2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT) | 978-1-6654-7224-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCiCT56684.2022.00062

Application of ensemble Machine Learning models for phishing detection on web networks

[1]Navyah Puri
Department of Information Technology
Guru Tegh Bahadur Institute of Technology
Delhi, India
purinavyah08@gmail.com

[2]Pranay Saggar
Department of Information Technology
Guru Tegh Bahadur Institute of Technology
Delhi, India
pranaysaggar@gmail.com

[3]Dr. Amandeep Kaur
Department of Information Technology
Guru Tegh Bahadur Institute of Technology
Delhi, India
amandeep.gtbit@gmail.com

[4]Puneet Garg
ABES Engineering College
Ghaziabad, U. P.
India
puneetgarg.er@gmail.com

***Abstract*–Phishing is a technique of fraud and identity stealing that includes convincing Website visitors to provide confidential info and details such as their user id, secret key, payment info, and so on. It is one of the real safety concerns that the online revolution is worried about, and it may cost businesses and users money. The use of SHAP values to better comprehend the model employed in phishing URL detection is the research's highlight. This research examines multiple machine learning models for detecting phishing by examining various aspects of the website's URL. The dataset that was used to train the model is open source, consisting of datasets from Alexa, UCI, Phishtank, and Kaggle. There are 11,055 rows and 32 columns in the data set. The data were normalized using the SMOTE analysis technique, which resulted in a larger data set. This data was then fed into a variety of classification and ensemble models (K-means, Random forest, decision tree, CatBoost classifier, LightGBM classifier, AdaBoost, and voting classifier). The Accuracy and F1 values of the models were compared. The model's accuracy was tested before and after using smote. After putting all of the strategies to the test, we observed that CatBoost Classifier produced the best results for accuracy and F1 value. To conclude, SHAP values are a crucial part in model interpretation and are utilized to identify important features in the model and how they impact the output of the model. This model can be used by authorities and companies to stop phishing attacks and identify suspicious sites before someone is harmed by them.**

Index Terms–Shapley Additive Explanations (SHAP), SMOTE, Machine Learning, CatBoost, Phishing, Web Networks

I. INTRODUCTION

The administrations of various associations provide a platform for data dissemination, collaboration, but they also provide entry points for hostile clients [1]. Phishing attacks are unavoidable, and they target associations at all times and in all places. Phishers are progressing towards developing new advanced phishing techniques in web networks by employing social design concepts and innovation in their attacks [2]. Blacklist-based tactics[3] and heuristic-based strategies are widely used to detect evolving phishing pages regularly. Blacklist-based procedures looked at the recent files inside the blacklist to detect such attempts, but they couldn't handle the newer ones[4]. The detection should be quite precise. When the exactness of URLs is an issue, the visit repetition of URLs should also be taken into account [5]. Blacklists are a collection of URLs which were already determined as harmful [6]. Assailants use ingenious methods to avoid blacklists and deceive clients, such as changing the URL to make it "appear " legal using obscurity. Phisher provides instructional pages that give users expensive information. Links to Facebook, Gmail, and Twitter are also available on those websites [7]. For the most part, phishing attempts, whether through emails or other means of media, the goal is to get the victim to tap on a domain name that

978-1-6654-7224-1/22/$31.00 ©2022 IEEE
DOI 10.1109/CCiCT56684.2022.00062

looks to lead to a trustworthy page, but it isn't. The most basic step in manipulating links is to construct a malicious link that points people to the malicious page that the guest is looking for the assailant wants [8].

The primary goal of this study will be to investigate and analyze several ML algorithms such as K-means, decision tree, CatBoost classifier, LightGBM classifier, Random forest, AdaBoost, and voting classifier as well as SMOTE analysis and SHAP values for detecting phishing URLs to protect users from scams in web networks that can be financially and psychologically devastating.

The task of detecting a phishing assault is difficult. This attack can be clever enough to trick even the most knowledgeable guest, like replacing some values in the domain with similar values. phishing can occur because of carelessness, for example, instead of a DNS server, a Port number is used, as a disadvantage [9]. Given that a single website might be blocked across multiple browsers, such a circumstance can cause as much trouble for the parties as a successful fraud [10].

This paper's important part is an unique approach for dealing with vagueness in e-phishing web page evaluation, as well as a clever, adaptable, and effective model for identifying phishing websites. SMOTE was used, a data augmentation method, to improve our classification predictions. The primary idea behind SMOTE is to create synthetic data along the line that separates minority cases from their nearest neighbors. To gain better predictive ability, ML algorithms such as Random Forest, Decision Tree, Ensemble models, XGBoost, CatBoost, Adaboost, were integrated. Hyperparameter tweaking was utilized to determine the best collection of hyperparameters for the algorithm, which was highly valuable as the model accuracy and prediction had altered dramatically.

SHAP Values were chosen to show the Blackbox algos in the Model interpretation because, while having an output that is difficult or impossible to understand, these algorithms may be characterized as procedures with an undefined outcome. In this case, it means that you get an output from input but are ignorant of why. These Values show how all the factors influence the prediction. We define complex algos like neural networks, gradient boosting and more using a breakdown of SHAP values, and we also try to get a deeper grasp of how the model makes choices. Models just look at the "how much" part of the issue and ignore the rest. The SHAP framework has significantly aided machine learning model interpretation. Scott Lundberg and Su-In Lee, the creators of SHAP, created a simple, theoretically sound method to comprehend predictions for any model. The SHAP model describes the effect of having a particular value for a specific feature against the prediction.

II. LITERATURE REVIEW

Phishing can be detected using a variety of methods of detection, including ML based, heuristic based, list based and deep learning based. Because the phishing problem is so complex, there is no single solution that can effectively counteract all threats; as a result, numerous strategies are frequently used to combat specific attacks[11].

Thaker suggested a method that would detect both old and new phishing URLs that have no prior practice to make a judgment after using Data Mining. [12] created a cloud-based classification model for the equivalent, in which different distinct characteristics will be used as input data through the URL. The non-technical approach offers no defense against phishing websites' ability to disappear quickly. Using the WEKA equipment, the C4.5 (J48) data mining method was completed. [13] presented C4.5, a benchmark data mining technique that can accurately identify phishing websites.

A training dataset of 750 URLs was created to train the computation J48, which is a WEKA implementation of C4.5 algorithms. A closeness coordinating system is used to employ detection rates. New URLs may be detected by attackers' regular URL control techniques, according to [14]. This technique covers a large number of harmful URLs with limited features. Li et al. [15] developed a stacking model to detect phishing websites using URL and HTML data.

In recent articles, search engines have been utilized to detect phishing web pages. Garera., Whittaker., and Zhang have all employed the Google search engine. [16][17][18] Various methods are now being used to detect phishing websites. Maher Aburous et al. offer a fuzzy data mining method for intelligent phishing detection. The detection rate of e-banking phishing websites is calculated using six criteria in Hossain et al.[19]: URL and Domain Identity, Security & Encryption, Source Code & JavaScript, Page Style & Contents, Web Address Bar, and Social Human Factor. E-banking phishing websites are classified using fuzzy logic and data mining techniques. For identifying phishing attempts, Ram Basnet et al.[20] use a machine learning method. For the efficient prediction of phishing emails, support vector machines, biased support vector machines, and neural networks are utilized. The primary goal of this article is to identify phishing emails by integrating important structural characteristics into the emails and using various machine learning techniques to classify them.

In Begum & Badugu [21], the authors explored several methods for detecting phishing attacks. For phishing attack detection, they conducted a thorough review of current methods such as Machine Learning (ML)-based approaches, Non-machine Learning (NML)-based approaches, Neural Network-based approaches, and Behavior-based detection approaches. Yasin et al. [22] compiled numerous papers that academics have utilized to explain various social specialized activities. Furthermore, they suggested that using topical and game-based research methods, a better understanding of social engineering assault scenarios might be achieved. One such effort to enable individuals to understand broad attack situations is the suggested method for analyzing social engineering assault scenarios. The theoretically predictable system of this approach warrants future enhancement and re-performance.

PhishI was proposed by Fatima et al. [23] as a precise method to deal with structuring real games for security training. They utilized stick phishing as a model to demonstrate how the suggested method works, and then evaluated the game's learning effects based on observational data collected from the students' movements. Members of the PhishI game are required to exchange phishing messages and have the opportunity to comment on the attack scenario's feasibility. The results showed that students' awareness of spear-phishing risks has increased, and their defense against the first possible assault has been enhanced. Furthermore, the game had a positive impact on participants' understanding of excessive internet data and information sharing.

III. MATERIAL AND METHODS

In this part, we show how to employ ensemble methodology to enhance a CatBoost ensemble algorithm, as well as ways for explaining the model. The whole technique followed in our study is depicted in Figure 1.

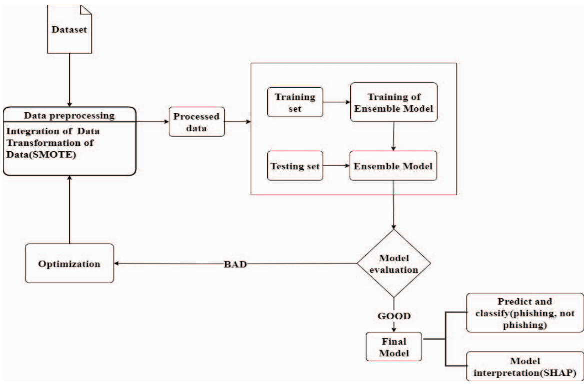
Dataset
Data preprocessing
Integration of Data Transformation of Data(SMOTE)
Processed data
Training set
Training of Ensemble Model
Testing set
Ensemble Model
Optimization
BAD
Model evaluation
GOOD
Final Model
Predict and classify(phishing, not phishing)
Model interpretation(SHAP)

Fig. 1 Ensemble-based Model for phishing site detection based on domain/URL of a website

A. Data Preparation

It is the process of converting acquired data into a useful format; in our case, category based as well as number based data are used. As a result, category based data is firstly changed to numbers. This was followed by a data augmentation or oversampling stage, which expanded the data by around 12%. We used SMOTE for oversampling. The oversampling methodology is used because random forest and decision tree methods, as well as other ML models, have a biased structure that ignores the minority class. As a result, slight mistakes in forecasting the majority class occur, and the minority class is misclassified in comparison to the dominant class. In layman's terms, a skewed dataset with a strong majority class makes our model more vulnerable to instances when the minority class has poor or no memory.

- SMOTE(Synthetic Minority Oversampling Technique): SMOTE is one of the most often utilised oversampling approaches for resolving the balancing issue. Its goal is to establish a more fair allocation of classes by re-creating minority classes at random. SMOTE brings together existing minorities to create new ones. It uses linear interpolation to construct virtual training records for the minority class. For each example in the minority class, these synthetic training records are constructed by picking one or more of the k-nearest neighbours at random. The database is reconstructed following an oversampling process, and several classification models may be applied to the changed data.

B. Model Construction

The machine learning job of inferring a function from supervised training data is known as supervised learning. A collection of training examples makes up the training data. Each example in supervised learning is made up of an input object and a supervisory signal, which is the desired output value. A supervised learning algorithm examines the training data and generates a classifier, which is an inferred function. After that, the classifier is utilized to predict the precise output value for each valid but unseen input item. Kmeans, Random Forest, Decision Tree, CatBoost, XGBoost, LGBMClassifier, AdaBoost, and Voting Classifier are the eight classification algorithms used to learn the website data:

- **K-Neighbors Classifier (KNC)**: The K-nearest neighbors (K-NN) technique is a supervised machine learning algorithm which is utilized to address both regression, classification problems. KNC is one of the most basic machine learning techniques for classifying a set of features into the most often occurring class among the dataset's k-nearest neighbor.

- **Random Forest Classifier (RFC)**: It is an ensemble tree-based learning technique that consists of a succession of decision trees taken from a portion within the training examples at randomness. The final classification of the items is determined by the votes of various decision trees collected by random forest classifiers. Classification as well as Regression problems both can be solved using this algorithm.
- **Decision Tree Classifier:** It is categorized as a Supervised Learning algorithm. Both regression, classification issues can be solved using decision tree techniques. It solves problems using tree representation, in which each leaf node defines the class tag and the characteristics are found on the tree's internal node. Boolean functions on various attributes are also represented using decision trees. The benefit of this technique is that it is simple to implement, explain, and illustrate, while the downside is that it requires a step-by-step analysis of the transactions.
- **CatboostClassifier(CBC):** Catboost stands for Categorical Boosting as it deals with categorical data. It is an algorithm used for gradient boosting on Decision Trees. It works well with a variety of data types, including audio, text, and images, as well as historical data. Since we used heterogeneous data, Catboost is the best classification method to apply because in the initial run it outperforms the majority of boosting algorithms.
- **LightGBM Classifier (LGBM)**: It stands for lightweight gradient boosting machine. It is yet another gradient boosting classifier that uses tree-based learning techniques. It is mainly designed to give more efficient and accurate results. Also, it has a faster training speed and uses lower memory space.
- **AdaBoost Classifier (ABC):** AdaBoost stands for Adaptive Boosting Classifier. It was introduced as an ensemble enhabcing classifier in the mid 1990's. The main function of Adaboost is that it combines many classifiers to enhance algorithms. Also, it is a way for generating iterative ensembles. It trains the algorithm iteratively on varied weighted training instances. It seeks to produce a decent match to these instances in each iteration by minimizing training errors.
- **Voting Classifier (VC):**The voting classifier is employed in the ensemble model to categorise the data into multiple classes. Voting is a mechanism for combining the findings of many classifiers, not a classifier in and of itself. A voting classifier's container is made up of numerous trained classes that give the desired category to each data determined by the number of voters. To build the machine and ensemble it to obtain the correct output, the identical datasets are provided to all potential categories. For multi-class situations, the ensemble voting classifier is the best option.

C. Dataset description

Several high-quality datasets may be found on a variety of trustworthy websites. The websites Alexa, Phishtank, UCI, and Kaggle are well-known providers for a variety of interesting datasets. The Kaggle websites will be utilized for testing reasons in this project. The database has 31 characteristics. The last property is " Result," which indicates whether or not a phishing website exists.

D. Hyperparameter Tuning

Machine learning methods often need the configuration of a few dozen hyper-parameters before model training. Particularly for Catboost, ADABoost, or LightGBM, which contain a large number of hyper-parameters, hyper-parameter settings have a considerable effect on model performance. Optimizing a mapping function over a configuration space, which defines the hyper-parameter values to be investigated for each hyper-parameter, is the foundation for hyper-parameter tuning.

E. Model Interpretation and Discussion

Studies show that the novel SHAP value estimate technique is more aligned with human perception and effectively discriminates across model output classes than many current methods. This approach necessitates retraining the model on all feature subsets $S\subseteq F$, where S is the set of all features and F is the set of all features. It gives a significance rating to each feature that reflects the impact of adding that information on model prediction. To calculate this effect, a model $f_{S\cup\{i\}}$ is trained with that feature present, whereas a model f_S is trained without it. Then, given the current input $f_{S\cup\{i\}}(x_{s\cup\{i\}})$- $f_S(x_S)$, where x_S represents the values of the input features in the set S, the predictions from the two models are compared. The previous differences are calculated for all feasible subsets $S\subseteq F\setminus\{i\}$ since the impact of withholding a feature is dependent on other characteristics in the model. Following that, the Shapley values are calculated and utilized as feature attributions.

IV. RESULTS

The implementation results for our dataset are discussed in this section. To build our detection model for falsified URLs, we initially used three groups of ML algorithms namely Decision Tree, Kneighbors classifier, Random Forest and in ensemble learning, we used 4 algorithms: CatBoost, LGBM AdaBoost, XGBoost. Their performances are compared using the formula for Accuracy in which is shown in Table 1, and F1 score as shown in Table 2. The accuracy and F1 score are graphically represented in figure 2 and figure 3 After a round of optimization, we discovered that the ensemble approach CatBoost produces the best results. We also see that when we validate and test existing approaches, we always get the lowest result. The boosting and bagging method outperforms other standard standalone methods, increasing accuracy from 93.83 percent to 97.88 percent.

The accuracy of several machine learning models is listed in the table below. We tested different models with SMOTE Analysis and found that the accuracy of the models without SMOTE Analysis was lower than the accuracy of the models with SMOTE Analysis. CatBoost Classifier has the highest accuracy, with a score of 98.61 percent.

TABLE 1
With and without SMOTE, the accuracy of various models

Classifier	**Accuracy without SMOTE**	**Accuracy with SMOTE**
Random Forest	97.88%	97.88%
K-means	95.29%	95.96%
CatBoost	97.51%	98.61%
Decision Tree	97.48%	98.53%
LGBMClassifier	95.93%	96.87%
XGBoost	97.24%	97.44%
Voting Classifier	91.77%	93.91%
AdaBoost	93.83%	93.96%

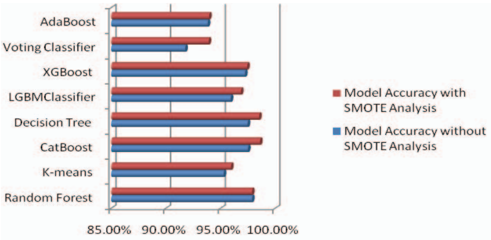

Fig.2 Graphical representation of accuracies with and without SMOTE analysis of different models

The F1 score provides additional details on class accuracy as well as precision, and recall effectiveness. As seen in Table 1, the ensemble models were exceptionally accurate in classifying phishing and legitimate URLs. In the example at hand, if successful classification of a URL class was desired for any acceptable reason, the F1 will aid in establishing differences between the categories and determining which predictor is the most successful. Individual classifiers that augment and bag outperform traditional classifiers.. The maximum F1-Score was observed for the CatBoost Classifier and Random Forest Classifier with a value of 0.98 for both class 0 and class 1.

TABLE 2
Models' F1 scores for two different classes

Classifier	**F1 Score of Class 0 (Phishing)**	**F1 Score of Class 1 (legitimate)**
Random Forest	98%	98%
K-mean	95%	95%
CatBoost	98%	98%
Decision Tree	97%	97%
LGBM Classifier	98%	97%
XGBoost	98%	97%
Voting Classifier	94%	94%
AdaBoost	94%	94%

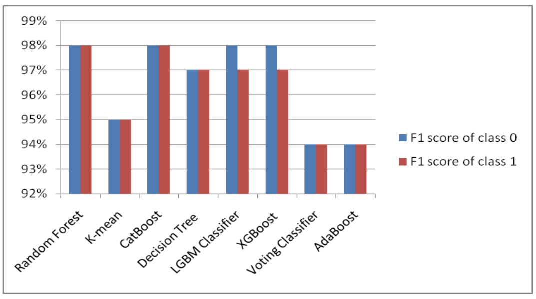

Fig. 3 comparison of F1 scores of different classes of different models

The results have been visually represented using count plot and heat map in figure 4 and figure 5.

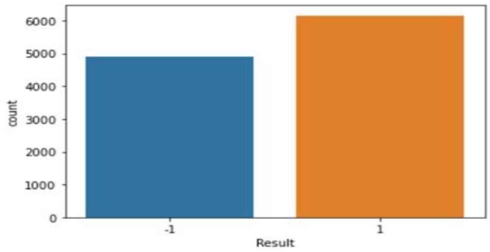

Fig. 4 Countplot

Heatmap - A heatmap is a visual representation of a matrix plot. The data for a heatmap should be in a matrix format. By matrix, we mean that the index and column names must be similar in some way in order for the data we enter into the cells to be relevant.

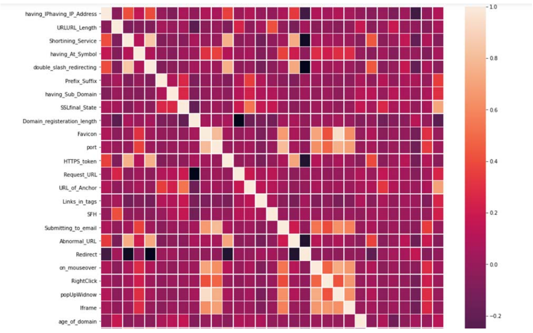

Fig. 5(a) Heatmap

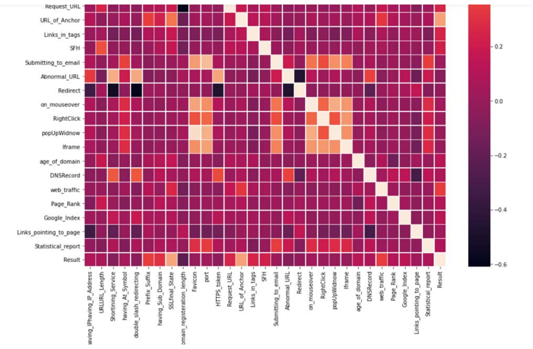

Fig. 5(b) Heatmap

A. Interpretation of Model

We used the Tree SHAP technique [24], which has shown to be a strong tool for reliably understanding ensemble models, to better interpret our optimum helpfulness model implemented in CatBoost. Tree SHAP estimates the contribution of each parameter to the predicted value (Tree SHAP values) for each person in the training dataset. The global feature contributions are then sorted across all samples using the mean (|Tree SHAP|). Figure 6 depicts the global feature contributions obtained from theCatBoost helpfulness model.

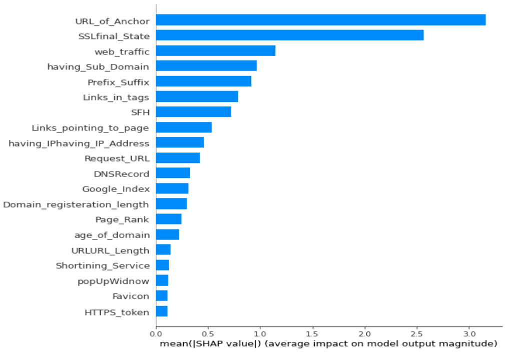

Fig. 6 Global feature attribution across the phishing dataset

When a component is eliminated from the model, the x-axis in Figure 6 depicts the mean degree shift in predicted value. The characteristics are ranked according to the absolute sum of their model effect magnitudes. It was initially deduced that feature contribution varied between categories, with some characteristics contributing much more than others. For example, URL_of_Anchor outperformed the other characteristics in the Phishing dataset. The <a> tag defines an anchor as an element. This option is taken care of in the similar method as "Request URL." However, for this feature, we will look at: If the domain names of the <a> tags and the webpage are different. If the per cent of URL_Of_Anchor is more than 67 per cent, the site is very likely to be a phishing site, according to the results. Another result is that certain extracted features, such as HTTPS_Token, Favicon, PopUpWindow, Shortening_Services, and URLURL_Length, contributed little or nothing to the model outputs. As a result, a re-evaluation of the model's performance is required to determine if such low contribution or no contribution characteristics should be excluded.

Because Tree SHAP values are generated using a customized model interpretation method, the model may provide an individualized interpretation for each sample. Figure 4 depicts how the impact of a particular characteristic influences its participation in the predicted value. The x location of the dot represents the influence of the feature on the review helpfulness, and the shade of the dot indicates the cost of that characteristic for said review.

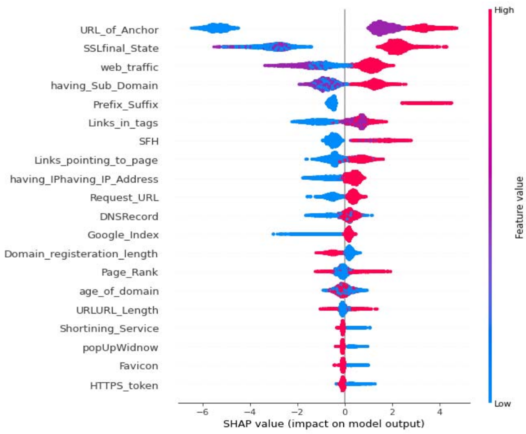

Fig. 7 Individual feature contributions(Summary plot)

The importance of characteristics is shown in the summary plot by putting features in decreasing order. The summary plot also shows the feature's effect in terms of color, with pink suggesting a high impact and blue indicating a low impact, as well as the feature's positive and negative relationship with the objective variable. As displayed on the X-axis. As a consequence, global interpretability is a problem for summary charts. As shown in the summary plot in figure 7, URL_of_Anchor is the most important feature for determining whether a site is phishing or not. The SSLfinal_State is the second most important phishing site determinant. The summary graphic demonstrates which factors are most important. And if it is related to the objective variable in a good or negative way. In this research, the contribution of a feature to a single record prediction is utilized to calculate its shap value. These visuals may aid authorities in comprehending the main influences on phishing sites.

V. MANAGERIAL AND SOCIAL IMPLICATIONS

The study has a wide range of applications:

- In the cyber cell, the model may assist the authorities in identifying phishing websites and blocking them before they can do any damage to the user.
- In the long term, an all-encompassing phishing assault detection mechanism might be built to discover, notify, and prohibit harmful website pages even without user's involvement. Financial loss, property rights theft, brand harm, and disruption of operational processes are just a few of the adverse effects of phishing.
- Malware websites appear to somehow be real, but fraudsters imitate the look and functionality of legal websites, making them hard to spot. Anti-phishing frameworks or third - party add are necessary to avoid fraud. Furthermore, these plug-ins or systems may do content filtering as well as prevent possible malicious urls.

VI. CONCLUSION AND FUTURE SCOPE

This research presents phishing URL prediction as a classification problem, it also shows how a ML based method can be used to predict if a given website is genuine or fraudulent. The prediction model was trained using K-means, Random Forest, Decision Tree, CatBoost, XGBoost, LGBMClassifier, AdaBoost, and Voting Classifier. For phishing and genuine websites, features were collected from the websites Alexa, Phishtank, UCI, and Kaggle, and the dataset was balanced using the SMOTE method, after which the final training dataset was produced to ease training and deployment. The models' performance was assessed using the F1 score and predicted accuracy. According to the findings, the CatBoost classifier outperforms the other models, with a state-of-the-art accuracy of 98.61%. Finally, the Shaply value estimate method is utilized to identify the characteristics influencing the model to properly interpret the best model. As the number of research on such subjects grows, the dataset's size may be expanded in the future. To improve forecast accuracy, a variety of different models and algorithms may be used. To properly understand the model, additional characteristics linked to Shaply values may be utilized.

For the model to be interpretable, the SHAP value technique needs to be run through "all possible combinations" of parameters. When there are a lot of features, there are a lot of potential combinations, which means a lot of SHAP value calculation and a lot of temporal complexity. It turns out that this is computationally impossible. To address the challenges of balancing the difficulty and understandability of Phishing Detection methods, as well as to improve model precision and openness, we may use a combination of ML models and SHAP values. The study is expected to provide new theoretical material to expand and deepen the ML library, as well as give important models and observations to enhance education throughout the world.

REFERENCES

[1] Abutair, H.Y., Belghith, A.: A multi-agent case-based reasoning architecture for phishing detection. Proced. Comput. Sci. 110, 492–497 (2017)

[2] Aburrous, M., Hossain, M.A., Dahal, K., Thabtah, F.: Intelligent phishing detection system for e-banking using fuzzy data mining. Expert Syst. Appl. 37(12), 7913–7921 (2010)

[3] Inuwa-Dutse, I., Liptrott, M., Korkontzelos, I.: Detection of spamposting accounts on Twitter. Neurocomputing 315, 496–511 (2018)

[4] Huang, D., Xu, K., Pei, J.: Malicious URL detection by dynamically mining patterns without predefined elements. World Wide Web 17(6), 1375–1394 (2014)

[5] Sahingoz, O.K., Baykal, S.I., Bulut, D., Phishing detection from urls by using neural networks

[6] Makawana, P.R., Jhaveri, R.H.: A Bibliometric analysis of recent research on machine learning for cyber security. Intelligent Communication and Computational Technologies, pp. 213–226. Springer, Singapore (2018)

[7] Basnet, R.B., Sung, A.H., Mining web to detect phishing URLs. In: 2012 11th International Conference on Machine Learning and Applications, vol. 1. IEEE, pp. 568–573 (2012)

[8] Parekh, S., Parikh, D., Kotak, S., &Sankhe, S. (2018, April). A new method for detection of phishing websites: URL detection. In 2018 Second international conference on inventive communication and computational technologies (ICICCT) (pp. 949-952). IEEE. doi:10.1109/ICICCT.2018.8473085

[9] Zouina, M., &Outtaj, B. (2017). A novel lightweight URL phishing detection system using SVM and similarity index. Human-centric Computing and Information Sciences, 7(1), 1-13. doi:10.1186/s13673-017-0098-1

[10] da Silva, C. M. R., Feitosa, E. L., & Garcia, V. C. (2020). Heuristic-based strategy for Phishing prediction: A survey of URL-based approach. Computers & Security, 88, 101613. doi:10.1016/j.cose.2019.101613

[11] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., &Niyigena, J. P. (2020). An effective phishing detection model based on character level convolutional neural network from URL. Electronics, 9(9), 1514. doi:10.3390/electronics9091514

[12] Thaker, M., Parikh, M., Shetty, P., Neogi, V., Jaswal, S.. Detecting phishing websites using data mining. In: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, pp. 1876–1879 (2018).

[13] Priya, A., Meenakshi, E.. Detection of phishing websites using C4. 5 data mining algorithm. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information amd Communication Technology (RTEICT). IEEE, pp. 1468–1472 (2017)

[14] Kim, S., Kim, J., Kang, B.B.: Malicious URL protection based on attackers' habitual behavioral analysis. Comput. Secur. 77, 790–806 (2018)

[15] Li, Y., Yang, Z., Chen, X., Yuan, H., Liu, W.: A stacking model using URL and HTML features for phishing webpage detection. Future Gener. Comput. Syst. 94, 27–39 (2019)

[16] Garera, S., Provos, N., Chew, M., & Rubin, A. D. (2007, November). A framework for detection and measurement of phishing attacks. In Proceedings of the 2007 ACM workshop on Recurring malcode (pp. 1-8). Doi: 10.1145/1314389.1314391

[17] Whittaker, C., Ryner, B., &Nazif, M. (2010). Large-scale automatic classification of phishing pages.

[18] Zhang, Y., Hong, J. I., &Cranor, L. F. (2007, May). Cantina: a content-based approach to detecting phishing web sites. In Proceedings of the 16th international conference on World Wide Web (pp. 639-648). Doi: 10.1145/1242572.1242659

[19] Hossain M.A, Keshav Dahal, Maher Aburrous, "Modelling Intelligent Phishing Detection System for e-Banking using Fuzzy Data Mining".

[20] Andrew H.Sung, Ram Basenet, Srinivas Mukkamala, "Detection of Phishing Attacks: A machine Learning Approach".

[21] Begum, A., & Badugu, S. (2020). A study of malicious url detection using machine learning and heuristic approaches. In Advances in decision sciences, security and computer vision, image processing (pp. 587–597). Berlin: Springer.

[22] Yasin, A., Fatima, R., Liu, L., Yasin, A., & Wang, J. (2019). Contemplating social engineering studies and attack scenarios: A review study. Security and Privacy, 2(4), e73.

[23]. Fatima, R., Yasin, A., Liu, L., & Wang, J. (2019). How persuasive is a phishing email? A phishing game for phishing awareness. Journal of Computer Security, 27(6), 581–612.

[24] Lundberg, S.M.; Erion, G.G.; Lee, S.-I. Consistent individualized feature attribution for tree ensembles. In Proceedings of the 34th International Conference onMachine Learning, Sydney, Australia, 6–11 August 2017; pp. 1–9