Phishing URL Detection Using Machine Learning

PROFESSOR
Dr.TATHAGATA BHATTACHARYA
Department of Computer Science
Auburn University at Montgomery
Montgomery,USA
tbhatta1@aum.edu

Alekya Ganta
Department of Computer Science
Auburn University at Montgomery
Montgomery,USA
aganta@aum.edu

Pavan Kumar Chitturi
Department of Computer Science
Auburn University at Montgomery
Montgomery,USA
pchittur@aum.edu

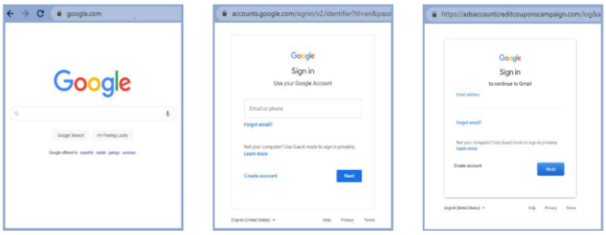
Siva Ganesh Kudumula
Department of Computer Science
Auburn University at Montgomery
Montgomery,USA
skudumul@aum.edu

***Abstract*— Phishing involves cybercriminals tricking users into disclosing credentials through deceptive login forms, transmitting data to malicious servers. This research compares machine learning and deep learning approaches for detecting phishing websites through URL analysis. In contrast to common practices that exclude login pages from the legitimate class, we include both, emphasizing real-world scenarios. Our analysis reveals higher false-positive rates when testing with URLs from legitimate login pages. Utilizing datasets spanning several years demonstrates a decline in model accuracy over time. A frequency analysis of current phishing domains uncovers diverse techniques employed by attackers. To validate our findings, we introduce a novel dataset, Phishing Index Login URL (PILU-90K), comprising 60K legitimate URLs and 30K phishing URLs. Finally, we present a Logistic Regression model using Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction, achieving 96.50% accuracy on the login URL dataset.**

Keywords*— Phishing, Cybercrime, Phishing Websites, URL, Login, Phishing Detection, Machine Learning.

I. Introduction

In recent years, the surge in web services usage, driven by digital transformation, has led companies to offer online services such as e-banking, e-commerce, and Software as a Service (SaaS) [1]. The COVID-19 pandemic has further accelerated the shift to remote work, resulting in an increased workload for services like email, student platforms, VPNs, and company portals [2]. This shift has also expanded the potential targets for phishing attacks, where cybercriminals mimic legitimate websites to steal credentials or payment information [3], [4]. Studies indicate that phishing, along with spam emails and malicious websites, is a significant social engineering attack during the pandemic [5], [6].



(a) Legitimate homepage (b) Legitimate login page (c) Phishing web page

Figure.1

Traditionally, identifying phishing sites based on their HTTP protocol has become obsolete. In the 3rd quarter of 2017 [7], the APWG reported that less than 25% of phishing websites were hosted under HTTPS protocol, The adoption of HTTPS has risen significantly, reaching 83% in the 1st quarter of 2021 [8]. Despite providing secure communication, this creates a false sense of security for users during online transactions [9]. The Anti-Phishing Working Group (APWG) reports a substantial increase in phishing attacks, reaching 611,877 websites in the 1st quarter of 2021 [10].

List-based approaches, like Google SafeBrowsing[1], PhishTank[2], OpenPhish[3] or SmartScreen[4] are popular for phishing detection but have limitations, such as relying on reported URLs [12]-[14] and continuous database updates [11]. To address these shortcomings, automatic detection using machine learning has gained attention [15], [16]. These methods categorize into URL-based, content-based, visual features, and networking information [17].

This paper focuses on phishing detection through URLs, offering advantages like fast computation and independence from 3rd party services and language. Existing datasets often use homepage URLs, neglecting the challenge of determining the legitimacy of login forms[18],[19]. The paper introduces the Phishing Index Login URL (PILU-90K)[5] dataset, extended from PILU-60K, publicly available for research [20].

Three detection pipelines are implemented and evaluated, showcasing the struggles of models trained with legitimate homepages when classifying login URLs. The robustness of the proposed phishing detection over time is assessed, and an analysis of phishing domains is presented.

II. Literature Review

In the literature, researchers have focused on phishing detection following three main approaches: List-based and automatic detection using Machine Learning and Deep Learning techniques.

[1] https://safebrowsing.google.com/
[2] https://www.phishtank.com/
[3] https://openphish.com/
[4] https://bit.ly/2OJDYBS
[5] https://gvis.unileon.es/dataset/pilu-90k/

A. LIST-BASED:

The list-based approach, known for detecting phishing URLs [22]–[24], utilizes whitelists or blacklists based on whether they store legitimate or phishing URLs. While Jain and Gupta [24] developed a whitelist-based system that block all websites not on the list, blacklist-based systems, like Google Safe Browse or PhishNet [23], are more common due to their zero false-positive rate. However, these systems can be compromised and depend heavily on update rates, making them less robust against the high volume and short lifespan of new phishing websites, which is estimated to be 21 days on average [12].

B. MACHINE LEARNING METHODS

To overcome blacklist limitations, researchers have developed machine learning models categorized into URL-based and content-based approaches.

B.1.URL-BASED:

Buber et al.[25] implemented a URL detection system with two feature sets, The first was a 209 word vector, obtained with ‘‘StringToWordVector’’ tool from Weka.[6] The second, 17 NLP (Natural Language Processing) handcrafted feature achieving high accuracy.

Sahingoz et al.[21] defined feature sets and obtained impressive accuracy on Random Forest.

Jain and Gupta[26] built an anti-phishing system using handcrafted URL descriptors, including some obtained using 3rd party services like WHOIS registers or DNS lookups. They obtained a notable accuracy of 76.87% and 91.28% with Naive Bayes (NB) and Support Vector Machine (SVM) classifiers, respectively, on a private dataset with 35, 491 samples.

Banik and Sarma[27] optimized features and accuracy using a lexical feature selection approach with Random Forest classifier they obtained 98.57% accuracy on an extension of PWD2016 [18] dataset.

B.2. CONTENT-BASED:

Content-based works use features extracted mainly from the websites source code. However, most of the current works combine these with URLs and other 3rd party services such as WHOIS[7] [28], [29].

CANTINA[30], based on TF-IDF, achieved 95% accuracy but faced false-positive issues, leading to the improved CANTINA+[31].

Moghimi and Vorjani[32] proposed a system independent of third-party services, using handcrafted features and SVM. They used Levenshtein distance [33] to detect typo-squatting by comparing the website and resource URLs.

Adebowale et al.[34] combined features from the URL, source code, and third-party services, achieving high accuracy with Adaptive Neuro-Fuzzy Inference System (ANFIS) and combined with the Scale-Invariant Feature Trans form (SIFT) algorithm, obtaining an accuracy of 98.30% on Rami et al. [35] dataset.

Yang et al.[36] used Extreme Learning Machine with diverse feature sets, obtaining significant accuracy.

Li et al.[29] presented a stacking model combining three models with features from various sources, achieving commendable accuracy.

Rao and Pais[28] developed a classifier using URL, HTML, and third-party services, reaching high accuracy.

Sadique et al.[29] created a framework using multiple URL feature sets and achieved solid accuracy on a Random Forest classifier.

C. DEEP LEARNING:

In the realm of Deep Learning, Some sha et al.[39] proposed an LSTM model achieving 99.57% accuracy.

Aljofey et al.[40] introduced an RCNN model with a character-level matrix representation, achieving 95.02% accuracy. Al-Alyan and Al-Ahmadi[41] presented a modified CNN obtaining 95.78% accuracy, while Zhao et al.[42] compared GRU with handcrafted features, showing superior performance for automatic feature extraction.

Overall, this extensive analysis presents a comprehensive overview of various phishing detection methodologies, emphasizing their strengths and challenges in the evolving landscape of cyber threats.

III. DATASET

PHISHING INDEX LOGIN URLS (PILU-90K):

Phishers use login forms to retrieve and steal users data. As far as we are concerned, the legitimate class in most phishing datasets are represented by URLs from their homepages [18], [19]. However, most websites have their login form in different locations, making models trained with such public datasets to be biased since the URLs of homepages tend to be shorter and simpler than others. An example of this is depicted in Figure 2.

In this paper, we present an extended version of the Phishing Index Login URL (PILU-60K) dataset [20] and we name it PILU-90K. PILU-90K contains 90K URLs divided into three classes (see Figure 2): 30K legitimate URLs of homepages, 30K legitimate login URLs and 30K phishing URLs.

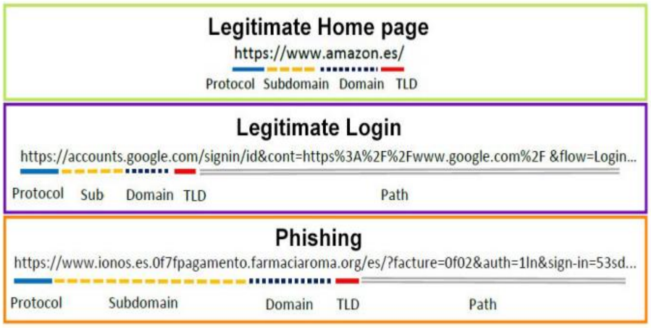

Legitimate Home page
https://www.amazon.es/
Protocol
Subdomain
Domain
TLD
Legitimate Login
https://accounts.google.com/signin/id&cont=https%3A%2F%2Fwww.google.com%2F &flow=Login...
Protocol
Sub
Domain
TLD
Path
Phishing
https://www.ionos.es.0f7fpagamento.farmaciaroma.org/es/?facture=0f02&auth=1ln&sign-in=53sd...
Protocol
Subdomain
Domain
TLD
Path

Figure.2

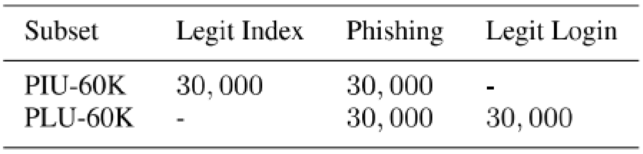
We obtained legitimate URLs from the Top Million Quantcast website[8], which lists the most visited domains in the United States. Since the provided list only included domain names, we visited the domains to extract complete URLs.

[6] https://www.cs.waikato.ac.nz/ml/weka/

[7] https://www.whois.net/

Employing the Selenium web driver[9] and Python, we navigated websites to identify login pages by checking buttons or links leading to login forms. To confirm a login form, we verified the presence of a password field. The collected phishing URLs were sourced from Phishtank [21] between November 2019 and February 2020.

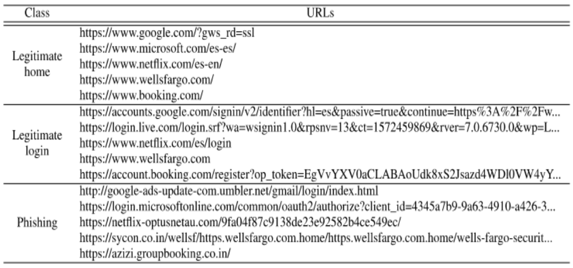
For our experiments, we created two subsets from the PILU-90K dataset. The first, PIU-60K (Phishing Index URLs), follows the configuration of many current approaches, using URLs from both legitimate and phishing homepages. The second, PLU-60K (Phishing Login URLs), follows our strategy, including URLs from both legitimate login pages and phishing sites. The distribution of URLs in each subset is detailed in Table 1.



Subset	Legit Index	Phishing	Legit Login
PIU-60K	30, 000	30, 000	-
PLU-60K	-	30, 000	30, 000

Table.1

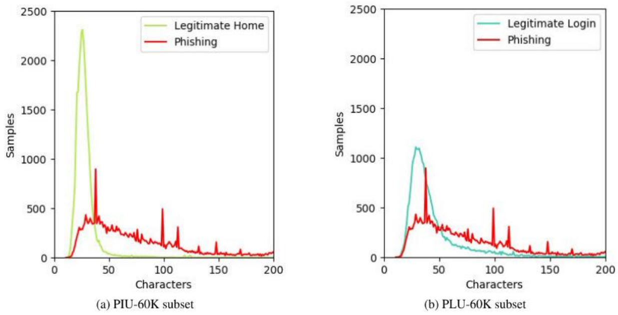
Our work introduces a unique aspect by utilizing legitimate login URLs, reflecting real-world scenarios and establishing an unbiased dataset in terms of URL length.



Class	URLs
Legitimate home	https://www.google.com/?gws_rd=ssl https://www.microsoft.com/es-es/ https://www.netflix.com/es-en/ https://www.wellsfargo.com/ https://www.booking.com/
Legitimate login	https://accounts.google.com/signin/v2/identifier?hl=es&passive=true&continue=https%3A%2F%2Fw... https://login.live.com/login.srf?wa=wsignin1.0&rpsnv=13&ct=1572459869&rver=7.0.6730.0&wp=L... https://www.netflix.com/es/login https://www.wellsfargo.com https://account.booking.com/register?op_token=EgVvYXV0aCLABAoUdk8xS2Jsazd4WDl0VW4yY...
Phishing	http://google-ads-update-com.umbler.net/gmail/login/index.html https://login.microsoftonline.com/common/oauth2/authorize?client_id=4345a7b9-9a63-4910-a426-3... https://netflix-optusnetau.com/9fa04f87c9138de23e92582b4ce549ec/ https://sycon.co.in/wellsf/https.wellsfargo.com.home/https.wellsfargo.com.home/wells-fargo-securit... https://azizi.groupbooking.co.in/

Table.2

Table 2 provides examples of URLs from each class in PILU-90K, highlighting differences, such as URL length and the presence of keywords like "login," "sign in," or "secure." Figure 3 illustrates the distribution of URL lengths in the proposed subsets, showing that PLU-60K has a more similar distribution between classes than the PIU-60K subset.



(a) PIU-60K subset

(b) PLU-60K subset

Figure.3

Apart from the feature count, PILU-90K presents a challenging scenario for URL phishing detection. A quarter of legitimate login URLs lack a path, aligning with homepage structures. Similarly, one in seven phishing samples lacks a path, posing a classification challenge, even for skilled observers.

IV. Methodology

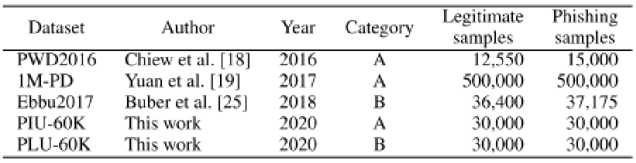
In this study, we conduct a comparative analysis of machine learning (ML) and deep learning (DL) methods for URL phishing classification. In the realm of ML techniques, our approach involves (i) utilizing the handcrafted features proposed by Sahingoz et al. [21], and (ii) employing statistical features through Term Frequency-Inverse Document Frequency (TF-IDF) combined with character N-grams. As for DL techniques, we adopt the CNN models proposed by Zhang et al. [43] and Kim [44].

A. MACHINE LEARNING TECHNIQUES

Text classification using supervised ML encompasses three primary stages: text preprocessing, text representation for converting input text into a feature vector and employing a classifier. In this section, we detail the two techniques for feature extraction along with the evaluated classifiers.

For handcrafted features, URLs are parsed using the tldextract[10] library. Subsequently, raw words are extracted from different URL parts by string splitting using specified symbols ('/', '-', '.', '@', '?', '&', '=', '_'). After preprocessing, 38 features proposed by Sahingoz et al. [21] are extracted, incorporating URL rules and NLP features. These include the frequency of symbols, the count of digits in the domain, subdomain, and path, along with their respective lengths. Additional features involve the number of subdomains, domain randomness using the Markov Chain Model, the presence of common TLDs, and the positioning of 'www' or 'com' outside the TLD. From raw words, metrics such as maximum, minimum, average, and standard deviation of word length, the number of words, compound words, words resembling famous brands or specific keywords like 'secure' or 'login', consecutive characters in the URL, and the presence of Punycode are extracted.

These features are then used to train and compare eight supervised classifiers widely used in the literature: Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), Random Forest (RF), Support Vector Machines (SVM), k-Nearest Neighbours (kNN), Naïve Bayes (NB), and Logistic Regression (LR).



Dataset	Author	Year	Category	Legitimate samples	Phishing samples
PWD2016	Chiew et al. [18]	2016	A	12,550	15,000
1M-PD	Yuan et al. [19]	2017	A	500,000	500,000
Ebbu2017	Buber et al. [25]	2018	B	36,400	37,175
PIU-60K	This work	2020	A	30,000	30,000
PLU-60K	This work	2020	B	30,000	30,000

Table.3

In natural language processing (NLP), another prevalent feature extraction technique is the TF-IDF algorithm, a statistical approach assigning weight to terms based on their occurrence frequency across documents. Given the potential absence of common word terms in URLs, we adopt character N-grams for TF-IDF, focusing on patterns of N consecutive characters. Following Al-Nabki et al. [50], we extract grams ranging from two to five characters (N = [2, 5]). Text

[8] https://www.quantcast.com/products/measure-audience-insights/

[9] https://selenium.dev/projects/

[10] https://pypi.org/project/tldextract/

preprocessing is limited to converting the text to lowercase. Extracted features are employed to train an LR classifier, chosen for its efficacy in handling noisy text tasks, such as File Name Classification [50], [51].

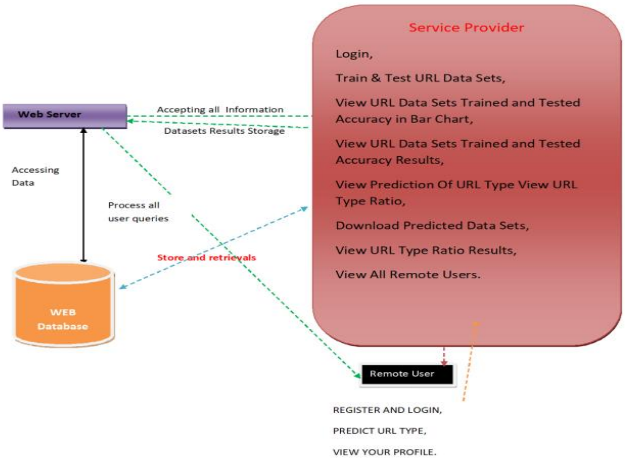
B. DEEP LEARNING TECHNIQUES

In addition to ML approaches, we explore the use of CNNs for URL [19], [41] classification, specifically architectures proposed by Zhang et al. [43] and Kim et al. [44], both operating at a character level.

Kim et al.'s model, originally designed as a character-based language model, is adapted for URL classification by replacing subsequent recurrent layers with a dense layer for SoftMax operations over classes. Conversely, Zhang et al.'s model requires no architectural modifications, as it was initially designed for text classification. It's noteworthy that no text preprocessing steps were applied to either model.

4.1 SYSTEM DESIGN:

SYSTEM ARCHITECTURE:



Service Provider
Login,
Train & Test URL Data Sets,
View URL Data Sets Trained and Tested Accuracy in Bar Chart,
View URL Data Sets Trained and Tested Accuracy Results,
View Prediction Of URL Type View URL Type Ratio,
Download Predicted Data Sets,
View URL Type Ratio Results,
View All Remote Users.
Web Server
Accepting all Information
Datasets Results Storage
Accessing Data
Process all user queries
Store and retrievals
WEB Database
Remote User
REGISTER AND LOGIN,
PREDICT URL TYPE,
VIEW YOUR PROFILE.

Figure.4

SYSTEM REQUIREMENTS:

H/W System Configuration:

1. Processor - Pentium -IV
2. RAM - 4 GB (min)
3. Hard Disk - 20 GB
4. Key Board Mouse -Standard Windows Keyboard
5. Monitor - Two or Three Button Mouse SVGA

SOFTWARE REQUIREMENTS:

1. Operating system - Windows 7 Ultimate
2. Coding Language - Python
3. Front-End – Python
4. Back-End – Django-ORM
5. Designing- HTML, CSS, JavaScript
6. Data Base - MySQL (WAMP Server)

4.2 Flow Chart: Remote User

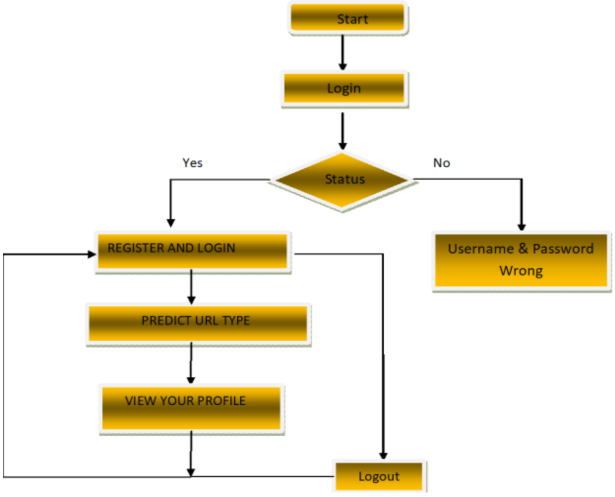
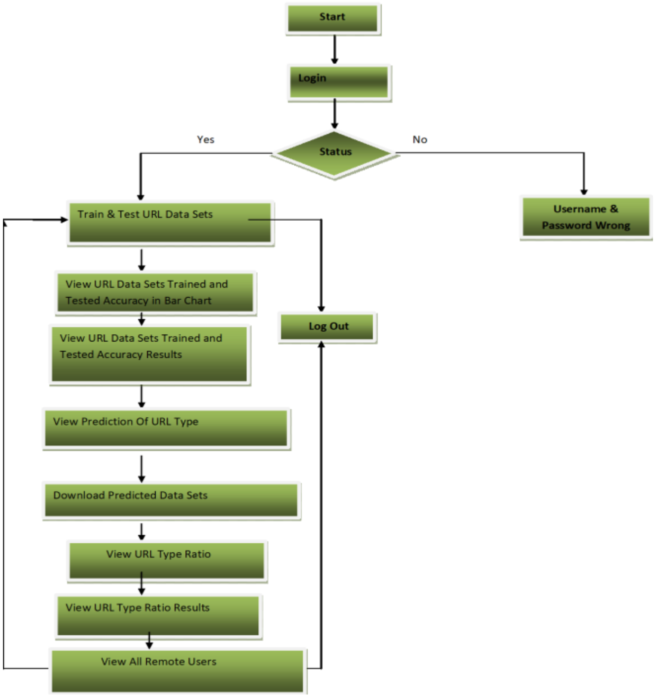


Figure.5

4.3 Flow Chart: Service Provider



Start
Login
Yes
Status
No
Train & Test URL Data Sets
Username & Password Wrong
View URL Data Sets Trained and Tested Accuracy in Bar Chart
Log Out
View URL Data Sets Trained and Tested Accuracy Results
View Prediction Of URL Type
Download Predicted Data Sets
View URL Type Ratio
View URL Type Ratio Results
View All Remote Users

Figure.6

4.4 Modules:

SERVICE PROVIDER:

Access requires a valid username and password, granting entry to numerous services. Users can explore Train and Test Data Sets for URLs, view the accuracy of these sets through a bar chart, check URL type predictions, download predicted datasets, and analyze URL type ratios.

USER MANAGEMENT

This feature enables the administrator to review a comprehensive list of registered users, providing access to user details such as usernames, email addresses, real addresses, and authorization rights.

REMOTE USER

Registration is essential for the n individuals in this section, with details stored in a database post-sign-up. Upon acceptance, users utilize their actual username and password for actions like registration, login, URL type prediction, and profile viewing.

4.5 Algorithm

1. Decision Tree Classifiers

Decision tree classifiers find applications across various disciplines, excelling in extracting detailed information for informed decision-making from given data. The process involves creating a decision tree from a training set, where items are categorized into classes C1, C2, …, Ck. The tree is formed by recursively partitioning the set based on the outcomes of a test, resulting in subsets and child decision trees for each potential outcome.

2. Gradient Boosting

Widely employed in machine learning, gradient boosting serves multiple purposes such as regression and classification. It incorporates decision trees as weak learners, showing superior performance compared to random forests. Notably, it optimizes differentiable loss functions, offering a more generalized solution during model construction.

3. K-Nearest Neighbors (KNN)

KNN is a high-performance, easy-to-implement classification method leveraging a similarity metric. It categorizes fresh data by locating its K-nearest neighbors in the training set. This instance-based learning approach operates without pre-learning, providing flexibility for various applications.

4. Logistic Regression Classifiers

Logistic regression analysis, used for categorical dependent variables, determines factors explaining the variable’s variation. The method accommodates both binary and multinomial logistic regression, offering flexibility with numerical or categorical independent variables. It provides comprehensive outputs, including regression equations, goodness of fit, odds ratio, and diagnostic reports.

5. Naïve Bayes

Naïve Bayes, based on the assumption of feature independence, is a directed learning method. While simple to implement and effective in certain contexts, it faces challenges in interpretation. However, simplifying its application through educational approaches enhances understanding and implementation. Comparative analysis with other linear methods ensures a comprehensive evaluation of results.

6. Random Forest

Random forests, an ensemble learning technique, find extensive use in classification and regression tasks. They address decision tree overfitting by aggregating predictions from multiple trees. Originating in 1995, the concept evolved to its present form, providing consistent predictions across diverse datasets.

7. Support Vector Machines (SVM)

SVM, a discriminant machine learning technique, excels in forecasting labels for new instances. It differs from generative approaches, offering advantages in reduced resource requirements and efficient handling of multidimensional feature spaces. SVM consistently provides optimal hyperplane values, ensuring stability and reliable outcomes.

V. EXPERIMENTS AND RESULTS

A. DATASETS

To test the model robustness against URLs collected in different periods, we used the five phishing datasets shown in Table 3.

These datasets are grouped into two different categories depending on their recollection strategy: (i) category A: PWD2016, 1M-PD and PIU-60K collected legitimate samples by inspecting the top-visited domains and (ii) category B: Ebbu2017 and PLU-60K visited those websites and performed further actions: in the case of Ebbu2017, its authors retrieved the inner URLs and, in the case of PLU60K, we looked for the login form page.

Therefore, most of the URLs include a path. Table 4 shows the distribution of sample structure within the datasets.

B. EXPERIMENTAL SETTINGS

Experiments are executed on an Intel Core i5. We used scikit-learn[11] and Python 3 for the implementation of the different experiments. For the machine learning experiments, we empirically assign the parameters that returned the best accuracy on the three different phishing datasets. These parameters are shown in Table 5.

[11] https://scikit-learn.org/stable

We used the averaged values of 10-fold cross-validation, reporting the accuracy (Eq. (3)), the F1-Score (Eq. (4)), the precision (Eq. (1)) and the recall (Eq. (2)) [21], [28], [34].

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

TP denotes the true positives, i.e., how many phishing websites were correctly classified. FP refers to the false positives and represents the number of legitimate samples wrongly classified as phishing. TN (i.e., the true negatives) denotes the number of legitimate samples correctly classified. Finally, FN represents the false negatives that represent the number of phishing websites misclassified as legitimate ones.

Regarding the clustering experiments, we used the same approach of Al-Nabki et al. [50] for text representation, as explained in Section IV-A and, for the clustering, we used the Agglomerative Hierarchical Clustering (AHC) [52]. The clustering process is repeated four times, and each time we initialized the AHC with the number n of the desired clusters, i.e. n ∈ {4, 5, 6, 7}

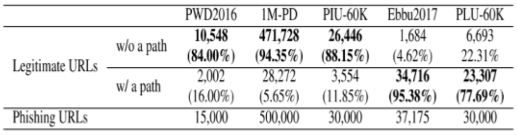
VI. RESULTS AND DISCUSSION

A. MACHINE LEARNING AND DEEP LEARNING APPROACHES

In the following, we report the result of the designed machine learning classifiers using both handcrafted and automatic feature extraction techniques. Then, deep learning approaches are presented and compared with the previous ones. Finally, we proved the impact of using legitimate login URLs against the current state-of-the-art approach.

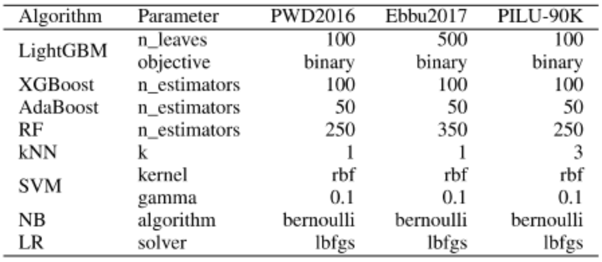
1) *HANDCRAFTED FEATURE EXTRACTION*

In this configuration, we extracted handcrafted features and benchmarked several classifiers, as explained in Section IVA. Each model was trained and tested on each subset of the PILU-90K dataset. Table 6 reports the performance of each classifier. XGBoost, LightGBM and RF outperform the rest of the classifiers on both subsets, obtaining 93.22%, 93.12% and 92.91% accuracy on PLU-60K, respectively. While for the PIU-60K sample subset, 94.63%, 94.67% and 94.42% accuracy were obtained, respectively. Results for the eight machine learning algorithms showed that Sahingoz et al. [21] descriptors achieve better performance on PIU-60K. Length-based features, the number of words and the presence of keywords enhance the performance when the difference between legitimate and phishing URLs is significant. Using the PLU-60K subset, such descriptors decrease their performance since their values are similar between classes.



		PWD2016	1M-PD	PIU-60K	Ebbu2017	PLU-60K
Legitimate URLs	w/o a path	**10,548 (84.00%)**	**471,728 (94.35%)**	**26,446 (88.15%)**	1,684 (4.62%)	6,693 22.31%
	w/ a path	2,002 (16.00%)	28,272 (5.65%)	3,554 (11.85%)	**34,716 (95.38%)**	**23,307 (77.69%)**
Phishing URLs		15,000	500,000	30,000	37,175	30,000

Table.4



Algorithm	Parameter	PWD2016	Ebbu2017	PILU-90K
LightGBM	n_leaves	100	500	100
	objective	binary	binary	binary
XGBoost	n_estimators	100	100	100
AdaBoost	n_estimators	50	50	50
RF	n_estimators	250	350	250
kNN	k	1	1	3
SVM	kernel	rbf	rbf	rbf
	gamma	0.1	0.1	0.1
NB	algorithm	bernoulli	bernoulli	bernoulli
LR	solver	lbfgs	lbfgs	lbfgs

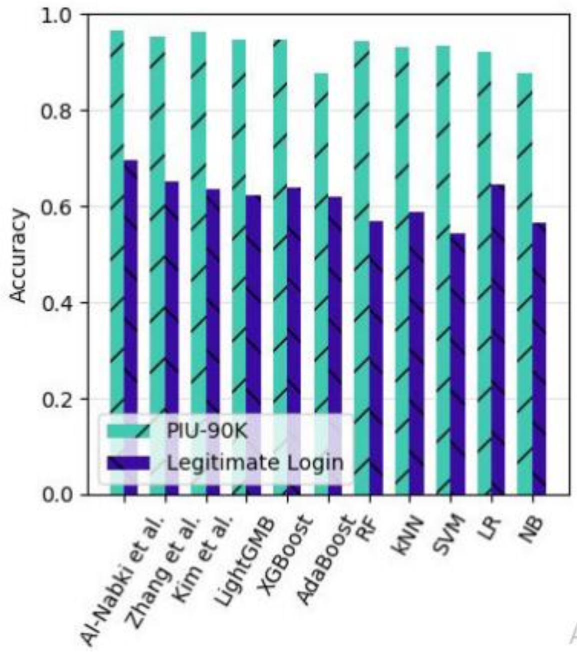
Table.5

2) *AUTOMATIC FEATURE EXTRACTION*

In this assessment, we scrutinized the classification pipeline utilizing TF-IDF and character N-gram for feature extraction, coupled with LR for classification (see Section IV-A). Results indicate superior F1-scores for automatic feature extraction methods compared to other approaches, even outperforming Deep Learning models. For PIU-60K, the classifier achieved 96.93% accuracy, while for PLU-60K, it reached 96.50%. Notably, this model surpasses benchmarked classifiers reliant on handcrafted features (refer to Table 6).

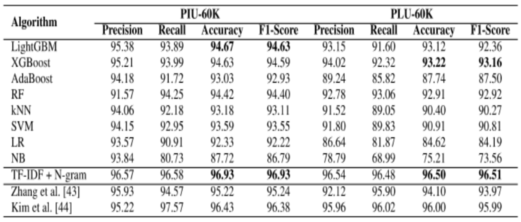
3) *EVALUATION OF DEEP LEARNING-BASED PHISHING DETECTION MODELS*

We also trained and assessed CNN character-based models proposed by Zhang et al. [43] and Kim [44] on both PILU-90K subsets. The Zhang et al. model achieved 95.22% accuracy on PIU-60K and 94.10% on PLU-60K, while Kim's model had a slightly better performance with an average accuracy of 96.43% on PIU-60K and 96.00% on PLU-60K (see Table 6). Although both CNN models outperformed handcrafted features, TF-IDF combined with N-gram [50] remained the most effective classifier for the proposed subsets.



4) IMPACT OF THE REPRESENTATION OF THE LEGITIMATE CLASS ON THE CLASSIFICATION

We explored the consequences of training URL phishing classifiers with datasets where the legitimate class is represented by homepage URLs, as exemplified by PIU-60K. Eleven classifiers were trained and assessed for accuracy, as depicted in Figure 4. Subsequently, these models classified 30,000 legitimate login URLs, revealing a substantial decrease in accuracy. For instance, the accuracy of the Al-Nabki et al. [50] model dropped by 27% to 69.50%, while SVM exhibited the lowest accuracy at 54.46%, with a decrease of 39.12%. CNN models by Zhang et al. [43] and Kim [44] achieved accuracies of 65.13% and 63.50%, respectively. Notably, our TF-IDF and N-gram approach, trained with PLU-60K, mitigates this issue by accurately classifying legitimate login samples (as shown in Table 6), effectively reducing false positives during user visits to login pages. This, however, results in a tradeoff with overall accuracy, emphasizing a reduction in false positives during login page visits.

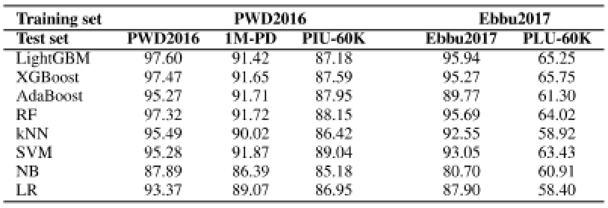


Algorithm	PIU-60K				PLU-60K			
	Precision	Recall	Accuracy	F1-Score	Precision	Recall	Accuracy	F1-Score
LightGBM	95.38	93.89	**94.67**	**94.63**	93.15	91.60	93.12	92.36
XGBoost	95.21	93.99	94.63	94.59	94.02	92.32	**93.22**	**93.16**
AdaBoost	94.18	91.72	93.03	92.93	89.24	85.82	87.74	87.50
RF	91.57	94.25	94.42	94.40	92.78	93.06	92.91	92.92
kNN	94.06	92.18	93.18	93.11	91.52	89.05	90.40	90.27
SVM	94.15	92.95	93.59	93.55	91.80	89.83	90.91	90.81
LR	93.57	90.91	92.33	92.22	86.64	81.87	84.62	84.19
NB	93.84	80.73	87.72	86.79	78.79	68.99	75.21	73.56
TF-IDF + N-gram	96.57	96.58	**96.93**	**96.93**	96.54	96.48	**96.50**	**96.51**
Zhang et al. [43]	95.93	94.57	95.22	95.24	92.12	95.90	94.10	93.97
Kim et al. [44]	95.22	97.57	96.43	96.38	95.96	96.02	96.00	95.99

Table.6

B. ANALYSIS OF PHISHING MODEL PERFORMANCE OVER TIME

Recent machine learning advancements have demonstrated strong performance when trained with PWD2016 and Ebbu2017 datasets. However, given the evolving sophistication of phishing attacks and the URLs associated with them, we postulate that models trained with outdated datasets might witness a decline in performance when assessing recent URLs.



Training set	PWD2016			Ebbu2017	
Test set	PWD2016	1M-PD	PIU-60K	Ebbu2017	PLU-60K
LightGBM	97.60	91.42	87.18	95.94	65.25
XGBoost	97.47	91.65	87.59	95.27	65.75
AdaBoost	95.27	91.71	87.95	89.77	61.30
RF	97.32	91.72	88.15	95.69	64.02
kNN	95.49	90.02	86.42	92.55	58.92
SVM	95.28	91.87	89.04	93.05	63.43
NB	87.89	86.39	85.18	80.70	60.91
LR	93.37	89.07	86.95	87.90	58.40

Table.7

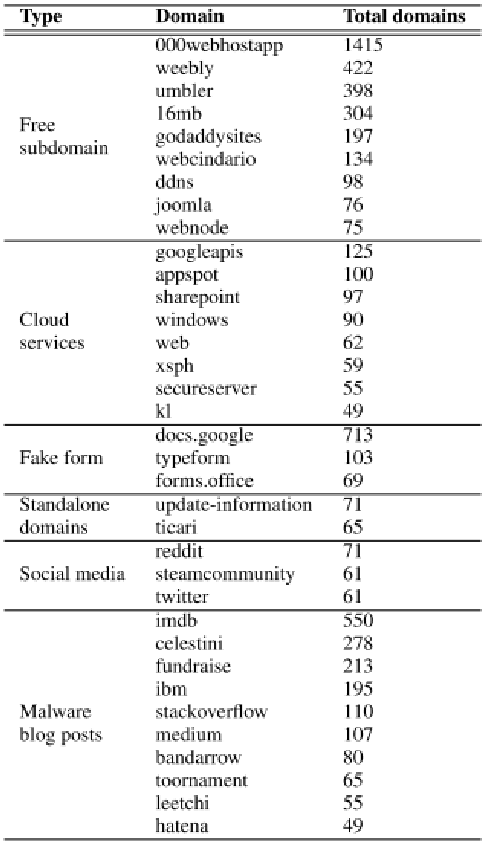
To validate this hypothesis, we employed PWD2016 and Ebbu2017 datasets, utilizing features from Sahingoz et al. [21], to train eight machine learning models (refer to Table 7). Subsequently, we tested these models using URLs from recent years, specifically 1M-PD from 2017, PIU-60K from 2020, and PLU-60K from 2020. Two categories were identified in the proposed datasets (see Section III): Category A, comprising legitimate homepage URLs without a path, and Category B, including URLs with a path. The second pipeline concentrated on URLs with a path, using Ebbu2017 for training and PLU-60K for testing.

Examining the results in Table 7, all models experienced challenges in maintaining performance over time, showcasing a decrease when assessed against datasets from subsequent years. While LightGBM exhibited the highest accuracy on both pipelines, it suffered the most significant impact over time, losing 10.42% and 30.69% accuracy in the first and second pipelines.

C. PHISHING URL CLUSTERING ANALYSIS

In this exploration, we sought to cluster phishing URLs to discern patterns Notably, at n = 7, we observed some associations between URLs.

In an endeavor to identify phishing categories, we conducted a term frequency analysis on the domain names of URLs. Utilizing the tldextract Python library, we extracted domains and sorted the results by frequency. A manual analysis of the 35 most common domains resulted in the identification of six categories outlined in Table 8.



Type	Domain	Total domains
Free subdomain	000webhostapp	1415
	weebly	422
	umbler	398
	16mb	304
	godaddysites	197
	webcindario	134
	ddns	98
	joomla	76
	webnode	75
Cloud services	googleapis	125
	appspot	100
	sharepoint	97
	windows	90
	web	62
	xsph	59
	secureserver	55
	kl	49
Fake form	docs.google	713
	typeform	103
	forms.office	69
Standalone domains	update-information	71
	ticari	65
Social media	reddit	71
	steamcommunity	61
	twitter	61
Malware blog posts	imdb	550
	celestini	278
	fundraise	213
	ibm	195
	stackoverflow	110
	medium	107
	bandarrow	80
	toornament	65
	leetchi	55
	hatena	49

Table.8

1. **Free Subdomains:** Phishers utilize services allowing the creation of custom subdomains for hosting fake websites. This tactic aids in deceiving users through popular company names or typo squatting techniques. While advantageous due to their free plans and SSL certificates, these services may have limitations on bandwidth, storage, and computation resources.
2. **Cloud Services:** Phishers leverage cloud platforms like Google or Azure to host phishing websites with SSL certificates. This strategy, offering fixed or random subdomains, entails drawbacks such as costs and the necessity for phishers to provide payment information.
3. **Fake Forms:** Phishers employ form platforms from major companies like Google or Microsoft to mimic legitimacy, urging users to input credentials. Recognizing these attacks, companies caution users against providing personal information.
4. **Social Media and Malware Blog Posts:** Domains in this category, often reported on PhishTank, lure users with promises of free recent films for download. These files are typically flagged as malware by commercial antivirus systems.
5. **Standalone Domains:** Phishers acquire or compromise standalone domains to host their websites. Some domains host various phishing campaigns over time, going online during active campaigns and offline post-campaign completion or after being reported to blacklists.

Screenshots of our experiment:

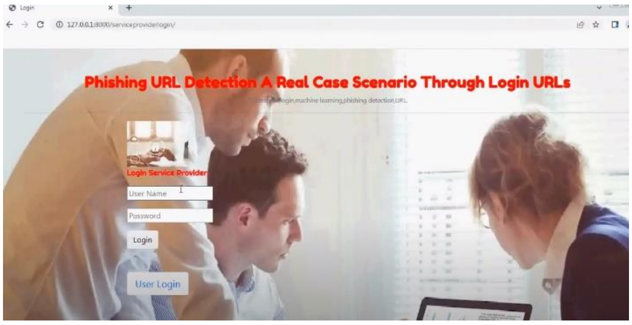
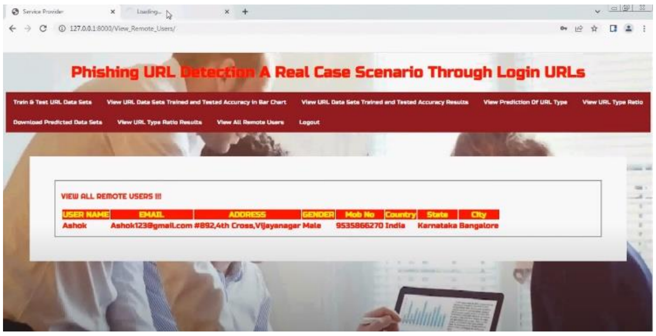

Login
127.0.0.1:8000/serviceproviderlogin/
Phishing URL Detection A Real Case Scenario Through Login URLs
Login Service Provider
User Name
Password
Login
User Login

Fig6.1

The screenshot shows a service provider's login page with fields for username and password, along with a "Login" button. The content mentions phishing URL detection and machine learning. The image may include text, a person, a human face, and indoor elements.

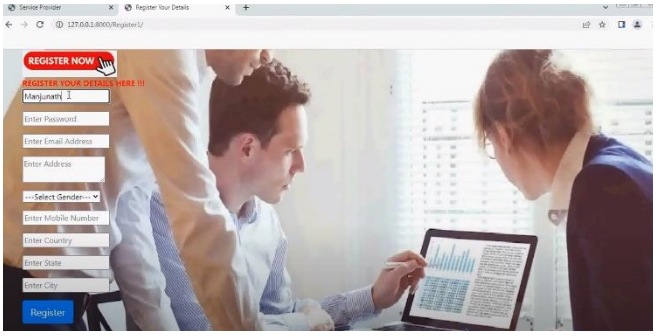

Service Provider
Loading...
127.0.0.1:8000/View_Remote_Users/
Phishing URL Detection A Real Case Scenario Through Login URLs
Train & Test URL Data Sets
View URL Data Sets Trained and Tested Accuracy in Bar Chart
View URL Data Sets Trained and Tested Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout

VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	ADDRESS	GENDER	Mob No	Country	State	City
Ashok	Ashok123@gmail.com	#892,4th Cross,Vijayanagar	Male	9535866270	India	Karnataka	Bangalore

Fig6.2

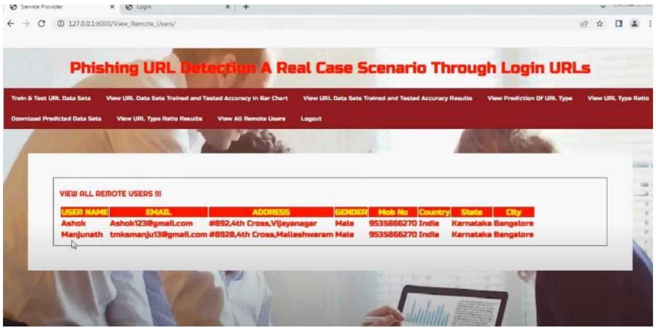
The image is a UI screenshot with features for remote user viewing, phishing URL detection, and related functionalities. User information fields like name, email, address, and gender is also visible on the interface.



Manjunath

Fig6.3

The screenshot showcases a user interface with features for remote user viewing, phishing URL detection, and other related functionalities. Additionally, user information fields, including name, email, address, and gender, are visible on the interface.

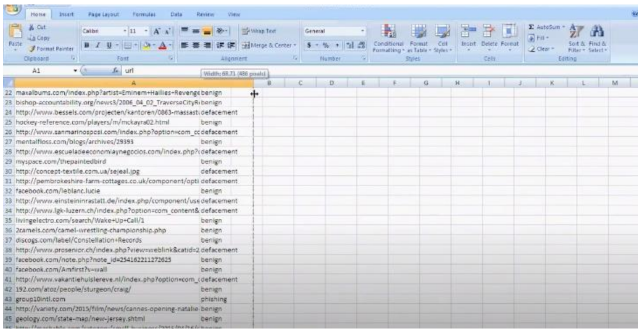

Service Provider
Login
127.0.0.1:8000/View_Remote_Users/
Phishing URL Detection A Real Case Scenario Through Login URLs
Train & Test URL Data Sets
View URL Data Sets Trained and Tested Accuracy in Bar Chart
View URL Data Sets Trained and Tested Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout

VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	ADDRESS	GENDER	Mob No	Country	State	City
Ashok	Ashok123@gmail.com	#892,4th Cross,Vijayanagar	Male	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	#8928,4th Cross,Malleshwaram	Male	9535866270	India	Karnataka	Bangalore

Fig6.4

The screenshot captures a website's graphical interface linked to a service provider, featuring login options, remote user viewing, and datasets/results for URL detection and prediction. Additionally, the interface exhibits details about remote users, encompassing their name, email, address, and contact information. The image appears to be associated with online advertising and software.

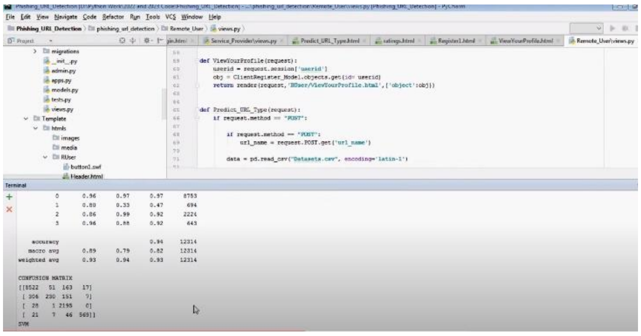


A1 url

	A	
22	maxalbums.com/index.php?artist=Eminem+Hailies+Revenge	benign
23	bishop-accountability.org/news5/2006_04_02_TraverseCityR	benign
24	http://www.bessels.com/projecten/kantoren/0863-massast	defacement
25	hockey-reference.com/players/m/mckayra02.html	benign
26	http://www.sanmarinospesi.com/index.php?option=com_cc	defacement
27	mentalfloss.com/blogs/archives/29393	benign
28	http://www.escueladeeconomiaynegocios.com/index.php?	defacement
29	myspace.com/thepaintedbird	benign
30	http://concept-textile.com.ua/sejeal.jpg	defacement
31	http://pembrokeshire-farm-cottages.co.uk/component/opti	defacement
32	facebook.com/leblanc.lucie	benign
33	http://www.einsteininrastatt.de/index.php/component/use	defacement
34	http://www.lgk-luzern.ch/index.php?option=com_content&	defacement
35	livingelectro.com/search/Wake+Up+Call/1	benign
36	2camels.com/camel-wrestling-championship.php	benign
37	discogs.com/label/Constellation+Records	benign
38	http://www.prosenior.ch/index.php?view=weblink&catid=2	defacement
39	facebook.com/note.php?note_id=254162211272625	benign
40	facebook.com/Amfirst?v=wall	benign
41	http://www.vakantiehuislereve.nl/index.php?option=com_	defacement
42	192.com/atoz/people/sturgeon/craig/	benign
43	group10intl.com	phishing
44	http://variety.com/2015/film/news/cannes-opening-natalie	benign
45	geology.com/state-map/new-jersey.shtml	benign

Fig 6.5

The screenshot displays an Excel application interface featuring a table with formatting and data manipulation options. It includes a list of URLs categorized as benign, defacement, or phishing. Tags indicate relevance to text, electronics, software, computer icon, web page, and office application.



```
def ViewYourProfile(request):
    userid = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})


def Predict_URL_Type(request):
    if request.method == "POST":

        if request.method == "POST":
            url_name = request.POST.get('url_name')

        data = pd.read_csv("Datasets.csv", encoding='latin-1')
```

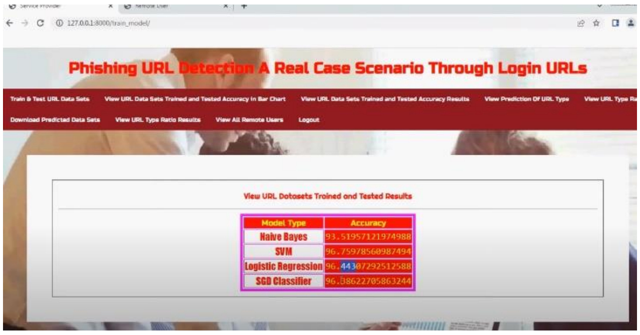
```
           0       0.96      0.97      0.97      8753
           1       0.80      0.33      0.47       694
           2       0.86      0.99      0.92      2224
           3       0.96      0.88      0.92       643

    accuracy                           0.94     12314
   macro avg       0.89      0.79      0.82     12314
weighted avg       0.93      0.94      0.93     12314

CONFUSION MATRIX
[[8522   51  163   17]
 [ 306  230  151    7]
 [  28    1 2195    0]
 [  21    7   46  569]]
SVM
```

Fig 6.6

The content appears to be a description of a graphical user interface related to an application for phishing URL detection. The information includes code snippets, file paths, and functions related to the application, as well as some statistical data. There are also mentions of web page elements and multimedia software.


127.0.0.1:8000/train_model/
Phishing URL Detection A Real Case Scenario Through Login URLs
Train & Test URL Data Sets
View URL Data Sets Trained and Tested Accuracy in Bar Chart
View URL Data Sets Trained and Tested Accuracy Results
View Prediction Of URL Type
View URL Type Ra
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout

View URL Datasets Trained and Tested Results

Model Type	Accuracy
Naive Bayes	93.51957121974988
SVM	96.75978560987494
Logistic Regression	96.44307292512588
SGD Classifier	96.38622705863244

Fig 6.7
The screenshot depicts a website interface for a service provider, offering options for remote user management, phishing URL detection, training/testing URL datasets, and accuracy results viewing.

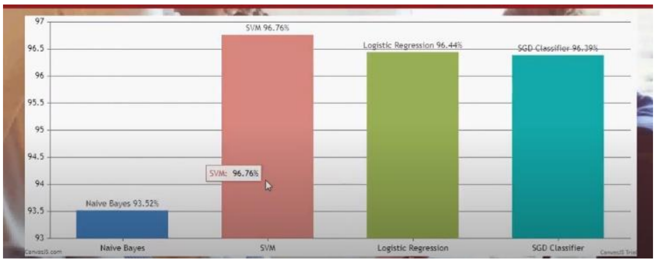
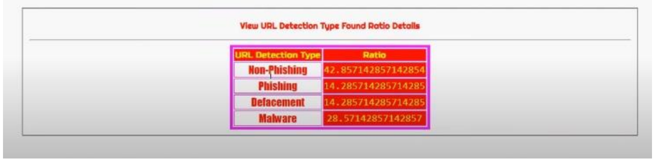


Fig 6.8

The image is a bar chart showing the accuracy percentages of different classifiers. The chart includes SVM at 96.76%, Logistic Regression at 96.44%, SGD Classifier at 96.39%, and Naive Bayes at 93.52%.



View URL Detection Type Found Ratio Details

URL Detection Type	Ratio
Non-Phishing	42.857142857142854
Phishing	14.285714285714285
Defacement	14.285714285714285
Malware	28.571428571428571

Fig 6.9

The image presents a table detailing URL detection type and their associated ratios, covering categories like Non-Phishing, Phishing, Defacement, and Malware. Tags include references to text, screenshot, rectangle, font, and design.

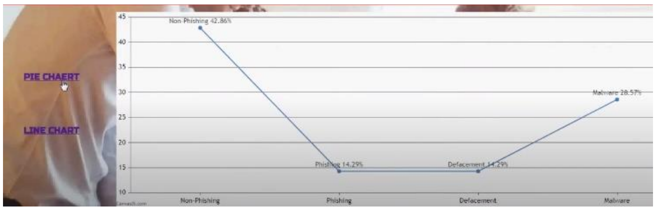


Fig 6.10

The image seems to be a screen capture of a chart featuring both a line chart and a pie chart. The line chart showcases data points for various categories, including Non-Phishing, Malware, Phishing, and Defacement. Additionally, the pie chart illustrates the distribution of these categories.

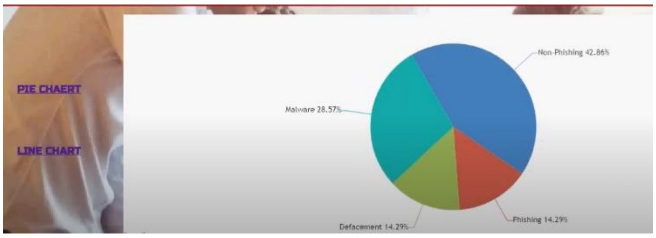
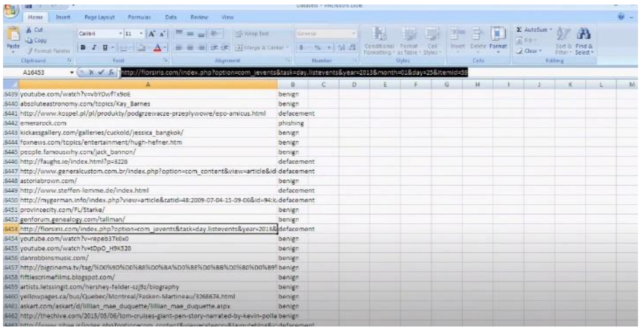


Fig 6.11

The image illustrates a pie chart that portrays various categories with their corresponding percentages. The categories consist of Non-Phishing at 42.86%, Malware at 28.57%, Defacement at 14.29%, and Phishing at 14.29%. The chart is labeled with the respective categories and their associated percentages.

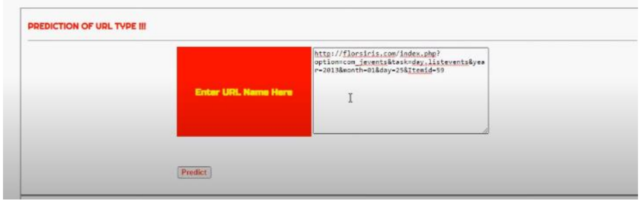


A16453 http://florsiris.com/index.php?option=com_jevents&task=day.listevents&year=2013&month=01&day=25&Itemid=59

	A	B
6439	youtube.com/watch?v=vbYDwfTx9oE	benign
6440	absoluteastronomy.com/topics/Kay_Barnes	benign
6441	http://www.kospel.pl/pl/produkty/podgrzewacze-przeplywowe/epo-amicus.html	defacement
6442	emerarock.com	phishing
6443	kickassgallery.com/galleries/cuckold/jessica_bangkok/	benign
6444	foxnews.com/topics/entertainment/hugh-hefner.htm	benign
6445	people.famouswhy.com/jack_bannon/	benign
6446	http://faughs.ie/index.html?p=3226	defacement
6447	http://www.generalcustom.com.br/index.php?option=com_content&view=article&id	defacement
6448	astoriabrown.com/	benign
6449	http://www.steffen-lemme.de/index.html	defacement
6450	http://mygerman.info/index.php?view=article&catid=48:2009-07-04-15-09-06&id=94:k	defacement
6451	provincecity.com/FL/Starke/	benign
6452	genforum.genealogy.com/tallman/	benign
6453	http://florsiris.com/index.php?option=com_jevents&task=day.listevents&year=2013&	defacement
6454	youtube.com/watch?v=repeb37k6x0	benign
6455	youtube.com/watch?v=tDpO_H9K520	benign
6456	danrobbinsmusic.com/	benign
6457	http://bigcinema.tv/tag/%D0%9D%D0%B8%D0%BA%D0%BE%D0%BB%D0%B0%D0%B9	benign
6458	fiftiescrimefilms.blogspot.com/	benign
6459	artists.letssingit.com/hershey-felder-szj9z/biography	benign
6460	yellowpages.ca/bus/Quebec/Montreal/Fasken-Martineau/3266674.html	benign
6461	askart.com/askart/d/lillian_mae_duquette/lillian_mae_duquette.aspx	benign
6462	http://thechive.com/2015/05/06/tom-cruises-giant-pen-story-narrated-by-kevin-polla	benign

Fig 6.12

The screenshot depicts a Microsoft Excel table showcasing diverse datasets and URLs categorized under Home, Insert, Page Layout, Formulas, Data, Review, and View. The table includes details about URLs and their respective classifications, such as benign, defacement, phishing, and more. The tags suggest that this is a text-based image associated with electronics, software, and web pages.



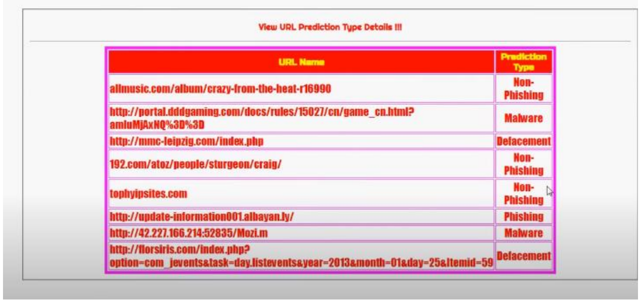
PREDICTION OF URL TYPE !!!

Enter URL Name Here

http://florsiris.com/index.php?option=com_jevents&task=day.listevents&year=2013&month=01&day=25&Itemid=59

Fig 6.13

The given information seems to describe a graphical user interface within an application, likely associated with predicting URL types. The associated tags imply that the interface may incorporate elements such as text, screenshots, rectangles, display, and design.

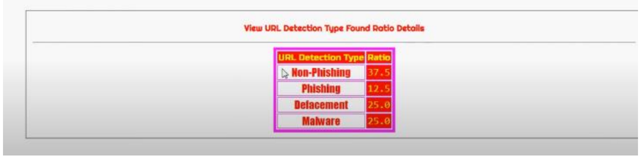


View URL Prediction Type Details !!!

URL Name	Prediction Type
allmusic.com/album/crazy-from-the-heat-r16990	Non-Phishing
http://portal.dddgaming.com/docs/rules/15027/cn/game_cn.html?amluMjAxNQ%3D%3D	Malware
http://mmc-leipzig.com/index.php	Defacement
192.com/atoz/people/sturgeon/craig/	Non-Phishing
tophyipsites.com	Non-Phishing
http://update-information001.albayan.ly/	Phishing
http://42.227.166.214:52835/Mozi.m	Malware
http://florsiris.com/index.php?option=com_jevents&task=day.listevents&year=2013&month=01&day=25&Itemid=59	Defacement

Fig 6.14

The image seems to capture a screenshot of a timeline featuring different URLs, each accompanied by its corresponding prediction type and details. The URLs are categorized into various types, including non-phishing, malware, defacement, and phishing. Tags linked to the image include text, electronics, screenshot, display, software, number, operating system, web page, font, computer icon, and website.



View URL Detection Type Found Ratio Details

URL Detection Type	Ratio
Non-Phishing	37.5
Phishing	12.5
Defacement	25.0
Malware	25.0

Fig 6.15

The image displays a table presenting the detection types along with their corresponding proportions for various URLs. The breakdown is as follows:

- Non-Phishing: 37.5%

- Phishing: 12.5%

- Defacement: 25.0%

- Malware: 25.0%

The image is tagged with descriptors such as text, screenshot, rectangle, and design.

VII. Conclusion

The primary objective of phishing detection mechanisms is to enhance existing blacklist methods, safeguarding users against deceptive login forms. Our contribution introduces the PILU-90K dataset, offering an updated resource for researchers to refine and assess their approaches. Uniquely, this dataset incorporates legitimate login URLs, mirroring a crucial scenario for real-world phishing detection.

Our investigation involves various URL-based detection models, employing both deep learning and machine learning solutions trained with phishing and legitimate home URLs. Notably, our approach exhibits a noteworthy advantage in achieving a low false-positive rate when classifying this category of URLs. Among the diverse models evaluated, the TF-IDF combined with N-gram and LR algorithm emerges as the most successful, boasting a 96.50% accuracy rate.

Comparing our methodology with the current state-of-the-art, as outlined in Section II, reveals three primary advantages:

1. **Independence from External Services:** Traditional methods relying on features such as WHOIS domain age, Google or Alexa page rankings, or online blacklists face limitations due to their dependence on external services. Our approach circumvents this constraint, ensuring real-time execution feasibility, crucial for promptly warning users against accessing phishing websites given their short lifespan.

2. **Login Website Detection:** Unlike methods trained with homepage URLs as proxies for the legitimate class, our model is trained with legitimate login websites. This ensures accurate classification of such pages, aligning with real-world scenarios where users need to discern whether a login form page is legitimate or phishing.

3. **Updated and Real-world Dataset:** PLU-60K emphasizes the use of current legitimate login URLs. Our findings underscore that models trained with outdated datasets struggle to maintain performance over time. By providing an updated phishing URL dataset reflecting contemporary trends, we facilitate enhanced learning for models, addressing the evolving landscape of phishing.

Our study reveals that phishing URL detection systems, relying solely on legitimate land page URLs for training, fall short in accurately classifying legitimate login URLs, resulting in a high false-positive rate. To enhance real-world applicability, we recommend training phishing detectors using datasets like PLU-60K focused on legitimate login websites rather than homepages. Though this approach slightly reduces overall accuracy due to the similarity between phishing and legitimate samples, it represents a fair tradeoff considering the high false-positive rates associated with state-of-the-art methods.

A domain frequency analysis uncovers distinct categories prevalent in current phishing attacks, ranging from standalone and compromised domains to the utilization of free hosting services, cloud web servers, and malware blog posts. These categories demonstrate the diverse strategies employed in phishing campaigns, balancing cost-effectiveness and efficacy.

REFERENCES

[1] Statista. (2020). Adoption Rate of Emerging Technologies in Organizations Worldwide as of 2020. Accessed: Sep. 12, 2021. [Online].

Available: https://www.statista.com/statistics/661164/worldwide-cio-survey operati%onal-priorities/

[2] R. De', N. Pandey, and A. Pal, ''Impact of digital surge during COVID 19 pandemic: A viewpoint on research and practice,'' Int. J. Inf. Manage., vol. 55, Dec. 2020, Art. no. 102171.

[3] P. Patel, D. M. Sarno, J. E. Lewis, M. Shoss, M. B. Neider, and C. J. Bohil, ''Perceptual representation of spam and phishing emails,'' Appl. Cognit. Psychol., vol. 33, no. 6, pp. 1296–1304, Nov. 2019.

[4] J. A. Chaudhry, S. A. Chaudhry, and R. G. Rittenhouse, ''Phishing attacks and defenses,'' Int. J. Secur. Appl., vol. 10, no. 1, pp. 247–256, 2016.

[5] M. Hijji and G. Alam, ''A multivocal literature review on growing social engineering based cyber-attacks/threats during the COVID-19 pandemic: Challenges and prospective solutions,'' IEEE Access, vol. 9, pp. 7152–7169, 2021.

[6] A. Alzahrani, ''Coronavirus social engineering attacks: Issues and recommendations,'' Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 5, pp. 154–161, 2020.

[7] Phishing Activity Trends Report 3Q, Anti-Phishing Working Group, International, 2017. Accessed: Sep. 12, 2021.

[8] Phishing Activity Trends Report 1Q, Anti-Phishing Working Group, International, 2021. Accessed: Sep. 14, 2021.Johnson, M. R. (2020). "Phishing in Government: Trends and Countermeasures." International Journal of Cybersecurity Research, 5(2), 78-93.

[9] R. Chen, J. Gaia, and H. R. Rao, ''An examination of the effect of recent phishing encounters on phishing susceptibility,'' Decis. Support Syst., vol. 133, Jun. 2020, Art. no. 113287.

[10] Phishing Activity Trends Report 4Q, Anti-Phishing Working Group, International, 2020. Accessed: Sep. 12, 2021.

[11] S. Bell and P. Komisarczuk, ''An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank,'' in Proc. Australas. Comput. Sci. Week Multiconf., Feb. 2020, pp. 1–11.

[12] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, A. Doupé, and G.-J. Ahn, ''Phishtime: Continuous longitudinal mea surement of the effectiveness of anti-phishing blacklists,'' in Proc. 29th USENIX Secur. Symp., 2020, pp. 379–396.

[13] L. Li, E. Berki, M. Helenius, and S. Ovaska, ''Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: What do usability tests indicate?'' Behaviour Inf. Technol., vol. 33, no. 11, pp. 1136–1147, Nov. 2014.

[14] N. Samarasinghe and M. Mannan, ''On cloacking behaviours of malicious websites,'' Comput. Secur., vol. 101, pp. 102–114, Feb. 2021.

[15] L. Halgas, I. Agrafiotis, and J. R. C. Nurse, ''Catching the phish: Detecting phishing attacks using recurrent neural networks (RNNs),'' in Information Security Applications (Lecture Notes in Computer Science), vol. 11897. Cham, Switzerland: Springer, 2020, pp. 219–233.

[16] R. S. Rao and A. R. Pais, ''Jail-phish: An improved search engine based phishing detection system,'' Comput. Secur., vol. 83, pp. 246–267, Jun. 2019.

[17] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani, ''Systematization of knowledge (SoK): A systematic review of software based web phishing detection,'' IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2797–2819, 4th Quart., 2017.

[18] K. L. Chiew, E. H. Chang, C. Lin Tan, J. Abdullah, and K. S. C. Yong, ''Building standard offline anti-phishing dataset for benchmarking,'' Int. J. Eng. Technol., vol. 7, no. 4, pp. 7–14, 2018.

[19] H. Yuan, Z. Yang, X. Chen, Y. Li, and W. Liu, ''URL2 Vec: URL modeling with character embeddings for fast and accurate phishing website detection,'' in Proc. IEEE Int. Conf. Parallel Dis trib. Process. With Appl., Ubiquitous Comput. Commun., Big Data Cloud Comput., Social Comput. Netw., Sustain. Comput. Commun. (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Dec. 2018, pp. 265–272.

[20] M. Sánchez-Paniagua, E. Fidalgo, V. González-Castro, and E. Alegre, ''Impact of current phishing strategies in machine learning models for phishing detection,'' in Computational Intelligence in Security for Informa tion Systems Conference, vol. 12676. Cham, Switzerland: Springer, 2021, pp. 87–96.

[21] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, ''Machine learning based phishing detection from URLs,'' Expert Syst. Appl., vol. 117, pp. 345–357, Mar. 2019.

[22] Y. Cao, W. Han, and Y. Le, ''Anti-phishing based on automated individual white-list,'' in Proc. 4th ACM Workshop Digit. Identity Manage. (DIM), 2008, pp. 51–59.

[23] 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.

[24] A. K. Jain and B. B. Gupta, ''A novel approach to protect against phishing attacks at client side using auto-updated white-list,'' EURASIP J. Inf. Secur., vol. 2016, no. 1, pp. 1–11, Dec. 2016.

[25] E. Buber, B. Diri, and O. K. Sahingoz, ''NLP based phishing attack detection from URLs,'' in Proc. Int. Conf. Intell. Syst. Design Appl., vol. 736, 2018, pp. 608–618.

[26] A. K. Jain and B. B. Gupta, ''PHISH-SAFE: URL features-based phishing detection system using machine learning,'' in Advances in Intelligent Systems and Computing, vol. 729. Singapore: Springer, 2018, pp. 467–474.

[27] B. Banik and A. Sarma, ''Lexical feature based feature selection and phishing URL classification using machine learning techniques,'' in Proc. Int. Conf. Mach. Learn., Image Process., Netw. Secur. Data Sci., vol. 1241. Singapore: Springer, 2020, pp. 93–105.

[28] R. S. Rao and A. R. Pais, ''Detection of phishing websites using an efficient feature-based machine learning framework,'' Neural Comput. Appl., vol. 31, no. 8, pp. 3851–3873, Aug. 2019.

[29] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, ''A stacking model using URL and HTML features for phishing webpage detection,'' Future Gener. Comput. Syst., vol. 94, pp. 27–39, May 2019.

[30] 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 (WWW), 2007, pp. 639–648.

[31] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, ''CANTINA+: A feature rich machine learning framework for detecting phishing web sites,'' ACM Trans. Inf. Syst. Secur., vol. 14, no. 2, pp. 1–28, Sep. 2011.

[32] M. Moghimi and A. Y. Varjani, ''New rule-based phishing detection method,'' Expert Syst. Appl., vol. 53, pp. 231–242, Jul. 2016.

[33] V. I. Levenshtein, ''Binary codes capable of correcting deletions, insertions, and reversals,'' Sov. Phys.-Dokl., vol. 10, no. 8, pp. 707–710, 1966.

[34] M. A. Adebowale, K. T. Lwin, E. Sánchez, and M. A. Hossain, ''Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text,'' Expert Syst. Appl., vol. 115, pp. 300–313, Jan. 2019.

[35] M. Rami, M. Lee, and T. Fadi, ''UCI machine learning repository,'' Univ. Huddersfield, Huddersfield, U.K., Tech. Rep., 2015.

[36] L. Yang, J. Zhang, X. Wang, Z. Li, Z. Li, and Y. He, ''An improved ELM based and data preprocessing integrated approach for phishing detection considering comprehensive features,'' Expert Syst. Appl., vol. 165, Mar. 2021, Art. no. 113863.

[37] F. Sadique, R. Kaul, S. Badsha, and S. Sengupta, ''An automated framework for real-time phishing URL detection,'' in Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC), Jan. 2020, pp. 335–341.

[38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ''Distributed representations of words and phrases and their

compositionality,'' in Proc. Adv. Neural Inf. Process. Syst., vol. 26, no. 4, Dec. 2013, pp. 3111–3119.

[39] M. Somesha, A. R. Pais, R. S. Rao, and V. S. Rathour, ''Efficient deep learning techniques for the detection of phishing websites,'' Sadhan ¯ a¯, vol. 45, no. 1, pp. 1–18, Dec. 2020.

[40] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, ''An effective phishing detection model based on character level convolutional neural network from URL,'' Electronics, vol. 9, no. 9, pp. 1–24, 2020.

[41] A. Al-Alyan and S. Al-Ahmadi, ''Robust url phishing detection based on deep learning,'' KSII Trans. Internet Inf. Syst., vol. 14, no. 7, pp. 2752–2768, 2020.

[42] J. Zhao, N. Wang, Q. Ma, and Z. Cheng, ''Classifying malicious URLs using gated recurrent neural networks,'' in Proc. Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput., vol. 773, 2019, pp. 385–394.

[43] X. Zhang, J. Zhao, and Y. Lecun, ''Character-level convolutional networks for text classification,'' in Proc. Adv. Neural Inf. Process. Syst., Jan. 2015, pp. 649–657.

[44] Y. Kim, ''Convolutional neural networks for sentence classification,'' in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1746–1751.

[45] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, ''A new hybrid ensemble feature selection framework for machine learning-based phishing detection system,'' Inf. Sci., vol. 484, pp. 153–166, May 2019.

[46] R. S. Rao, T. Vaishnavi, and A. R. Pais, ''CatchPhish: Detection of phishing websites by inspecting URLs,'' J. Ambient Intell. Humanized Comput., vol. 11, no. 2, pp. 813–825, Feb. 2020.

[47] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, ''LightGBM: A highly efficient gradient boosting decision tree,'' in Proc. Adv. Neural Inf. Process. Syst., Dec. 2017, pp. 3147–3155.

[48] T. Chen and C. Guestrin, ''XGBoost: A scalable tree boosting system,'' in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vols. 13–17, 2016, pp. 785–794.

[49] A. Aizawa, ''An information-theoretic perspective of tf–idf measures,'' Inf. Process. Manage., vol. 39, no. 1, pp. 45–65, 2003.

[50] M. W. Al-Nabki, E. Fidalgo, E. Alegre, and R. Aláiz-Rodríguez, ''File name classification approach to identify child sexual abuse,'' in Proc. 9th Int. Conf. Pattern Recognit. Appl. Methods, 2020, pp. 228–234.

[51] C. Peersman, C. Schulze, A. Rashid, M. Brennan, and C. Fischer, ''ICOP: Live forensics to reveal previously unknown criminal media on P2P networks,'' Digit. Invest., vol. 18, pp. 50–64, Sep. 2016.

[52] W. H. E. Day and H. Edelsbrunner, ''Efficient algorithms for agglomerative hierarchical clustering methods,'' J. Classification, vol. 1, no. 1, pp. 7–24, 1984.

[53] J. Spaulding, S. Upadhyaya, and A. Mohaisen, ''The landscape of domain name typosquatting: Techniques and countermeasures,'' in Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES), Aug. 2016, pp. 284–289.Brown, S. L. (2019). "Phishing as a Lucrative Business: A Comprehensive Study." Cybercrime Research Quarterly, 14(1), 32-47.