. 3, pp. 302–316, Jul. 2015.
[15] R. Prasad and V. Rohokale, ''Cyber threats and attack overview,'' in Cyber Security: The Lifeline of Information and Communication Technology. Cham, Switzerland: Springer, 2020, pp. 15–31.
[16] T. Nathezhtha, D. Sangeetha, and V. Vaidehi, ''WC-PAD: Web crawling based phishing attack detection,'' in Proc. Int. Carnahan Conf. Secur. Technol. (ICCST), Oct. 2019, pp. 1–6.
[17] R. Jenni and S. Shankar, ''Review of various methods for phishing detection,'' EAI Endorsed Trans. Energy Web, vol. 5, no. 20, Sep. 2018, Art. no. 155746.
[18] (2020). Accessed: Jan. 2020. [Online]. Available: https://catches-of-themonth-phishing-scams-for-january-2020
[19] S. Bell and P. Komisarczuk, ''An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank,'' in Proc. Australas. Comput. Sci. Week Multiconf. (ACSW), Melbourne, VIC, Australia. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–11, Art. no. 3, doi: 10.1145/3373017.3373020.
[20] A. K. Jain and B. Gupta, ''PHISH-SAFE: URL features-based phishing detection system using machine learning,'' in Cyber Security. Switzerland: Springer, 2018, pp. 467–474.
[21] Y. Cao, W. Han, and Y. Le, ''Anti-phishing based on automated individual white-list,'' in Proc. 4th ACM Workshop Digit. Identity Manage., Oct. 2008, pp. 51–60.
[22] G. Diksha and J. A. Kumar, ''Mobile phishing attacks and defence mechanisms: State of art and open research challenges,'' Comput. Secur., vol. 73, pp. 519–544, Mar. 2018.
[23] M. Khonji, Y. Iraqi, and A. Jones, ''Phishing detection: A literature survey,'' IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2091–2121, 4th Quart, 2013.
[24] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, ''Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions,'' in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., Apr. 2010, pp. 373–382.
[25] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, ''PhishNet: Predictive blacklisting to detect phishing attacks,'' in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–5.
[26] P. K. Sandhu and S. Singla, ''Google safe browsing-web security,'' in Proc. IJCSET, vol. 5, 2015, pp. 283–287.
[27] M. Sharifi and S. H. Siadati, ''A phishing sites blacklist generator,'' in Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl., Mar. 2008, pp. 840–843.
[28] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, ''An empirical analysis of phishing blacklists,'' in Proc. 6th Conf. Email Anti-Spam (CEAS), Mountain View, CA, USA. Pittsburgh, PA, USA: Carnegie Mellon Univ., Engineering and Public Policy, Jul. 2009.
[29] Y. Zhang, J. I. Hong, and L. F. Cranor, ''Cantina: A content-based approach to detecting phishing web sites,'' in Proc. 16th Int. Conf. World Wide Web, May 2007, pp. 639–648.
[30] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, ''CANTINA+: A featurerich machine learning framework for detecting phishing web sites,'' ACM Trans. Inf. Syst. Secur., vol. 14, no. 2, pp. 1–28, Sep. 2011