Phishing Web Page Detection using Web Scraping

Mallika Boyapati[1], Ramazan Aygun[1,2]
[1]*School of Data Science and Analytics,* [2]*Department of Computer Science*
Kennesaw State University
Kennesaw, GA, USA
{mboyapat@students,raygun}.kennesaw.edu

SoutheastCon 2023 | 978-1-6654-7611-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SoutheastCon51012.2023.10115148

***Abstract*—In today's Internet era, webpages act as major user interfaces for the applications hosted by the organizations. One of the most common security attacks on the web page applications is phishing attacks. Phishing is a social engineering attack in which an adversary tries to steal user credentials by tricking them to believe that they are on a legitimate web page. Adversaries are using sophisticated and new ways to forge the web page designs craftily to trick the users into visiting the malicious links. The phishing webpages are used as a medium to carry out the art of phishing attacks. Web scraping is a methodology to extract features of each webpage. In this paper, web scraping is employed to extract hybrid feature set to implement Machine Learning (ML) models. The machine learning models like XGBoost, Multilayer Perceptron, Logistic Regression, SVM, Auto Encoder, Random Forest, Decision Trees, K-means, and Naive Bayes are built. All the models are tested with and without applying principal component analysis (PCA), a feature reduction technique. Extracting hybrid features by employing web scraping to train the ML models and finding the key features contributing to phishing detection on Phishpedia, Kaggle, and PhishTank datasets is the key contribution of this paper. Results show that XGBoost algorithm outperformed the all other classifiers with 98% accuracy or higher on the web scraped features on all the datasets. The model achieved higher precision and recall when compared to other approaches like CANTINA+, URL based approaches, SenseInput, Knowing Thy Domain, and Phishpedia on the Phishpedia dataset.**

***Index Terms*—Phishing webpage detection, machine learning models, web scraping, hybrid feature extraction.**

I. Introduction

In recent years, billions of dollars are transferred to fraudulent accounts by an hacker after luring the user for the credentials. IC3 recent survey states that US businesses have suffered an adjusted loss of over 54 million dollars only through phishing attacks[1]. Availability of personal information through online social networks enable attackers develop sophisticated phishing attacks [1]. The threats are growing every day as hackers steal users' personal information in many ways [2]. One way is via webpages as a medium launched as phishing attacks. The behavior and reactions of users Any wireless device connected to the Internet through a web browser capable of opening web pages or web applications is prone to web page phishing attacks. The adversary or hacker will be able to host a web page similar to real sites from any server without even getting noticed by the legitimate owners. Although the adversary tries to replicate the web page applications as legitimate, there are a lot of differences that are being observed between legitimate and phishing web pages. By observing these differences, we were able to come up with some solutions that can detect phishing web pages using machine learning models by performing web scraping to extract the features of webpages.

In recent years, many research studies use visual approaches to detect phishing webpages. These approaches use logo and webpage screenshot as the basis for the detection techniques [3]–[6]. Some studies were developed using the state of art technologies that can be used to transform the webpages [7]. A few studies utilize URL based features to detect phishing detection [8], [9]. However, the general phishing detection approaches are not performing better than the visual detection approaches. Some research studies detect phishing webpages based on the ranking from search engines [10], [11]. Indexing and ranking perform better than the URL based approaches. Moreover, text-based approaches by employing Natural Language Processing techniques were also applied [6], [12], [13]. These approaches try to maintain a repository of content from the benign webpages to compare and detect phishing. There were recent studies using the state of art techniques in deep learning models to detect phishing [14]–[16]. Some of these approaches include empirical analysis and data augmentation [17]. Alternatively, list-based approaches for detecting phishing webpages have been used as well [18], [19]. These approaches use whitelist and blacklists to keep a track of benign and phishing URLs.

There are some limitations of the aforementioned techniques. The phishing webpages that are created using the same logos as the original webpage and similar HTML script for the layout may not be easy to detect with the visual approaches. URL based features will not be able to detect lengthy benign URLs or few such outliers based on heuristic rules. Moreover, search engine based approaches will not be able detect phishing in the wild and zero-day phishing webpages. In addition, the text-based approaches use repositories to store content and need a huge storage space. Also, these approaches might not be able to detect the phishing webpages that uses the same content as benign pages. The deep learning state of art approaches have usual issues like manual parameter-tuning, long training time, deficient detection accuracy, etc. The list-

[1]https://expertinsights.com/insights/50-phishing-stats-you-should-know/

based approaches have their own disadvantages like keeping track of recent webpages and updating the lists. Also, these approaches take a lot of time to compare with all URLs, and a decision cannot be made if the URL is in neither of the lists.

Web scraping is a technique to extract the data from the websites. The websites have a lot of public information that can be extracted by automated software to be used for different purposes. Hybrid set of features can be extracted from webpages to overcome the disadvantages or problems from each feature set as mentioned above. In recent years, many research studies focus on webpage feature detection either by utilizing one of the feature sets or by using complex CNNs and deep neural net models.

There is no perfect approach in the Internet era to detect the phishing web pages. There are different approaches like URL analysis, domain analysis, content analysis or search-based approaches. Each approach has its own advantages and limitations. To develop a system that is more accurate in phishing detection, the choice of approach and features plays a vital role. A model that depends on web scraped features for detecting the webpages that detect as accurately as possible need to be developed. The main intention of this paper is to extract hybrid features by employing web scraping techniques to train the machine learning models that can detect the phishing webpages. In this paper, web scraped features are tested to check if they perform any better than other features.

In this paper, web scraping is employed to extract a hybrid feature set for each URL present in the data. The hybrid feature set includes search engine-based features, URL based features, network layer-based features, domain-based features, and content-based features. Most of the features of webpage are researched and cited in an article by Guptta et. al. [20]. Machine Learning models like XGBoost, Logistic Regression, Multilayer Perceptron, Support Vector Machines, Decision Trees, XGBoost, AutoEncoder, Naive Bayes, etc., are built on these extracted features to detect phishing webpages. All the models are tested by implementing Principal Component Analysis (PCA) to check for any improvement.

The XGBoost model with the features extracted from web scraping is the best model on all the datasets considered. The accuracy on all diverse datasets with the proposed scheme is greater than 0.98 with XGBoost classifier. The major contributions of this paper can be summarized as follows:

- Extracting a comprehensive set of hybrid features from each URL by implementing web scraping methods
- Determining if hybrid feature extraction using web scraping technique would give better model accuracies in comparison to other techniques on a public phishing dataset.

This paper is organized as follows. The following section provides the related work. Section III explains the proposed method. The experiments are discussed in Section IV. The last section concludes our paper.

II. Related Work

Detecting phishing web pages has been an ongoing research for a long time. There are many features like URLs, screenshots of a web page, HTML content, etc. that can be used in phishing web page detection. Boyapati et al. [21] in their paper have reviewed all the phishing detection techniques. Visual similarity based methods are quite popular for a while now. Lin et al. [22] used Faster-RCNN and Siamese pipelined deep learning models to identify phishing pages based on visual similarity. Phishpedia model was implemented by accurate recognition of identity logos on webpage screenshots and matching the logo variants of the same brand. It gives relatively accurate results in phishing detection than any other baseline approaches. They were able to achieve 98.2% precision when compared to other viual similarity based approaches like EMD [23] and PhishZoo [24]. Fu et al. [23] proposed an approach which employs Earth Mover's Distance (EMD) to detect phishing by calculating the visual similarity of webpages. Afroz et al. [24] developed PhishZoo using a database to store profiles and images of websites to compare the loaded sites with the stored profiles. To detect phishing sites more robustly, visual similarity based techniques were used.

Phishing webpage detection can be performed by extracting multiple features from webpages. Some of these techniques utilize machine learning algorithms on just the URL. Yuan et al. [25] came up with a framework where character embedding was implemented using SkipGram. They also adopted CBOW to predict the missing words and doc2vec, which converts the URL into vectors. Classification algorithms were applied to the vector representations of URLs to detect the phishing websites. Compared to other classification algorithms, XGBoost gives higher performance with an accuracy of 98.69%. Some of the other methods is using the data present in the dataset. On the other hand, Verma et al. [26] used character N-grams (overlapping sequences of N consecutive characters) to extract features from the URL. They used learning algorithms like perceptron, passive-aggressive and adaptive regularization of weights to classify and compare the performance. Wang et al. [14] proposed a URL detection model using deep learning. They followed word embedding based on character embedding for vector expression of the URL. Dynamic Convolutional Neural Networks (DCNNs) were used for feature extraction which replaces pooling layer with k-max pooling layer. Finally DCNN was connected to a fully connected layer while adopting Adam optimization algorithm to find the best results. They were able to achieve an accuracy of 95.8% Many research studies in phishing web page detection tried to rely on deep neural networks and CNNs.

In addition to analysis of URLs, feature extraction and analysis of combined features is an alternate way of phishing detection analysis. Chiew et al. [27] developed Hybrid Ensemble Feature Selection (HEFS), a machine learning based phishing detection system. A cumulative Distribution Function Gradient (CDF-g) is used to get the primary feature subsets

which are then sent to a data perturbation ensemble to get a secondary subset of features. HEFS gave best results with random forest classifier in detecting phishing websites. Guptta et al. [20] proposed a hybrid feature based anti-phishing procedure to utilize features from a URL and hyperlink based features from a website. They combined extracted features to generate hybrid features from URL and content of the webpage, which are used to train the models for website classification by using machine learning classifiers. URL based features on a Kaggle dataset gave an accuracy of 98.42%. The accuracy of hyperlink features/ content based was 84.67%. They conclude stating URL-based and hyperlink-based features are useful for individual web-based detection, but they are not sufficient to detect various types of phishing URLs. If a hybrid feature set is used to detect phishing URLs, we could be able to detect phishing attacks more accurately. Acharya et. al [28] proposed a scheme to utilize crawlers to detect phishing by using prior profiling methods but the accuracy by utilizing only search engine based techniques was around 92.7%. Content based features have also been studied for phishing detection. Zhang et al. [29] put forward CANTINA, a content-based anti-phishing technique that makes use of TF-IDF for detecting malicious sites. However, this process suffered from performance issues as it has large data set and takes more time in querying networks. *On the basis of results given different feature sets, in this paper we integrate all possible hybrid features from URL, content, domain, network, and search engine rankings.*

A few researchers tried to compare machine learning models for phishing detection. Shahrivari et al. [30] provided a relative and analytical evaluation of twelve different machine learning classifiers to detect phishing websites. For getting the best results, feature selection methods and hyperparameter tunings were applied. They followed the process of ensembling the classifiers which gave the best performance in computation of duration and accuracy. Marchal et al. [31] aimed to classify phishing by using a few key terms. If it is a phishing page, then the target is identified by relying on the key terms in the webpage. Rao et al. [32] proposed a model where the login field is fed with fake credentials, followed by a filtering process comprised of LoginCheck, FeedFakeCredentials and HeuristicsCheck to classify whether the site is phishing or benign. Among these filters, the FeedFakeCredentials filter gave notable results than other ones.

Among aforementioned approaches, Phishpedia has attained better accuracies in visual similarity based approaches. We have implemented our web scraping based model with hybrid feature set on their data set to compare the accuracies. Other datasets that were considered include PhishTank and Kaggle. In this paper, hybrid feature set are extracted from the web by employing customized web scraping techniques that uses web crawler and scraper to detect phishing webpages with better accuracies on diverse datasets. Different machine learning classifiers are tested to determine the best model achieving the highest accuracy.

III. Proposed Method

We propose a framework shown in Figure 1 to detect phishing websites using machine learning models with hybrid features extracted from web scraping. The schematic flow of the process is shown in Figure 2. The machine learning models are applied on the extracted hybrid features. Each model is trained before and after applying Principal Component Analysis (PCA) as a feature reduction technique. A train and test split of 70 and 30 percentage was employed on the dataset. The models are trained and tested for performance evaluation. The accuracy of the model is being taken as an evaluation metric.

Using the URL dataset collected from the different data sources, the web pages are scraped for the features as described in the section below during the training process. Features extracted are illustrated in Figure 3. The web scraping techniques are employed using Python libraries.

A. Features for Phishing from Web Pages

A detailed description of the features selected and used are given below.

1) *URL Features:* This feature set includes extracted from the URL of the webpage.
   - Length of URL: Usually, the length of URL more than 60 characters is an indicator of phishing web page.
   - Length of host name: The longer the length of host name, the higher is the probability that web page is phishing.
   - Number of special characters: Usually, the presence of numerous special characters like space, dot, ampersand, hash, hyphen, etc., is considered abnormal.
   - Ratio of digits: The number of digits to letter count should be small in the URL. If this ratio exceeds a threshold, the URL is likely to be a phishing webpage.
   - Https protocol: Presence of HTTPS is preferred for any communication on the Internet.
   - Shortening services: If any URL shortening service is used and the long URL is converted to case-sensitive alphanumeric code, the webpage is a candidate for a phishing webpage.
   - Redirection: The URL should never be redirected to another URL when a user intends to open a web page.
2) *Network layer-based features:* This feature set is extracted from the network layer of the webpage.
   - IP address: Presence and validation of IP address is one of the important features to determine phishing. If the IP cannot be validated, the webpage is likely to be a phishing webpage.
   - Port: Port on which the webpage is hosted indirectly reveals the protocol (http or https) or any encryption used on the webpage communication channels.

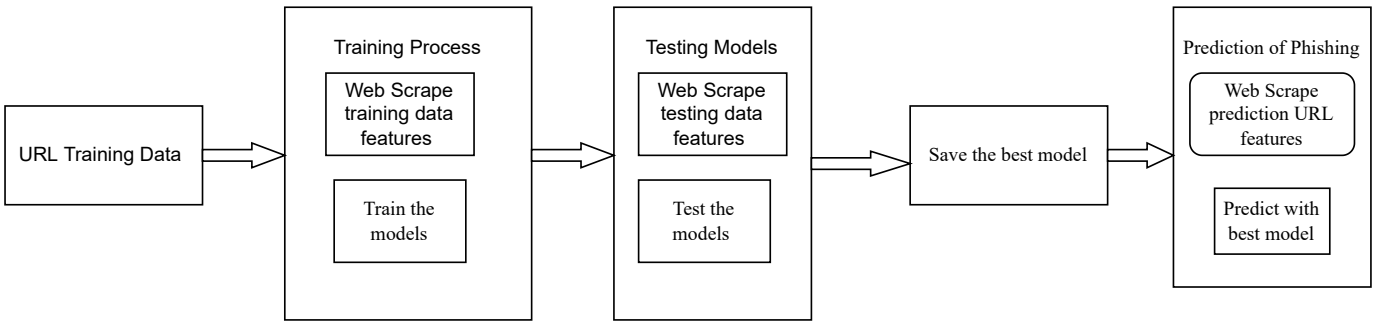

Fig. 1. Modeling process with Web Scraping module

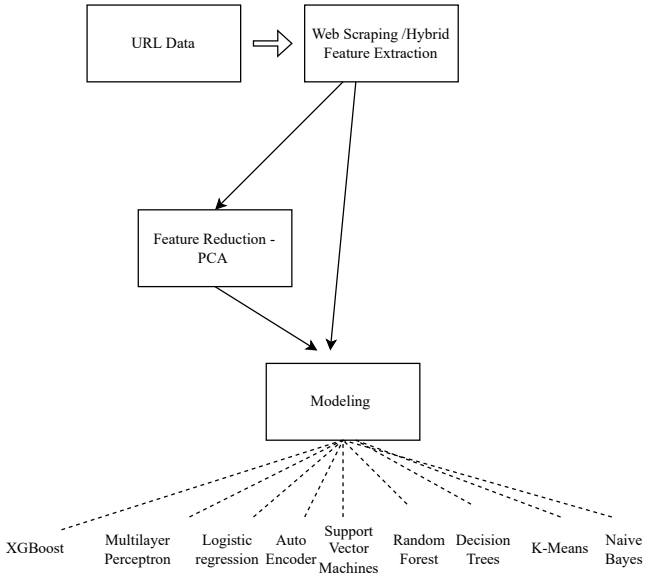

Fig. 2. Schematic Flow

3) *Content-based features:* The feature set is extracted from the HTML or CSS content of the webpages.
   - Number of hyperlinks: There should not be a lot of hyperlinks on webpages. In this work, we have considered webpages with more than 10 hyperlinks per page as an indicator of phishing.
   - Number of iFrames: The iFrame count should be as low as possible. The advertisements might have a hidden malicious URL.
   - Pop-up windows: A webpage should not have a lot of pop-up windows as there might be malicious links in the window.
   - Mouse-over: When a mouse is hovered over a blank space or anywhere on the webpage, it should not be redirected to malicious links.
   - Right-click: Right-click should not be disabled on the webpages. Disabling right click on the webpage can be considered suspicious.
4) *Domain-based features:* This feature set is extracted from the domain of the webpage.
   - Domain registration length: The domain should be registered for at least two more years in the mere future. If the domain is taken for shorter periods, it can be suspected as a phishing page as the adversaries' main goal is to steal credentials and disappear.
   - Domain age: Age of the domain is also an important factor to consider. New domains or zero-day web pages usually have less domain age and high likelihood that they might be phishing.
   - Domain validation: Invalidated domains could be signs of phishing.
   - Whois registration: Availability of whois registration information is seen in legitimate webpages. Whois info has many details like organization name, domain name, registration date, etc.
   - Domain copyright: Checking if the domain has a copyright is another important feature. It indicates that the web page content should not be copied as an indicator of legitimate web page.
5) *Search engine-based features:* This feature set is extracted from the search engines such as Google and Alexa are listed.
   - Alexa ranking: Ranking domain based on the webpage traffic, familiarity, etc. by Alexa is considered as legitimacy. Domain name can be searched, and URL can be compared with the highest ranked webpages. Legitimate webpages usually have low ranks.
   - Google indexing: Google indexes webpage domains and subdomains after verification. Web page having an index on Google can be considered as a legitimate webpage.

B. Feature reduction using PCA

The multi-collinearity among the extracted features might sometimes lower the performance of the model if the feature set is directly used for modeling. The Principal component Analysis (PCA) is employed to the data set to see if the performance of the classifiers can be improved in any way by reducing the multi-collinearity. The most prominent principle components are used to curtail the redundancy among features.

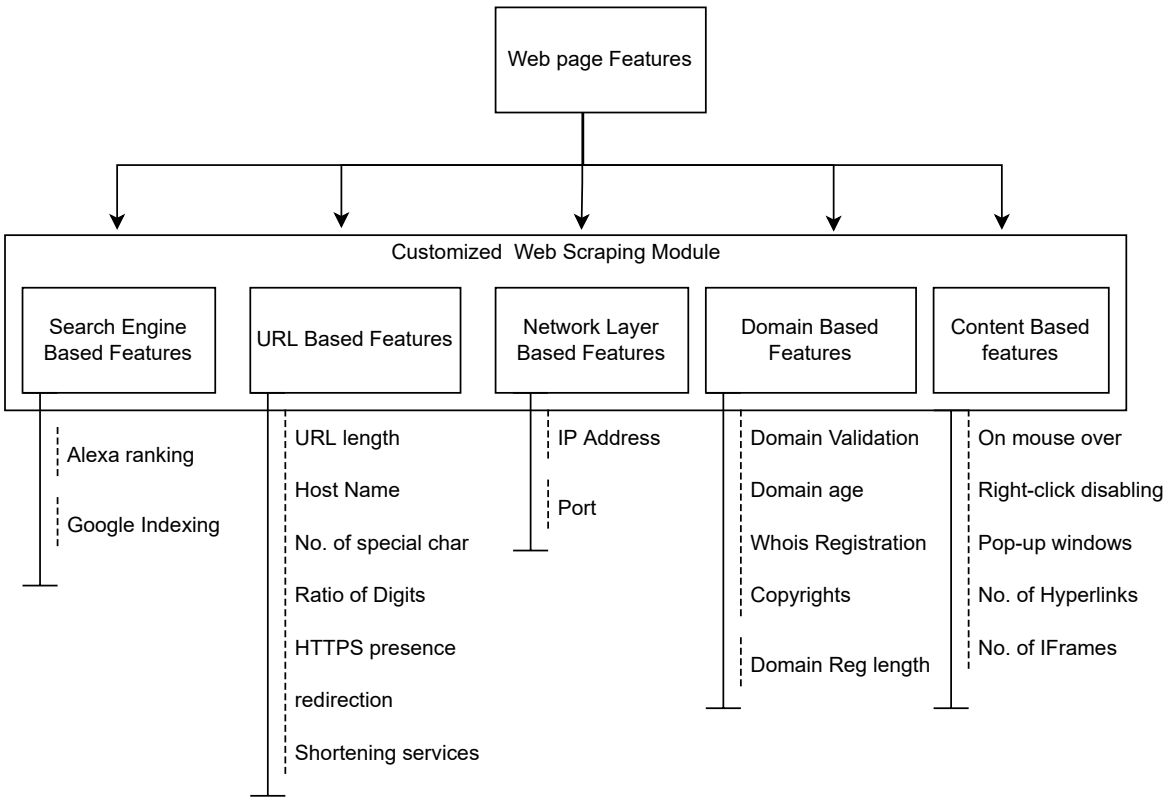

Fig. 3. Hybrid Features Extracted by Customizing Web Scraping

The feature extracted data sets were used for training machine learning models with and without applying PCA technique.

C. Web Scraping

Getting data manually from each webpage is a long process. Automated web scraping is always preferred when data needs to be obtained from numerous webpages. Web scraping uses intelligence automated methods to get data sets from different webpages in a short time. Most of the data obtained from webpages will be in an unstructured format. Web scraping automatically converts the unstructured data to a structured format and stores them in a database or spreadsheets so that the user can use it for different applications. There are different ways to obtain the necessary information from the webpages by scraping them. By customizing the web scraping module, the data needed for the feature extraction from the webpages is collected in this paper. The process of web scraping is illustrated in the Figure 4.

Web scraping has two essential components: crawler and scraper. The crawler is an Artificial Intelligence (AI) algorithm that browses the web to collect the data that is required from an input URL. Once the crawler locates the particular data, the scraper will be able to extract the data from the webpages. The ideal scenario is to usually specify the particulars of the data needed so that the scraping would be fast and efficient.

The URL data is given as an input from the web scraper. The crawler would go ahead and search for the relevant URL on the web. The scraper initially loads all the HTML, CSS, and JavaScript contents. Then, the scraper would extract the specified information and outputs the data in a specified format. In our training process, the scraper is designed to spawn the information into Excel files or spreadsheets.

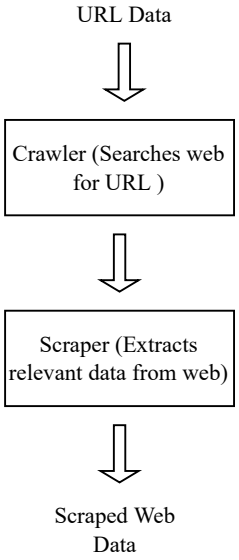

Fig. 4. Web Scraping Methodology

IV. Experiments

We have conducted our experiments using High Performance Computing (HPC) cluster at Kennesaw State University [33] for fast web scraping as well as training machine learning models using Python3. Python packages used in the experiments include Pandas, Numpy, Whois, Urllib, Socket, Requests, Beautifulsoup, Validators, Seaborn, Sklearn, Keras, Tensorflows, PyTorch.

A. Dataset Properties

Three datasets are used to train and test the classification models.

1) *OpenFish/FishTank Data Set:* This data set has 12,564 records and is obtained from both Phishing and Legitimate URLs. Phishing URLs are collected from OpenPhish and PhishTank. Legitimate URLs are taken from Google top ranked webpages. The collected URL data is used for performing web scraping.
2) *Kaggle Data set:* This data set from Kaggle has 11,482 records and 87 features for each URL[2]. This dataset is also used to train and test to check for the performance of the developed models on a diverse dataset. The URLs from this dataset are used to generate the same set of hybrid features extracted as other datasets.
3) *Phishpedia data set:* This dataset is available publicly[3] and contains URL, logo, and HTML contents. The URL information is used and scraping is applied on this dataset.

B. Modeling and Results

The three datasets with URLs were considered for training process. Scikit learn methods are used for defining and fitting supervised learning classifiers. Tensorflow is used for defining and fitting Multilayer Neural Networks and Auto Encoder. The URL datasets are sent to web scraper to get the hybrid feature datasets. The feature sets are then sent into the model for training. The test-train split used for this model is 70-30. Once the training is done, the test set is used for prediction. The prediction accuracy is calculated by comparing the generated label to the original label. The classifier that performs the best on all the datasets is saved.

In the experiments, we have evaluated ensemble tree algorithms (XGBoost, Decision Trees, Random Forests), unsupervised learning algorithms (K-means), machine learning algorithms (Support Vector Machines, Logistic Regression, Naive Bayes), and neural network models (Multi-layer Perceptrons, Auto Encoders). The performance of these models are provided in Table I. The results show that the accuracy of XGBoost and Multilayer perceptron are above 95% on all the datasets. The accuracy of Decision Trees and Random Forest is above 90% on all the datasets. Logistic Regression, SVM and Naive Bayes performance is between 60% and 90%. The classifiers based on unsupervised learning techniques were used with the number of clusters set as two. Once the clustering is done, the propagated labels were compared with the original labels to the obtain the accuracies. K-means and Auto Encoders accuracies are in the range of 60% to 85% on all the datasets. The original features that tend to be most important for making a decision on the best performing model (XGBoost) are the ratio of digits, Alexa ranking, domain validation, length of URL, and shortening services.

We have applied PCA for feature reduction and used only two principal components to check to see the reduced features can simplify phishing detection. Rather than to expect increase in accuracies, our goal was to evaluate a few PCA features would be enough to get similar accuracies. The accuracies after applying PCA on Phishpedia dataset have maintained similar accuracies. The accuracy is still low after applying the PCA on two datasets with K-Means and Auto Encoders, which is in the range of 50% to 68%. Surprisingly, multi-layer perceptron was able to achieve better accuracy for 200 epochs with PCA on all the datasets. Overall, XGBoost classifier performs a lot better than other classifiers even after applying PCA. The performance results using PCA are shown in Table II. These results show that some of the classifiers perform as good as the original classifiers by using just 2 features from PCA.

Different approaches were applied on the Phishpedia dataset that has 30k benign and 30k phishing web URLs, and the comparison of these methods is provided in Table III. Aong others, SenseInput [34] approach was able to achieve the highest precision and recall of 96.78% and 94.98% on Phishpedia dataset as reported in their paper. Our method of using web scraped hybrid features is able to achieve even better precision and recall of 98.91% and 97.94% respectively.

C. Discussion on Phishing Detection

The URL that is randomly detected by the crawler in the wild can be scraped, and the features can be extracted. The model will be able to detect the phishing in the wild as there will be no prior repository of particular URLs. The wild phishing detection is also possible and is similar to phishing detection for any other webpage. A URL is randomly passed into our model for prediction, and the model was able to detect phishing by scraping its feature set. One among 20 URLs from the wild is misclassified as phishing by our model.

Zero-day phishing webpages are webpages that try to steal credentials and assets from users starting on the day they are launched. In other words, zero-day webpages are the pages that have a life of one day. Due to the presence of search engine based feature extraction in the model, zero-day phishing webpages can be easily detected. For zero-day phishing detection, Alexa ranking serves as an important feature. Some webpages that are not phishing and were just born could be the outliers. However, our model is able to detect those outliers by relying on other features. Nine out of ten zero-day phishing webpages are properly detected in this way.

[2]https://www.kaggle.com/manishkc06/web-page-phishing-detection
[3]https://sites.google.com/view/phishpedia-site/home

TABLE I
MODELING RESULTS ON CLASSIFIERS

	Dataset 1: Recent PhishTank Data		**Dataset 2: Kaggle Data**		**Dataset 3: Phishpedia Data**	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
XGBoost	**1.000**	**0.991**	**1.000**	**0.981**	**0.996**	**0.990**
Multilayer Perceptron	0.962	0.962	0.983	0.957	0.953	0.951
Random Forest	0.958	0.959	0.950	0.944	0.903	0.901
Decision Tree	0.951	0.952	0.943	0.926	0.903	0.901
SVM	0.888	0.891	0.946	0.943	0.869	0.864
Logistic Regression	0.866	0.861	0.941	0.938	0.874	0.869
Naive Bayes	0.619	0.609	0.716	0.721	0.850	0.848
K-Means	0.687	0.680	0.612	0.621	0.827	0.822
AutoEncoder	0.843	0.839	0.786	0.782	0.73	0.727

TABLE II
MODELING RESULTS ON CLASSIFIERS AFTER PCA

	Dataset 1: Recent PhishTank Data		**Dataset 2: Kaggle Data**		**Dataset 3: Phishpedia Data**	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
XGBoost with PCA	1.000	0.986	1.000	0.976	0.961	0.981
Multilayer Perceptron with PCA	0.982	0.975	0.990	0.966	0.963	0.961
Random Forest with PCA	0.958	0.951	0.926	0.915	0.900	0.897
Decision Tree with PCA	0.905	0.896	0.886	0.875	0.900	0.897
SVM with PCA	0.889	0.891	0.946	0.943	0.869	0.864
Logistic Regression with PCA	0.866	0.861	0.941	0.938	0.874	0.869
Naive Bayes with PCA	0.647	0.642	0.683	0.694	0.871	0.867
Autoencoder with PCA	0.644	0.640	0.525	0.536	0.966	0.964
K-Means with PCA	0.687	0.68	0.610	0.619	0.827	0.822

TABLE III
PHISHING DETECTION ACCURACY USING DIFFERENT MODELS/METHODS ON PHISHPEDIA DATASET

Approach	**Precision**	**Recall**	**F1- score**
CANTINA + [35]	87.47%	37.80%	52.79%
Knowing the Domain [36]	97.68%	59.33%	73.82%
URL based approach [37]	98.30%	88.97%	93.67%
Phishpedia [5]	96.77%	82.89%	89.29%
SenseINput [34]	96.78%	94.98%	95.87%
Web Scraping based hybrid Features (Our Approach)	98.91%	97.94%	98.39%

V. CONCLUSION

In this paper, we have analyzed the classification of phishing webpage detection utilizing hybrid feature extraction using web scraping and benefiting different machine learning models. The accuracy of classifiers is compared with and without applying PCA. XGBoost algorithm outperformed all other classifiers on all three datasets by using the web scraped hybrid feature set and achieved 99% accuracy on two datasets. The features that contributed the most on the XGBoost algorithm are ratio of digits, Alexa ranking, domain validation, length of URL, and shortening services. Usually, the feature extraction is applied through artificially learned models. The performance of the models is dependent on the quality and reliability of the extracted features. By looking at the results, we can conclude that web scraping based hybrid feature set with URL, domain, network, search engine based, and content based features is giving good results when compared to other methods like CNN character embedding as mentioned in the related work. Also, the web scraping based hybrid feature extracted model achieved higher accuracies in terms of precision and recall when compared to other methods like CANTINA+, URL based approaches, SenseInput, Knowing Thy Domain, and Phishpedia on the Phishpedia dataset.

We plan to extend this work in the future as follows. The current features utilized do not make use of logos or visual similarity of webpages. By comparing the visual similarity of webpages, there is a chance of detecting the phishing webpages. Although we have obtained promising results on the three data sets, the proposed features should be evaluated for larger data sets. To experiment on larger datasets, the web scraping algorithm needs to be optimized. Another direction for improvement would be trying different feature set combinations during web scraping.

ACKNOWLEDGEMENT

This work was supported in part by research computing resources and technical expertise via a partnership between

Kennesaw State University's Office of the Vice President for Research and the Office of the CIO and Vice President for Information Technology.

REFERENCES

[1] U. Karabiyik, A. M. Canbaz, A. Aksoy, T. Tuna, E. Akbas, B. Gonen, and R. S. Aygun, "A survey of social network forensics," 2016.

[2] T. Tuna, E. Akbas, A. Aksoy, M. A. Canbaz, U. Karabiyik, B. Gonen, and R. Aygun, "User characterization for online social networks," *Social Network Analysis and Mining*, vol. 6, no. 1, pp. 1–28, 2016.

[3] L. Barlow, G. Bendiab, S. Shiaeles, and N. Savage, "A novel approach to detect phishing attacks using binary visualisation and machine learning," in *Proceedings of the IEEE World Congress on Services (SERVICES)*. IEEE, October 2020, pp. 177–182.

[4] S. Abdelnabi, K. Krombholz, and M. Fritz, "VisualPhishNet: Zero-day phishing website detection by visual similarity," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, October 2020, p. 1681–1698.

[5] 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 *Proceedings of the 30th USENIX Security Symposium*. USENIX, August 2021, pp. 3793–3810.

[6] B. van Dooremaal, P. Burda, L. Allodi, and N. Zannone, "Combining text and visual features to improve the identification of cloned webpages for early phishing detection," in *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES)*. ACM, August 2021, pp. 1–10.

[7] S. K. Deval, M. Tripathi, B. Bezawada, and I. Ray, ""x-phish: Days of future past": Adaptive amp; privacy preserving phishing detection," in *2021 IEEE Conference on Communications and Network Security (CNS)*, 2021, pp. 227–235.

[8] M. Abutaha, M. Ababneh, K. Mahmoud, and S. A.-H. Baddar, "Url phishing detection using machine learning techniques based on urls lexical analysis," in *2021 12th International Conference on Information and Communication Systems (ICICS)*, 2021, pp. 147–152.

[9] M. Khonji, Y. Iraqi, and A. Jones, "Lexical url analysis for discriminating phishing and legitimate e-mail messages," in *2011 International Conference for Internet Technology and Secured Transactions*, 2011, pp. 422–427.

[10] A. Niakanlahiji, B.-T. Chu, and E. Al-Shaer, "PhishMon: A machine learning framework for detecting phishing webpages," in *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, November 2018, pp. 220–225.

[11] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, are you listening? privacy perceptions, concerns and privacy-seeking behaviors with smart speakers," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, November 2018.

[12] E. S. Gualberto, R. T. De Sousa, T. P. De Brito Vieira, J. P. C. L. Da Costa, and C. G. Duque, "The answer is in the text: Multi-stage methods for phishing detection based on feature engineering," *IEEE Access*, vol. 8, pp. 223 529–223 547, December 2020.

[13] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[14] Z. Wang, S. Li, B. Wang, X. Ren, and T. Yang, "A malicious url detection model based on convolutional neural network," in *Security and Privacy in Social Networks and Big Data*, Y. Xiang, Z. Liu, and J. Li, Eds. Singapore: Springer Singapore, 2020, pp. 34–40.

[15] A. S. S. V. L. Pooja and M. Sridhar, "Analysis of phishing website detection using and bidirectional LSTM," in *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, November 2020, pp. 1620–1629.

[16] C. Opara, B. Wei, and Y. Chen, "Htmlphish: Enabling phishing web page detection by applying deep learning techniques on html analysis," in *Proceedings of the International Joint Conference on Neural Networks (IJ)*. IEEE, July 2020, pp. 1–8.

[17] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5948–5959, 2014.

[18] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 1245–1254.

[19] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: Predictive blacklisting to detect phishing attacks," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–5.

[20] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling hybrid feature-based phishing websites detection using machine learning techniques," *Annals of Data Science*, pp. 1–26, 2022.

[21] G. B. B. M. S. J. Boyapati, M., "Anti-phishing approaches in the era of the internet of things," in *Towards a Wireless Connected World: Achievements and New Technologies*, 2022.

[22] 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 *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 3793–3810.

[23] A. Y. Fu, L. Wenyin, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd)," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 301–311, 2006.

[24] S. Afroz and R. Greenstadt, "Phishzoo: Detecting phishing websites by looking at them," in *2011 IEEE Fifth International Conference on Semantic Computing*, 2011, pp. 368–375.

[25] H. Yuan, Z. Yang, X. Chen, Y. Li, and W. Liu, "Url2vec: Url modeling with character embeddings for fast and accurate phishing website detection," in *2018 IEEE Intl Conf on Parallel Distributed Processing*, 2018, pp. 265–272.

[26] R. Verma and A. Das, "What's in a url: Fast feature extraction and malicious url detection," in *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics*, 2017, pp. 55–63.

[27] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153–166, 2019.

[28] B. Acharya and P. Vadrevu, "PhishPrint: Evading phishing detection crawlers by prior profiling," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 3775–3792.

[29] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 639–648.

[30] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," *CoRR*, vol. abs/2009.11116, 2020.

[31] S. Marchal, K. Saari, N. Singh, and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 2016, pp. 323–333.

[32] R. Srinivasa Rao and A. R. Pais, "Detecting phishing websites using automation of human behavior," in *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, ser. CPSS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 33–42.

[33] Research Computing, "High performance computing system, kennesaw state university." [Online]. Available: https://digitalcommons.kennesaw.edu/training/10.

[34] S.-C. Lin, P.-C. Wl, H.-Y. Chen, T. Morikawa, T. Takahashi, and T.-N. Lin, "Senseinput: An image-based sensitive input detection scheme for phishing website detection," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 4180–4186.

[35] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "Cantina+ a feature-rich machine learning framework for detecting phishing web sites," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 2, pp. 1–28, 2011.

[36] H. Shirazi, B. Bezawada, and I. Ray, "¡i¿"kn0w thy doma1n name"¡/i¿: Unbiased phishing detection using domain name based features," in *Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies (SACMAT)*. ACM, June 2018, p. 69–75.

[37] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.