PhishRepo: A Seamless Collection of Phishing Data to Fill a Research Gap in the Phishing Domain

Subhash Ariyadasa[1], Shantha Fernando[2] and Subha Fernando[3]
Department of Computational Mathematics, University of Moratuwa, Sri Lanka[1,3]
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka[2]
Department of Computer Science and Informatics, Uva Wellassa University, Sri Lanka[1]

***Abstract*—Machine learning-based anti-phishing solutions face various challenges in collecting diverse multi-modal phishing data. As a result, most previous works have trained with little or no multi-modal data, which opens several drawbacks. Therefore, this study aims to develop a phishing data repository to meet the diverse data needs of the anti-phishing domain. As a result, a gap-filling solution named PhishRepo was proposed as an online data repository that collects, verifies, disseminates, and archives phishing data. It includes innovative design aspects such as automated submission, deduplication filtering, automated verification, crowdsourcing-based human interaction, an objection reporting window, and target attack prevention techniques. Moreover, the deduplication filter, used for the first time in phishing data collection, significantly impacted the collection process. It eliminated the duplicate data, which causes one of the most common machine learning errors known as data leakage. In addition, PhishRepo enables researchers to apply modern machine learning techniques effectively and supports them by eliminating phishing data hassle. Therefore, more thoughtful use of PhishRepo will lead to effective anti-phishing solutions in the future, minimising the social engineering crime called phishing.**

***Keywords*—Cyberattack; crowdsourcing; internet security; phishing; machine learning; multi-modal data**

I. Introduction

Industry 4.0, or the fourth industrial revolution that marks the beginning of the imagination age, has opened various opportunities for human beings through automation and data exchange. However, it is a double-edged sword where criminals also optimise the revolution change to effectively operate their criminal activities on the Internet. Phishing is an illegal activity that relies on the Internet, which has gained a top rank in the cyber threat landscape [1]. It is a social engineering threat that damages Internet users illegally using digital assets-incredibly personal and confidential information [2]. Phishing is known as 'identity theft' because it impersonates one's identity in cyberspace for the phisher's benefit [3], [2].

The phishing threat first occurred in 1996 [2], and initially, online banking and e-commerce services were popular among phishers [4]. The direct or indirect financial gains motivate phishers in phishing, and fame and notoriety are also attractive [3]. Phishers are constantly moving with technology. Therefore, they are keen to experiment and improve attacking strategies in the phishing domain without failing in front of the available security countermeasures [5], [6]. The number of phishing attacks is still rising. Interestingly, the Anti-Phishing Working Group (APWG) has stated that phishing attacks had doubled in 2020, and in October 2020, only they have detected 225,304 unique phishing websites [7].

In phishing attacks, phishers commonly send an email to a user with an embedded link to redirect the user to a phishing site [5], [2]. This email often denotes a specific scenario like updating account details or security upgrades and creates a way to convince users to believe it [2]. The phishers recently used the Coronavirus pandemic (COVID-19) to raise phishing campaigns to fool Internet users [2]. However, by accepting, an unsuspicious user might click on the given link and move to the phishing website, which is very similar to the legitimate website, to feel more confident about his previous action [5], [2]. Then, the most dangerous thing happens in the process. By believing this is the legitimate website, the unsuspicious user enters his vital information to the phisher's website that impersonates the legitimate website. That information could be bank details, login credentials, a social security number, a credit card number, or other personal or confidential information [2]. However, this would be the main harvest of the phisher of this phishing process, and he might use it or sell it for his benefit.

According to the literature, 95% of phishing attacks were successed due to human errors [2]. Therefore, numerous solutions [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] were introduced in the last two decades to safeguard humans from this prevalent Internet threat by detecting phishing attacks. Those solutions could be mainly categorised into user education and software-based solutions [20]. Out of those categories, software-based solutions were more prominent in the past since user education is associated with a high cost and requires fundamental knowledge of computer security [2]. The software-based solutions also use different approaches when finding an effective anti-phishing solution [20]. Of those approaches, machine learning shows promising results due to its unique advantages like handling frequent data changes and automating the learning process [6]. However, machine learning studies in the phishing domain suffer from labelled phishing data [21], [22]. Therefore, researchers primarily work with their data [12], [15], [17], [16], [23] due to a lack of benchmark datasets available in the current anti-phishing domain and their limitations [22].

The current study mainly focuses on finding an effective solution to the difficulty of getting labelled data or training with a limited amount of features in the phishing domain. It is essential in the present context since advanced techniques like deep learning can work effectively with many high-dimensional data when identifying complex attacks like phishing [24]. Further, these labelled data may be more effective in retraining the trained machine learning models since phishing

attacks are rapidly changing over time [21], [22]. The proposed solution to fill the identified research gap in the phishing domain is an online repository that can constantly collect, verify, disseminate, and archive real-time phishing data. This solution allows automatic submission of phishing data by anti-phishing solutions and guarantees the diversity of data through different filters like deduplication.

Further, it effectively uses existing phishing verification systems and crowdsourcing techniques to review the labelling of those collected data. Moreover, it manages the essential aspects of the submitted phishing records to open data to the scientific community, especially for anti-phishers. The main contributions of this study are an online phishing data repository for collecting, validating, disseminating and archiving real-time phishing data; a large-scale, diverse phishing data in raw format for research purposes; and a set of design artefacts to have in a real-time phishing data repository.

This article aims to introduce the gap-filling solution that touches the data needs in the anti-phishing domain and demonstrates the effectiveness of the used architecture of PhishRepo in the problem domain. The other sections of this paper include Section II - a high-level description of the problem domain; Section III - a review of the literature; Section IV - the architecture of the proposed solution; Section V - experiments were performed on the collected data to demonstrate the diversity and effectiveness of the machine learning process; Section VI - a discussion of the significance and usage of PhishRepo, and finally, Section VII - the concluding remarks of the study.

II. Problem Definition

Machine learning-based anti-phishing solutions mainly have two steps [17]. First, the features required to detect phishing attacks are extracted and then a machine learning model is trained using the extracted features. The feature extraction happens based on multiple information sources available on a website. The Uniform Resource Locator (URL) of a website is the popular source for many of the recent anti-phishing solutions [12], [15], [25], [23], and third party-based features like Alexa ranking and age of the domain are also used in different solutions [26], [27], [28], [29]. The website content, either human-readable or markup content, refers to HTML content, another vital source for extracting features. It has been used in many recent studies [14], [16], [17], [18], [19] to extract different features for the learning model. Further, several studies [13] already used captured images of the web page (i.e., a screenshot) when training machine learning-based anti-phishing solutions.

Supervised learning is the dominant practice with many existing anti-phishing solutions in the machine learning area [12], [14], [15], [17]. Therefore, the labelled data is essential for training the learning model. However, constructing a large-scale, diverse phishing dataset effective in training is impossible in one night since the phishing websites are short-lived [30], [31]. Therefore, it should be a continuous process and take time. However, phishing verification systems such as PhishTank (https://phishtank.org/) and OpenPhish (https://openphish.com/) collect many phishing URLs [30], including optional information like the screenshot of the website page and network information (i.e., WHOIS information). Although these systems contain the URLs, those do not include all information sources required to extract the most recently exercised feature vectors directly [22]. It is a downside of these verification systems, and it negatively impacts research since the researchers need a systematic way to collect data in the initial step of their methodology. Since the data collection takes much time, many machine learning-based anti-phishing studies used less data during the training phase [28], [32], [33]. For example, [32] used 1,428 phishing data, and [28] used 2,000 phishing data during their experiments. However, some of the accessed solutions that used more data in model training are URL based solutions, and those may not be effective due to the challenges that exist only with URL based information [21]. Therefore, multi-modal features, marked as effective in phishing detection due to the representation of many phishing attack characteristics [34], are essential in the present phishing detection context. Free public access to such data sources is necessary for better detection solutions in future.

Moreover, the literature has shown that the researchers use old datasets due to the lack of new public datasets [22]. It results in inept learning models on recent phishing attacks [22]. These factors highlight the importance of an organised way of acquiring the latest multi-modal feature enabled diverse phishing data for future phishing detection. Therefore, the difficulty of getting labelled data or training with a limited number of features or data has become a significant problem in the machine learning-based phishing detection area that should be resolved to expect promising results in future research [21]. This study will resolve the identified issues by answering two questions: how can a phishing data repository be made to support anti-phishing research effectively? and what are the most effective design strategies that could be used in a real-time phishing data repository to collect, verify and disseminate large-scale, diverse phishing data?

III. Related Literature

As highlighted in the problem definition, the difficulty of getting the latest, labelled phishing data with multi-modal features is a significant research challenge in machine learning-based phishing detection. This challenge could be overviewed closely by the three most related topics to the current study: data collection for phishing websites detection, feature selection for phishing websites detection, and data labelling in machine learning.

A. Data Collection for Phishing Websites Detection

Phishing techniques are constantly evolving due to technological improvements, enhanced security countermeasures and educated public [2]. Early days, phishers used untargeted attacks, and unsuspicious users were caught [2]. However, now phishers are more into target attacks, and techniques like spear-phishing are more prominent in the phishing domain [2]. The literature highlighted that the success rate of untargeted phishing attacks is less than 5%, while 19% of target attacks like spear-phishing get success [6]. However, when the phishing threat grew, many different parties like brand owners, researchers, and law enforcement were interested in these attacks from different perspectives [35]. Therefore, numerous organisations like APWG, Phishing Incident Response Team,

Phishing Report Network, and Digital PhishNet started to collect phishing attack-related data, resulting in different levels of data collection [36]. Further, an organisation like APWG mainly depends on the public, anti-phishing working groups, Internet service providers and brand owners when collecting phishing data [36].

In contrast, the Phisherman project [36], [35] addressed this phishing data collection process differently. It changed the present way of collecting phishing data and introduced a web-based system to collect, validate, disseminate and archive real-time phishing data. It is a global data collection system and fulfils three basic requirements: submitting suspicion records, saving the records for future use and outputting historical phishing data to interested parties. Phisherman has used an automated phishing records verification process, and the submitted records are verified in two steps. However, the first step is only for the submissions collected from individuals and high-volume spam feeds. The important feature in Phisherman in the current study's perspective is the dissemination of the collected data. Phisherman supports the data distribution in two ways: subscription and queries. However, these options allow downloading a blocklist or a full incident report in XML format.

PhishTank is one of the favourites to collect phishing data by many anti-phishing tool introducers [15], [34], [33], [23]. PhishTank was launched in 2006, and it is a community-based phishing verification system [30]. The PhishTank facilitates submitting phishing URLs, and the community votes those submissions to be a phishing website or marked as a legitimate website. However, when looking at those studies, the researchers used the PhishTank only to get phishing URLs and have not been used to extract other information sources like the screenshot of the phishing website and WHOIS information present in some of the submitted phishing records. The data distribution strategy used by the PhishTank might be the closest reason for such a trend.

OpenPhish is another phishing verification system that collects phishing data via an autonomous phishing detection algorithm shaped through research. It is also popular among anti-phishers [30], and it distributes the collected data like URL, target brand and screenshot to interested parties. However, OpenPhish is not for free, and a free account gets only the phishing URLs, which also gets in every twelve-hour frequency.

The UCI Phishing dataset available in the University of California - Irvine's (UCI) repository is also popular among researchers in the phishing domain [31]. However, it is an old dataset with limited data (i.e., 11,055 maximum). Further, it has a preprocessed set of features and bounds the research scope to those features. It is one of the main drawbacks of this dataset, and the heuristics used to preprocess those data [37] are also not examined in the current environment. Similarly, [22] presented high quality, a diverse phishing data source for benchmarking purposes, and one output of their study is implementing a benchmarking framework called PhishBench. The dataset was constructed using the sources like PhishTank, OpenPhish, and APWG and a systematic approach was undertaken when collecting data. However, it needs such an approach again when collecting new data, which may be costly and time taken.

Furthermore, Web2Vec [19] and PhishPedia [38] are another two datasets collected with the support of PhishTank and OpenPhish. PhishPedia collected phishing records from OpenPhish using a premium account to get additional information like target brands. However, these datasets also contain old data compared to today and since these studies are not focused on updating these datasets, implementing anti-phishing tools to detect the latest phishing attacks is problematic.

Phisherman project is the only landmark for a deliberate phishing data collection and dissemination approach. However, Phisherman is not publicly available [39]. Therefore, it is not a solution for the identified data collection problem. The solutions like PhishTank and OpenPhish have different intentions, such as maintaining blocklists and identifying target brands. Those data collections are more into URL related information extraction in phishing detection, therefore not effective in the data collection problem mentioned in this study. The individual data sources are an excellent approach to donating phishing data to others; however, the relevancy depends on the frequent update and the ability to support multi-modal features in the present machine learning-based anti-phishing domain.

B. Feature Selection for Phishing Websites Detection

Feature selection is essential in phishing detection research since it impacts detection accuracy [6], [22]. The researchers in the literature introduce different feature sets that represent the essential information sources that need to include in a dataset. A more complex categorisation of phishing features is found in [22]. They used more than 250 phishing detection-based studies and divided the phishing features mainly into two classes: URL and website. Again, those two classes divide into lexical, network and script level features. Then these features were furthermore analysed based on the format and categorised into three. 1). Syntactic - syntactic correctness (i.e., port number and Term frequency-inverse document frequency referred to as TF-IDF), 2). Semantic - the meaning and interpretation of the content (i.e., presence of the target brand in URL and web page), and 3). Pragmatic – the features do not directly relate to syntax or meaning (i.e., backlisted words in a URL, WHOIS information, and script loading time).

In a similar study, phishing features were primarily categorised into four feature sets [6]. 1). URL-based lexical features like the length of the URL and the presence of the HTTPS protocol, 2). URL-based host features like WHOIS information, 3). web page content features like page rank, hyperlinks and forms in the HTML content, and 4). visual similarity-based elements like images and colours. Further, [14] used only URL and web page content features in their study, and HTMLPhish [17] is a particular case that used superior web page content features in phishing detection. Furthermore, [40] introduced another set of features in their research on machine learning-based phishing attacks. In that, they have mentioned four main groups of features, namely, URL-based features (i.e., number of subdomains, length, and number of digits), domain-based features (i.e., age of the domain, and whether it is blocked in reputed services), page-based features (i.e., page rank, and Alexa rank) and content-based features (i.e., page title, body text, a web page screenshot, and images).

After analysing the available feature sets in explored literature, it is clear that the URL and the web page are the most

important information sources to extract different features for model training. Although a website could be callable if the URL is saved in general, the phishing web pages cannot be recovered only from the URL since those are short-lived [30], [31]. Therefore, the instant saving of the phishing page and relevant resources like the web page screenshot, images, CSS, and JavaScript when the attack is active and online is essential for future use [22].

Moreover, the screenshot of the web page is an essential feature to consider in the machine learning-based visual similarity area [13]. Therefore, the study identified three primary sources for feature extraction for machine learning-based phishing detection. These are URL, web page, and third-party services. Further, those information sources are essential to consider when constructing an adequate dataset for future research since it supports the extraction of all the required features from one dataset.

C. Data Labelling in Machine Learning

Machine learning-based anti-phishing solutions are more toward the supervised learning paradigm. Therefore, labelled data is essential for model training [21], and expert labelling is a popular approach when labelling data in machine learning [41]. However, expert-based labelling is often costly and time-consuming since modern machine learning needs large-scale datasets [41]. Due to the limitations of the expert approach, crowdsourcing has become a widespread technique in data collection [42], [41], [43]. Crowdsourcing is based on collective intelligence, which beliefs together is better than a single entity [44]. It has advantages like low cost, fast labelling and diverse opinions than the expert approach [43]. However, the main drawback is getting high-quality labelled data [41], [43]. Factors like the poor commitment of workers, uncertainty in the tasks, prior knowledge of the given task, and novice workers are some of the reasons for imperfect quality labels in crowdsource based data labelling [41].

However, the quality of labels in crowdsourcing could be improved through several techniques discussed in the literature. Those are pre-training [41], task pricing [41], [43], calculation of labelling quality of workers [45], [46], and peer review of the crowd worker work [47]. As mentioned in [46], identifying incorrect labelling data points is not sufficient in crowdsourcing since the labelling quality of the workers also matters. In another study, [42] proposed a relabeling strategy called absolute cumulative majority relabeling (ACMR). It allows relabeling of the same data point multiple times. It uses a voting mechanism to select majority voting, and if a label achieves more than 50% voting, it sets that as the correct label for that data point. However, if none of the labels could earn more than 50% are discarded in the ACMR strategy. Revolt [41] is another solution that uses crowdsourcing when collecting data for machine learning tasks. In revolt, the dataset is divided into multiple batches for the crowd workers' easiness, and groups collectively contribute to each batch. From a different perspective, [48] studied the effect of cognitive biases in crowdsourcing. The study identified that the anchoring, bandwagon, and decoy effects occur in crowdsourcing, and an experiment has shown that a 28% accuracy loss was recorded due to the anchoring effect.

As identified, crowdsourcing is a practical approach to phishing data labelling. Further, the quality of the workers needs to be evaluated, and peer review of the workers' work is essential to maintain the quality. ACMR is a better strategy for phishing labelling since it allows adding many labels to a single record, and a majority voting technique is used when selecting an appropriate label. Multiple batches in the labelling process are also aligned with the current study since it reduces the overhead of seeing more labelling tasks simultaneously. Further, avoiding the dependency on one information or specific people is vital in crowdsourcing-based labelling and getting a quick explanation about the submitted label, as Revolt [41] proposed, is essential in current work to avoid doubtful labels. However, the task pricing and pre-training are not applicable here since the cost is incurred with those techniques, and the proposed solution is freeware.

IV. PhishRepo

PhishRepo is an online phishing data repository for collecting, validating, disseminating and archiving real-time phishing data. The researchers and other interested parties would use it for their customised needs associated with data. PhishRepo architecture consists of three primary modules: input, verification, and distribution.

PhishRepo is a data collection solution for industry 4.0. Therefore, it introduces a safe architecture to integrate with autonomous anti-phishing tools to submit their phishing data directly to the repository for others' use. Therefore, this solution can provide effective results by collaborating with existing anti-phishing solutions. The other speciality of the repository is the type of data it collects. The repository is designed to collect most information sources, namely URLs, web pages and third-party service information relevant to a submission. However, when collecting third-party service information, the repository limits the free account level and possible third-party details are stored in the phishing records. Following is an in-depth explanation of PhishRepo modules.

A. Input Module

The primary task of the input module is the collection of phishing data which includes collecting phishing URLs from outside, acquiring the relevant information sources and archiving those in the repository, as shown in Fig. 1. Phishing data collection seems challenging since 63% of the phishing campaigns last within the first two hours [20]. It is not challenging to archive only the URLs [22]. However, PhishRepo is responsible for collecting URLs, web pages and possible third-party information sources for each inputted phishing record. Therefore, the detection time and reporting time are crucial in data collection. Thus, as a specific design consideration, PhishRepo allows external anti-phishing tools to submit their findings (i.e., phishing URLs) automatically. Then, it minimises the difference between detection and reporting time. It also helps collect the most active and online phishing URLs to effectively acquire the required information sources. Although the manual submission exists in PhishRepo, as in Fig. 1, PhishRepo encourages automated submissions to get the most active and online phishing URLs in the data collection process.

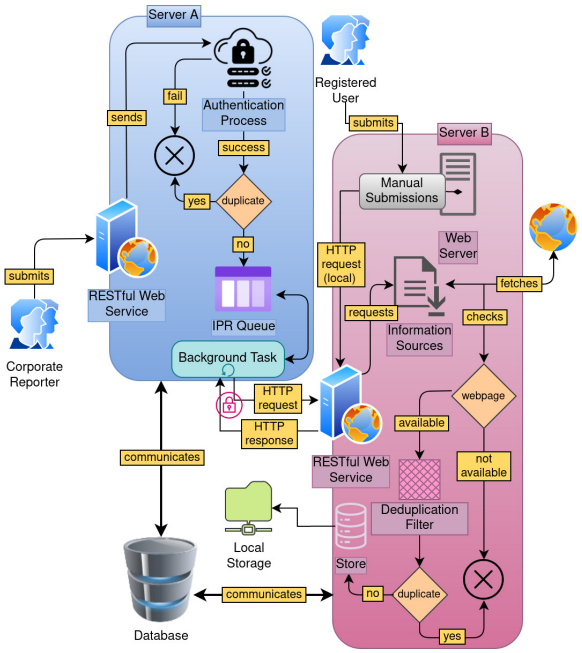

Fig. 1. Workflow Diagram of the Input Module

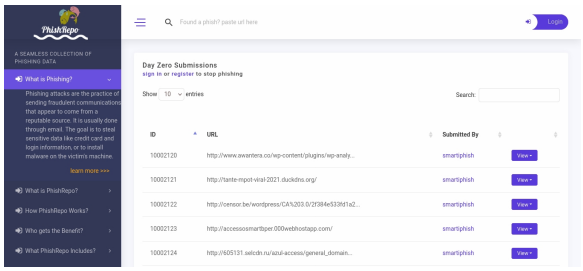

Fig. 2. The Landing Page of PhishRepo

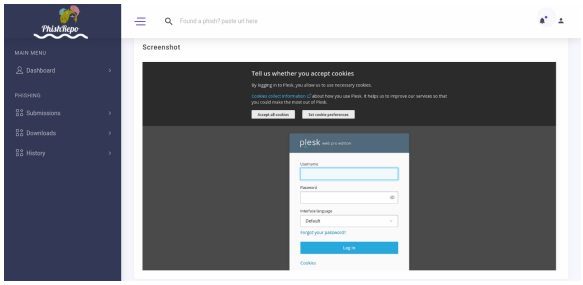

Fig. 3. A Captured Screenshot of a Visible Web Page

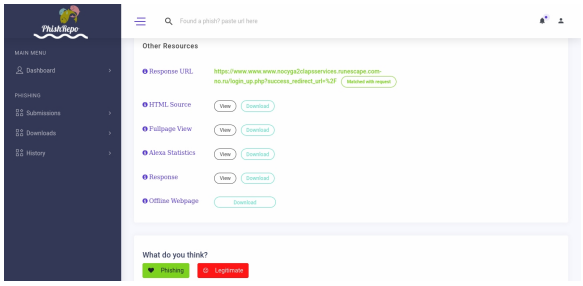

Fig. 4. PhishRepo Displays Additional Information Sources under the Other Resources Section. The Registered user can Download or View that Information Relevant to a Submission.

PhishRepo's input module consists of five components: authentication, accumulation, deduplication, targeted attack prevention (TAP), and manual submission. The followings discuss these components in detail.

1) Authentication: PhishRepo has five users: administrator, editor, reporter, beneficiary, and guest. The administrator is the primary account holder with full privileges, and the editor is responsible for verifying the submitted phishing records. There are three levels in the editor account: newbie, competent, and expert. These levels are achieved by each user based on their performance. However, the expert editor is a chief editor type in PhishRepo and is responsible for the final decision of incorrect submission. The expert editor can modify the records if required and report phishing instantly. Therefore, the automatic account upgrade is turned off when upgrading a competent account, and the administrator is involved in forming an expert editor based on the recommendations provided by the system. Other levels are automatically upgraded based on the points earned through the correct marking of records. Next, the reporter can submit phishing records in PhishRepo, either manually or automatically. The beneficiary user connects with PhishRepo to download phishing data only. Since the data distribution process needs a registered user type, the beneficiary type is added to the proposed solution. Consequently, the guest user can only view the public information related to phishing records and use search facilities to search available phishing web pages in the repository. Fig. 2 shows the landing page of PhishRepo that is visible to all the accounts mentioned above.

A valid email is required when creating a PhishRepo account, and the administrator is responsible for the final confirmation of a new account. Since a human user or an anti-phishing tool could become a reporter, the reporter account has two subscriptions: individual or corporate. The anti-phishing tools always act as a corporate reporter, and those accounts own an application key for authentication. Further, the corporate account has an optional field called 'return address', which sends daily updates about the submissions. Section IV-C discusses this option in detail.

PhishRepo uses two authentication methods. 1). A login form 2). An application key. The first method is the standard approach in most scenarios, and it uses a username and a password for the verification process. The second one is only applicable to corporate accounts like anti-phishing tools and is only used when an automatic submission is processed with PhishRepo. In that case, the corporate account user must follow the predefined data format to submit a phishing record.

However, one request can submit only one phishing record, and multiple requests are required for numerous submissions. After the authentication is done, it checks for duplication within the repository through URL string matching, and if no duplication is found, the URL is added to the Initial Phishing Records (IPR) queue alongside the submitter information. In PhishRepo, the IPR queue is a double-ended queue (deque) that uses first-in, first-out (FIFO) logic.

2) Accumulation: This component is crucial in PhishRepo since it is responsible for the data collection. It starts once a request comes from Server A (Fig. 1). First, it tries to download the complete web page of the submitted URL. If the download process fails due to any exception, the accumulation component skips that URL and moves to the next since the web page is a vital and mandatory information source in PhishRepo. Meanwhile, the data collection process captured the response details and saved them under the record because, in some cases, there can be some mismatches in request and response details which may be helpful in the verification process. After the web page download is done, the screenshot of the web page is captured in full view and visible level (Fig. 3). Then, the possible third-party information is downloaded. This information is kept as additional information (Fig. 4) in PhishRepo and is not mandatory due to the service limitations that exist in the third-party services.

3) Deduplication: Deduplication is crucial in PhishRepo that eliminates redundant data such as duplicate phishing pages with the same target. At the beginning of the input module, the authentication component takes the necessary actions to eliminate duplications based on the URL. However, the different URLs do not guarantee that duplications could be avoided in a phishing dataset since most phishing pages are created using phishing kits [4] and released to the public. Therefore, a dataset could have different URLs for similar page structures, as shown in Fig. 5 and make duplicates to machine learning processes that cause data leakage at the end.

In that context, PhishRepo's deduplication component is responsible for eliminating such duplicates with the support of the Perceptual Hashing (pHash) technique [49] that was exercised in the literature for similar scenarios [50]. The component handles the elimination of duplicates as an inline task that affects before saving the accumulated phishing records to the local storage or database, as shown in Fig. 1. Therefore, none of the submissions is identified as duplicate records kept in PhishRepo's repository.

The deduplication process depends on the visual level screenshot downloaded during the accumulation process. It uses the pHash technique to determine the similarity of two phishing pages, and PhishRepo maintains a list of hashes computed for the saved records. During the filtering process, a perceptual hash value is first generated for the newly captured screenshot and compared with the already stored hash values to check whether the new one is a duplicate of an already saved web page. The comparison is made through the distance factor (d) calculated using the two instances' hash values. However, if an exact matching is found ($d = 0$), one of the records will be removed from the repository to address the redundancy factor. In that case, which one to eliminate is dependent on the PhishRepo's setting called 'Dedup Action'. The Dedup Action has two values: new and old. If the value 'new' is enabled, the component will save the new record and remove the old one from the repository, and in the other way, it is not going to save the new record, and the old will remain. The setting is introduced just to have a flexible deduplication process within the PhishRepo, and the administrator is responsible for activating a specific Dedup Action to have a diverse phishing data collection.

Since the deduplication process depends entirely on the screenshot captured during the accumulation process, PhishRepo is configured to check for near-duplicates for a given period to eliminate any page loading issues during screen capturing. However, it is not practical to check near-duplicates of a new screenshot with all the available records in PhishRepo since the comparison process takes time. Therefore, the deduplication component checks for near-duplicates only for the last three days since most phishing attacks end within three days [51], [31]. Then, theoretically, PhishRepo assumes that the near-duplicates that may exist out of the three days are different phishing attacks.

However, there should be an optimal distance threshold (d_α) for a meaningful selection for the near-duplicates. Therefore, d_α is selected based on 1,000 random samples from an older version of PhishRepo that does not include the deduplication component. Then, pHash values of the screenshots available in the sample were computed, and each pair's d values were calculated. After that, a manual investigation was carried out to examine the accuracy and noted that the accuracy of the similarity of a pair had been decreased drastically when d became more than 10, as shown in Fig. 6. Therefore, d_α was selected as 10, and $0 < d < 10$ are considered near-duplicates in PhishRepo. However, the near-duplicates elimination process does not affect the Dedup Action setting, and if a near duplication is found, the new submission will be entirely discarded from PhishRepo to maintain a diverse phishing data repository.

4) Targeted Attack Prevention (TAP): The main intention of PhishRepo is to collect phishing data to strengthen future anti-phishing tools against phishing attacks. That intention creates opponents (i.e., phishers) to PhishRepo. Therefore, PhishRepo may become a victim of some targeted attacks to disrupt the data collection process of the repository. The denial of service (DoS) attack is a possible threat [36], and there can be other specific attacks like false data injections. However, the network architecture presented in Fig. 1 strengthens the network level protection to a certain extent, and the implemented TAP component provides application-level protection to PhishRepo. The TAP component uses four strategies to have additional application security other than the standard security practices.

- Application key-based authentication – only the users with an application key can submit records automatically
- High-volume restriction – limit number of submissions from one corporate account
- Maximum IPR queue length – limit the number of request processes by the accumulation component
- False ban – Bans the reporters who have falsely recorded submission trend

As described in the authentication component, all the reporters should have an application key when submitting a phishing record automatically to PhishRepo. It limits the attacking trend since the attacker must obtain a valid application key to enter the system. If an attacker comes with a proper application key, then the subsequent countermeasures try to minimise the impact of those attacks. First, a high-volume restriction policy is implemented in PhishRepo to submit only

(a) https://cbahospitalar.com.br/002WG/well-fargo-RD528-detail/

(b) https://mail.cbahospitalar.com.br/002WG/well-fargo-RD528-detail/

Fig. 5. Different URL Examples for the Same Phishing Target

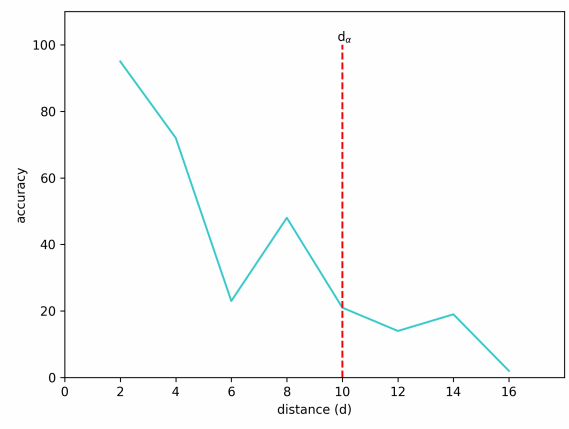

Fig. 6. Estimation of Distance threshold d for Near-Duplicate Detection

a limited number of records for a given period from one reporter. Since the chance of seeing a phishing URL is lower than a legitimate one [22], there is no possibility of submitting many records within a given period since PhishRepo requires real-time submission and discourages batch processing. As previously mentioned, the submissions come as requests, and one request carries only one submission. Therefore, the number of submissions equals the number of requests from a reporter side. Then, if he exceeds the limit (i.e., ten requests per minute), his account is automatically banned for five minutes, and if it is frequent, the account is blocked permanently by the TAP. Further, TAP is responsible for reporting abnormal behaviour to the administrator, and the administrator can take necessary actions on those.

In a specific situation, if an attacker bypassed both the mentioned countermeasures, the maximum IPR queue length is used to maintain a fixed-length queue to avoid overloads of the memory. Then there may be no performance hits, and PhishRepo may function without interruption. However, since the accumulation component takes URLs from the IPR queue, some submitted records may be removed without processing in a special attack. Although it seems wasted, PhishRepo does not intend to collect all the submitted phishing records and work only with the possible submissions when expanding the available phishing records.

In addition, the false ban strategy is another security consideration used in PhishRepo to avoid wrong data injections. The incorrect data may damage the proposed solution's trustworthiness and waste many resources. Therefore, PhishRepo is used to verify whether the collected phishing records are correct. That process is discussed in detail under Section IV-B. This false ban strategy checks the validity of the submitted phishing records week by week for each reporter and calculates an accuracy percentage. If the rate is less than a defined threshold value for an account, that account is suspended automatically and reported to the administrator for further actions.

5) Manual Submission: This component is implemented to cater to the generic manual submission needs. However, manual submission is not entertained in PhishRepo since real-time phishing records are required to store correct information sources. In some cases, a manual submission may be a particular need. Therefore, this component is added to the PhishRepo. Manual submission is a simple component, and the primary responsibility is to collect the phishing URL from the interface and send it to the accumulation module to process it further.

B. Verification Module

PhishRepo verifies all the submitted phishing records regardless of the source it gets. It is a two steps process named alpha verification and beta verification. Fig. 7 represents the workflow of the verification process, and the main two steps are explained in detail in the following sub-sections.

1) Alpha Verification: It is the first verification done by the PhishRepo after a record is successfully added to the repository. This alpha verification is done using two popular phishing verification solutions in the current context: PhishTank and Google Safe Browsing (GSB) [30]. These solutions have free Application Programming Interface (API) support to get information about phishing sites. Therefore, the collected URLs are submitted to both these services. If one or both marked the submissions as phishing, the verification module flags the relevant records as 'verified'. When the verification solutions do not provide any result for a specific submission, in that case, that record is marked with a 'processed' flag. It indicates that the alpha verification is processed on the record

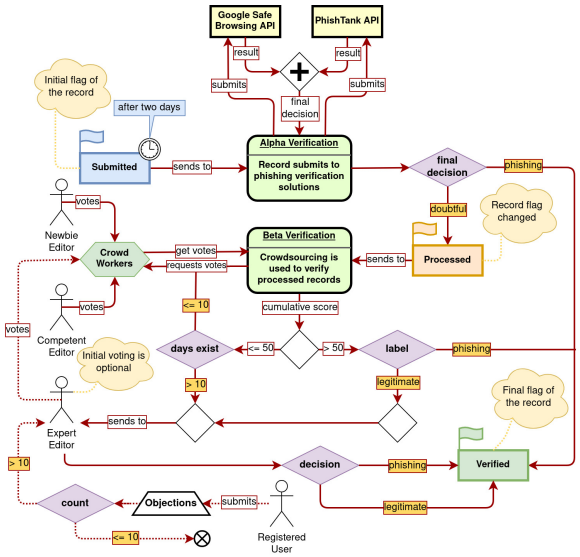

Fig. 7. PhishRepo's Record Verification Process

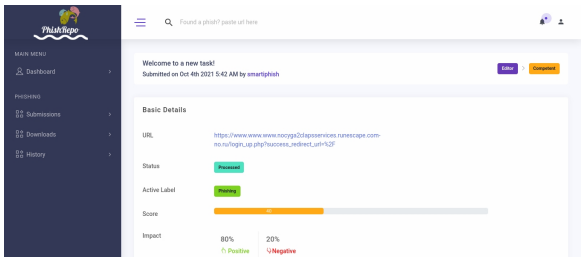

Fig. 8. PhishRepo has Displayed Basic Information to an Editor, such as the URL, Current Status, Active Label, Score, and the Impact of the Submission.

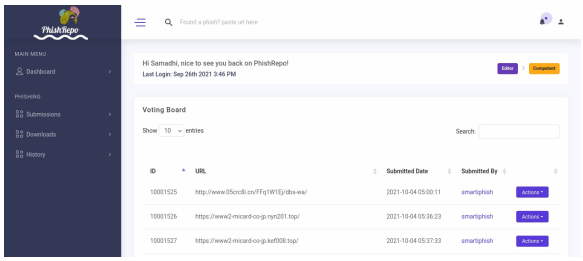

Fig. 9. Editor's Voting Board

but is yet to be finalized. However, this alpha verification is not executed immediately after adding a new phishing record to the repository. It waits 24 hours because phishing URLs are not added instantly to the blocklists, and 47%-83% are added after 12 hours [20].

2) Beta Verification: Beta verification manages only the 'processed' flagged phishing records–i.e. the records which have an unsuccessful alpha verification attempt. As the name implies, beta uses a crowd to collect opinions about the submitted record. Therefore, it is a crowdsourcing approach. As explained in Section III, crowdsourcing could be a tool to gather collective intelligence for a specific task like data labelling. Therefore, PhishRepo strategically uses this crowdsourcing technique to verify 'processed' flagged phishing records in beta verification. The editor is the leading actor in the beta verification. Out of the available editors, the expert editor is the final decision maker of an incorrect submission and gets a record if it gets majority voting as legitimate by a newbie or competent editor, or the record passed ten days from submission. However, the newbie, competent and expert levels have different impact points in the voting process. For example, suppose there is an incorrect submission. If a newbie marked it as legitimate, it has a 10% impact. If a competent level user is marked, the impact is 25%. However, the expert editor is the chief editor in PhishRepo; thus, he receives a 100% impact point.

Beta verification is done through a voting scheme. As seen in Fig. 8, each 'processed' flagged record appears to the editors to vote as phishing or legitimate (Fig. 4). Then, the editor can select either phishing or legitimate to award points for the verification process. For example, if a newbie selects one record as phishing, then the record gets 10 points to the phishing label. If a competent level user selects the same, the record receives 25 points. However, based on the ACMR strategy, the record needs to collect more than 50 points on the phishing label to become a verified record in PhishRepo.

In PhishRepo, the voting is both positive and negative. After a newbie marked a record as phishing, suppose that the same is recorded as legitimate by a competent user in the scenario mentioned above. Then, the PhishRepo checks the voting trend in that record, and since the voting trend is now on the phishing label, the record gets the new mark of -25, and the final score becomes 15 on the legitimate side. However, as a general rule in PhishRepo, if a record contains less than 50 points either in phishing or legitimate remains as a 'processed' flagged record and any record with more than 50 points on the phishing label upgrade verified state automatically. Further, if a record achieves more than 50 points on the legitimate side, the record is sent to an expert editor for review and is responsible for the final decision. However, after a record comes to a verified state, PhishRepo welcomes objections through the objection reporting module in the proposed solution. Therefore, all the user accounts except guests could raise objections to a verified phishing record, and if there are several objections, the expert editor reviews the record again. The expert editor could disable future objections at the review time to avoid misuse of the objection process. However, if a record exists in the repository for more than ten days without being verified, it appears in the expert editors' voting board to get their attention. Fig. 9 shows the voting board interface of an editor.

Further, PhishRepo uses unique design considerations to avoid cognitive bias throughout the beta verification. That hides the scoring history from all the editor levels and displays only the final score through a progress bar. Then the editor does not get to know any past editors. However, the expert editor gets an additional detail called impact, which describes how many negative (i.e., legitimate) and positive (i.e., phishing) votes were earned by a record when it comes to the current state. Further, PhishRepo always receives a brief explanation about the submitted label to avoid doubtful labels by asking a simple question from the editor such as *Can you find the targeted website?*, and *Can you find this website in the Google*

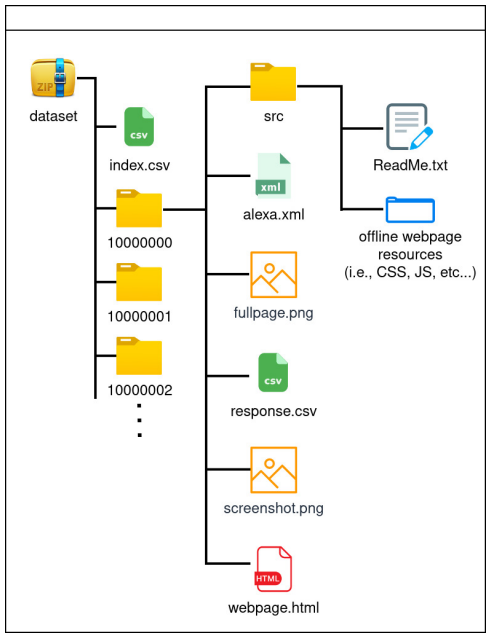

Fig. 10. The Hierarchical Structure of the Zip File

search engine?. The ultimate goal of these questions is to give a second chance to the editor to think about the decision before submitting it to PhishRepo.

C. Distribution Module

Data distribution is the primary goal of PhishRepo. Therefore, the distribution module plays a vital role in the proposed solution. However, the distribution module is only available for registered accounts, and as mentioned before beneficiary user type is specifically designed to support this process. PhishRepo provides several diverse information sources in raw format for each download. Those are the HTML page, visible level and full-page screenshots, response return for the made request, Alexa statistics and offline web page. Further, PhishRepo facilitates the data distribution in two ways: user queries and reporter subscriptions.

1) User Queries: The registered users can log in to PhishRepo and query for phishing data. PhishRepo requests data duration and the information sources required by the user. The full dataset could be downloaded from a separate menu item, and it includes all verified phishing records available in PhishRepo. The user query is either for total data or selected data; the final output of the download process is a zip file that includes an index file for easy navigation. Then it is available for users to download. Fig. 10 shows the hierarchical structure of the downloadable zip file.

However, there could be situations like the information sources missed in some folders due to exceptions in the accumulation process. In that case, the index file is vital to find out what is missing since it has columns like an eight-digit number that holds the mapping between the index file and the dataset folders, the request URL, the response URL, the data collection date, and attributes indicating the presence of the visible level screenshot, full-page view, Alexa statistic file, the offline web page, and response header file.

TABLE I. DETAILS OF THE USED PHISHING DATASETS

Dataset Name	Number of Data	
	Phishing	*Legitimate*
PhishRepo	5,275	0
Ex-PhishRepo	2,029	0
Web2Vec [19]	21,296	24,800
PhishPedia [38]	29,048	22,252

2) Reporter Subscription: The reporter subscription module's primary purpose is to give corporate reporters a unique benefit for their vital contributions. So far, it is clear that the corporate reporter is the key user who runs the proposed solution for the long term. Therefore, the PhishRepo is designed to automatically send feedback on what they have reported to the repository to encourage and admire the corporate reporter. As mentioned earlier, the reporter account has a particular field called 'return address'. PhishRepo uses this field value and sends daily feedback to the corporate reporter for their submission. However, feedback for a particular record waits until PhishRepo confirms the records label and keeps track of the sent feedback to avoid any duplication of feedback. The feedback report is sent as a CSV file in PhishRepo.

V. EXPERIMENTS AND RESULTS

The study used two main experiments to evaluate PhishRepo from diversity and its effectiveness in machine learning-based anti-phishing studies. Four primary datasets were used in those experiments, including two recently used public phishing datasets that include the URL, HTML page and screenshot of the relevant phishing instances.

A. Datasets

The four datasets used in the experiments are Web2Vec, PhishPedia, Ex-PhishRepo and PhishRepo. Table I presents the details of those datasets.

The PhishRepo and Ex-PhishRepo datasets were downloaded from the online phishing data repository presented in this paper. However, the Ex-PhishRepo dataset was downloaded before the deduplication component (Section IV-A3) was introduced to the proposed solution. Therefore, duplicate or near-duplicate phishing web pages were not filtered in the Ex-PhishRepo data. The PhishRepo dataset was downloaded after the deduplication filter was influential in the presented work. Therefore, the impact of the filter should be visible in the PhishRepo dataset. The initial level phishing URLs for both the Ex-PhishRepo and PhishRepo datasets were downloaded from PhishTank and OpenPhish. Therefore, the phishing data available in both datasets were valid phishing instances, and both datasets were available online [52] for further reference. Moreover, the Ex-PhishRepo dataset data were collected by the PhishRepo system from 29 September 2021 to 17 October 2021, and the PhishRepo dataset data were collected from 23 October 2021 to 02 February 2022.

The Web2Vec dataset is an online phishing dataset (https://github.com/Hanjingzhou/Web2vec) recently used by [19] when developing their anti-phishing solution. The dataset contained 21,303 phishing instances from PhishTank from September 2019 to November 2019 [19]. However, the current

work only could use 21,296 instances out of the total phishing instance of the dataset due to some data extraction issues. Similarly, the PhishPedia dataset is also a recently used phishing dataset by [38]. It contained 29,496 phishing web pages, and OpenPhish's premium account was used when downloading those data. The authors have publicly shared the dataset and are available online (https://drive.google.com/file/d/12ypEMPRQ43zGRqHGut0Esq2z5en0DH4g/view?usp=sharing) for anyone to download. The study could only use 29,048 phishing items from the PhishPedia phishing dataset since few data items reported some issues during the extraction.

Since the proposed PhishRepo solution distributes only phishing attack-related data, the PhishRepo and Ex-PhishRepo datasets do not contain legitimate data, as shown in Table I. However, recent anti-phishing studies already used the Web2Vec and PhishPedia datasets. Therefore, both these datasets were attached legitimate data used by those studies, and Alexa was the source for legitimate data in both cases.

B. Diversity of PhishRepo

The main objective of PhishRepo is to provide a diverse phishing dataset for machine learning-based anti-phishing studies. Therefore, PhishRepo output was evaluated from different perspectives to check whether the proposed solution achieved a diverse dataset. However, there is no widely accepted method to check the diversity of a dataset [22], but [22] have proposed two main criteria to use when measuring the diversity of a phishing dataset. Those are the number of different domains and the number of different top-level domains (TLDs). However, literature has shown that the HTTPS based phishing attacks and URL character length distribution are also essential to consider in the current phishing attack nature to have unbiased, accurate model training at the end [53], [7].

Even though these four could be taken as standard criteria to check the diversity, none of the studies in the literature considered the tendency of data leakage in a dataset that has been discussed in Section IV-A3. However, the current study has identified it as an essential factor and used it as the fifth criterion to check the diversity of the PhishRepo dataset. Although Table I presents four datasets, the PhishRepo and Ex-PhishRepo datasets had the exact behaviour in one to four experiments since the deduplication filter was the only noticeable difference in those two datasets. Therefore, the Ex-PhishRepo dataset was not used as a separate dataset during one to four experiments, and it was effectively used in experiment five to show the impact of the deduplication filter.

1) Distribution of Domains and TLDs: The domain and TLDs distribution of a dataset depends on the URL of the phishing page. Therefore, the study first extracted unique domains and TLDs from each dataset. Then frequencies of those were calculated separately. After that, the top fifty domains and TLDs were selected from each dataset. Finally, the percentage of the selected domains proportionally to the size of the relevant dataset was calculated, and those values were plotted in ascending order to have the relevant distribution. Fig. 11 and 12 shows the distribution of domains and TLDs, respectively.

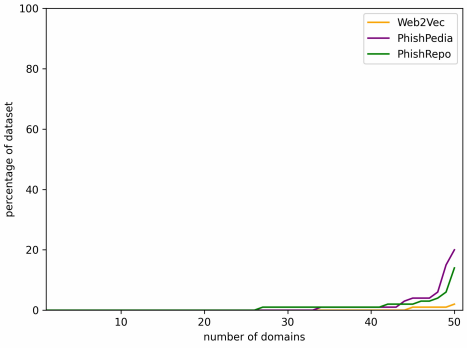

Fig. 11. Distributions of Domains in each Dataset

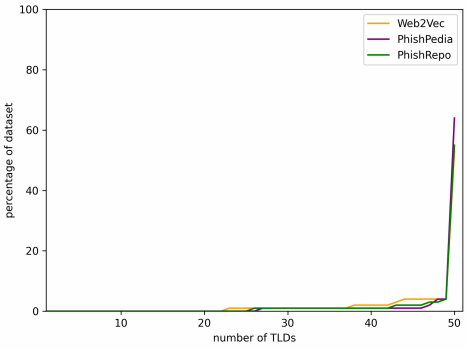

Fig. 12. Distributions of TLDs in each Dataset

As shown in Fig. 11, all the datasets have used more than fifty different domains and TLDs. PhishPedia had a high percentage of the same domain among all datasets, and it was 20% relative to the entire dataset. The PhishRepo dataset had 14% of the same domain, and Web2Vec recorded the lowest number of the same domains. However, the situation is slightly different regarding TLD distribution, as illustrated in Fig. 12. The three datasets have used more than 50% of the '.com' TLD, and it is acceptable because the popular TLDs like '.com' are more often used in phishing nature [22]. Although '.com' acquired a high percentage in all three datasets, more than 50 different TLDs have been included in Web2Vec, PhishPedia, and PhishRepo datasets. Such distribution in domains and TLDs signifies a diverse dataset [22]. Therefore, PhishRepo's dataset is diverse in the perspective of the distribution of domains and TLDs.

2) URL Character Length Distribution and Percentage of HTTPS: The current anti-phishing domain is more toward representation learning approaches like deep learning [12], [17], [18], [19], and it results in black box models that do not visualize the features used during the decision making [12]. Therefore, if a phishing dataset does not have a standard distribution in URL character length as presented in [14] work, it may result in inadequate models for real scenarios [53]. Further, as shown in the APWG report [7], more than 80% of present phishing attacks have come with the HTTPS label, indicating that a high percentage of HTTPS in a phishing dataset is also vital to have a realistic scenario during the model training. Therefore, the number of characters in a URL and

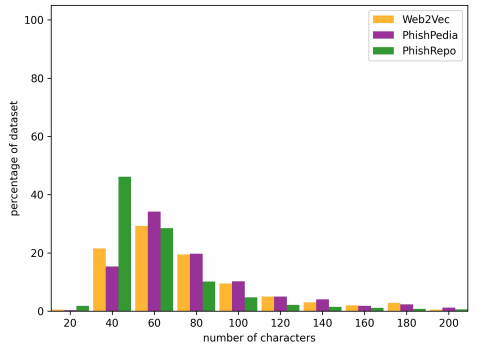

Fig. 13. Character Length Distribution in each Dataset

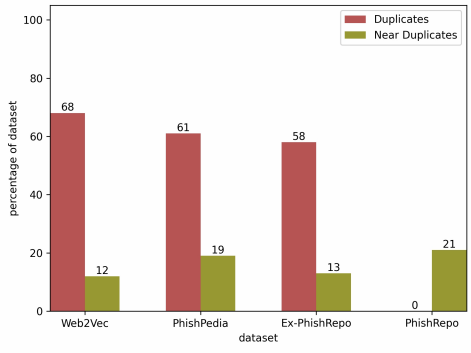

Fig. 15. Percentage of Duplicates and Near-Duplicates of each Dataset

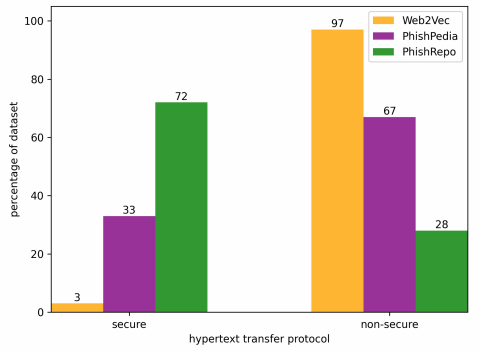

Fig. 14. Percentage of Secure Phishing URLs in each Dataset

percentage of secure phishing URLs proportionally to the size of the relevant dataset was calculated to have URLs character length distribution and percentage of secure phishing URLs.

According to Fig. 13, the URL character length in all three datasets has shown a standard distribution. The highest number of PhishRepo URLs belonged to the 20 to 40 character length category. The other two datasets had a high percentage of 40 to 60 character length URLs. However, all three datasets had URLs under different categories, indicating that these three datasets are diverse in terms of URL character length. In contrast, the secure URLs were deficient in the Web2Vec and PhishPedia datasets. Fig. 14 visualised that the Web2Vec and PhishPedia datasets had 3% and 33% secure URLs. However, current statistics highlighted that nearly 80% of phishing URLs are used HTTPS in the current phishing context [7]. Since it is not reflected in the Web2Vec and PhishPedia datasets, it may lead to inadequate models when these datasets are used in training. However, PhishRepo is shown a high percentage of secure phishing instances, and it has more than 70% of the used dataset. It indicates that the PhishRepo dataset is up to date, and the present phishing nature is sufficiently absorbed.

3) The Tendency of Data Leakage: Data leakage is one of the leading machine learning errors and results in poor prediction outcomes. It happens when the information used in the model train appears during testing time. In the context of phishing data, this can happen in two ways. First, the same data is used multiple times, like the phishing website https://xyz.com appears on many occasions in the dataset. Next, it can happen due to different URLs for the same phishing website, as shown in Fig. 5. In both cases, data leakage could happen if the percentage of duplication pairs is high. Therefore, the current study's data leakage tendency is measured based on the number of duplication pairs available in the used datasets. However, none of the previous studies used such a test to check the tendency of data leakage, which may be the first of its kind.

The experiment used the captured screenshots of phishing web pages since the phishers are always trying their best to have a similar fake page compared to the target page. Therefore, the current study assumed that web pages with similar appearances have the same HTML structure. The assumption is valid in most cases since most phishing websites are coming through phishing kits nowadays [4]. However, the failure of the mentioned assumption will not affect the result of the experiment since the tendency of data leakage is calculated using the number of duplicates. Further, the experiment used a commercial tool called pixolution Flow (https://pixolution.org/), an AI-powered visual search engine for managing and searching visual data when finding duplicates and near-duplicates. The pixolution Flow has a docker image that could index up to 5,000 images free, and that docker is used during the experiment. Therefore, 5000 random samples were selected from each dataset using the pixolution docker before starting the experiment.

The experiment used a 1.0 threshold when searching for duplicates, and the near-duplicates search was configured to use a 0.9 threshold since it is the recommended threshold by the tool. The duplicates and near-duplicate percentages of each dataset are presented in Fig. 15. However, during the indexing step of the tool, the Web2Vec dataset could not index all the screenshots listed since twenty-two images had some issues. Therefore, the presented percentages of the Web2Vec dataset are calculated from 4,978 data items.

After introducing the deduplication component, PhishRepo has improved by decreasing the duplicate images to 0 and keeping the near-duplicate percentage around 20, as shown in Fig. 15. Further, the experiment shows that the other datasets, Web2Vec, PhishPedia and Ex-PhishRepo, have higher duplication percentages (Fig. 15) than PhishRepo. Therefore, the study can claim that the present version of the PhishRepo dataset does not tend to leak data since it does not contain any duplicate pairs. However, phishing cannot eliminate the near-duplicates since phishers mainly target popular web-

TABLE II. ANTI-PHISHING SOLUTIONS USED IN THE EXPERIMENT

Solution	Description
URLNet[a] [54]	A deep learning approach that detects malicious URLs directly from the URL.
StackModel[b] [14]	Detect phishing attacks with the support of URL and HTML content features.
HybridDLM[c] [16]	A deep learning model uses direct URLs with manually extracted HTML content features.

[a]https://github.com/Antimalweb/URLNet
[b]https://drive.google.com/drive/folders/1T4uHRxb_Uk5_kXcJrq68mZ-ezWSQgs_e
[c]https://github.com/sna-hm/HybridDLM

sites, and those attacks may have slight differences. Although PhishRepo's deduplication filter is configured to discard near-duplicates, it checks near-duplicates only three days from a given date due to the previously mentioned practical limitations (Section IV-A3). Therefore, this experiment concludes that the PhishRepo dataset is well-suited for machine learning-based anti-phishing tasks from the perspective of data leakage.

Additionally, based on the experiments mentioned above, the study has shown that the proposed solution, PhishRepo produces a diverse dataset, and it is more suitable for machine learning-based phishing detection studies.

C. PhishRepo's Effectiveness in Anti-Phishing Studies

The ultimate goal of PhishRepo is to provide a phishing dataset for machine learning-based anti-phishing studies. Therefore, a different experiment was performed to prove the effectiveness of PhishRepo's output compared to recently used public phishing datasets. The Web2Vec and PhishPedia datasets were selected for this purpose since both are similar to a certain extent to the PhishRepo dataset from the perspective of the available information. Further, these two datasets were already exercised with two recent anti-phishing solutions [19], [38] that have shown high performances. Therefore, these two datasets, alongside the PhishRepo dataset, were used to train several existing machine learning-based anti-phishing solutions (Table II) separately and evaluated those against the latest phishing attacks.

1) Train and Test Datasets: PhishRepo is an online phishing data repository that expects to grow with time. Although PhishRepo is in the early stage of its journey, it managed to collect around 5000 latest phishing data, and this experiment was planned with these data to compare the effectiveness of PhishRepo data with the state of arts phishing datasets.

Generally, a machine learning model needs a training and test dataset. Therefore, as the first step, the required datasets were constructed. However, the primary intention of the proposed PhishRepo solution is to produce the latest phishing data for anti-phishing studies. Therefore, the experiment required the latest phishing data for the evaluation process. Out of all the selected datasets, the PhishRepo dataset had the latest phishing attacks since it collected phishing attacks up to 02 February 2022. Therefore, the last ten days of phishing attacks were initially separated from the PhishRepo dataset and had 518 records. Those 518 records were added to the test dataset under the phishing label, and the remaining data (i.e. data up to 21 January 2022) were selected as PhishRepo's training dataset.

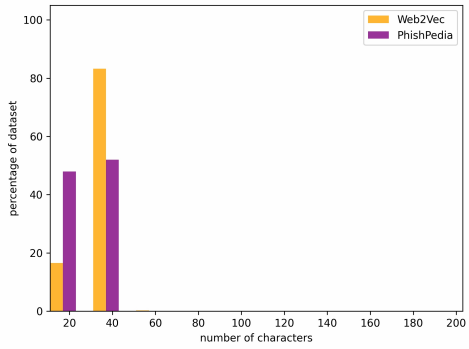

Fig. 16. Legitimate URL Character Length of Web2Vec and PhishPedia

It had 4,757 data, and the amount is reasonable to train a machine learning-based anti-phishing solution since [18] also did a successful anti-phishing study with 4,700 total phishing instances.

The experiment was planned to change only the phishing examples during the training. Therefore, other factors such as the dataset's size, the legitimate examples seen by the solutions and the test dataset, including phishing and legitimate, were kept constant. Since the number of phishing examples needs to be the same in all three cases, 4,757 phishing data were randomly selected from the Web2Vec and PhishPedia datasets to construct the Web2Vec and PhishPedia training datasets.

Next, the experiment required legitimate examples for effective learning. Table I shows that the Web2Vec and PhishPedia original datasets had a legitimate collection. However, those legitimate data were collected from Alexa. If a legitimate dataset is constructed using Alexa without a specific strategy and mixed with a phishing dataset collected from PhishTank, the URL character length plays a significant role and may produce malfunctioned classifiers [53], [22]. Therefore, the Web2Vec and PhishPedia datasets were initially examined by plotting the character length of available legitimate URLs. Fig. 16 shows the character length distribution of those legitimate URLs. It visualises that the mentioned URL character length issue exists with both Web2Vec and PhishPedia legitimate data compared with phishing URL character length available in Fig. 13. Since it affects the final evaluation process of the planned experiment, Web2Vec and PhishPedia legitimate data were not used to construct the training dataset. Therefore, an online phishing dataset named Phishing Websites dataset [55] was used to collect the required legitimate data since it had a reasonable URL character length distribution compared to the [14] work, as shown in Fig. 17.

The experiment planned to have a balanced dataset during the training. Therefore, 4,757 and 518 legitimate data were randomly selected from the Phishing Websites dataset for the training and test datasets. Finally, the training and test datasets contained 9,514 and 1,036 data. Since the experiment wanted consistent legitimate data to effectively evaluate the PhishRepo dataset performance, the same legitimate training samples were added to the Web2Vec, PhishPedia and PhishRepo training datasets. The test dataset was similar in all the experiments, and it has used to evaluate the selected model's performance in each case.

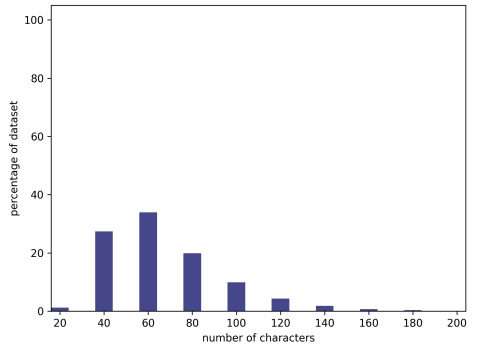

Fig. 17. Legitimate URL Character Length of Phishing Websites Datasets

TABLE III. TRAINED MODELS' PERFORMANCES WITH EACH DATASET

Solution	Dataset	Accuracy	*f1*-Score	FNR*
URLNet	Web2Vec	72.20%	70.79%	0.326
	PhishPedia	78.09%	77.72%	0.236
	PhishRepo	**82.24%**	**83.45%**	**0.104**
StackModel	Web2Vec	58.11%	36.55%	0.759
	PhishPedia	75.87%	69.96%	0.438
	PhishRepo	**89.00%**	**88.97%**	**0.112**
HybridDLM	Web2Vec	67.18%	51.98%	0.645
	PhishPedia	88.22%	87.07%	0.207
	PhishRepo	**93.92%**	**93.89%**	**0.066**

*False Negative Rate (FNR): The number of phishing instances is marked as legitimate proportionally to all existing phishing instances.

2) Performance Evaluation: First of all, Table II anti-phishing solutions were separately trained using the Web2Vec, PhishPedia and PhishRepo training datasets. Then, these models were evaluated using the test dataset. The obtained results during the evaluation process are shown in Table III. However, the URLNet experiment had different models based on the used embedding values, and the best-performed model was selected to present the final results.

As shown in Table III, the PhishRepo dataset has shown high accuracies and f1-score and low FNR in all three cases where the PhishRepo dataset was used. Since the datasets size, legitimate examples, and test dataset were constant in all cases, the phishing examples made the performance difference in each dataset. It implies that the models effectively learned most phishing scenarios with the PhishRepo dataset, and it is more effective when presenting phishing examples for the used models.

Although the PhishRepo dataset has shown some significant performance, one might argue that it is due to the latest phishing examples it contained. However, that is the main objective of the current work. The machine learning-based anti-phishing studies lack the latest phishing data. Therefore, most models are trained with old phishing data, and those models may not perform well with the latest phishing attacks since phishing characteristics constantly change over time. That is what exactly happened during the performed experiment. Since the test dataset contained the latest phishing attacks and both Web2Vec and PhishPedia had old phishing examples, current significant phishing characteristics might not be captured during the training. However, PhishRepo is constantly collecting these phishing examples. Therefore, it contained the latest phishing examples, and more significant characteristics are existed in PhishRepo examples to detect the latest phishing attacks. Thus, the model trained with the PhishRepo dataset captured these new characteristics and performed well with the latest attacks.

Furthermore, as shown in Fig. 15, the PhishRepo dataset does not contain duplicate phishing examples. However, in Web2Vec and PhishPedia, duplication is over 50%. Therefore, compared to the PhishRepo dataset, the Web2Vec and PhishPedia datasets may contain fewer unique phishing examples. Then, although the training dataset size is equal, the amount of learning a model can gain through the training set becomes lower in the other two datasets than in the PhishRepo dataset due to the high duplicate phishing instances available. Therefore, PhishRepo output is more suitable for machine learning-based anti-phishing studies since it produces the latest diverse phishing examples for an effective learning process.

VI. DISCUSSION

Machine learning-based phishing detection desires labelled phishing data at present. The unavailability of such data directs anti-phishing research into many challenges. Some of them are, lingered data collection, data obsoletion, low-quantity data, low-quality data, and lack of multi-modal feature representation. These challenges result in inept learning models, weakening the effort to combat phishing. Therefore, it is essential to fill the current gap in the anti-phishing domain to strengthen future detections. As a result, PhishRepo is introduced as a gap-filling solution to deliver future phishing data needs in the anti-phishing domain. However, it is not just a way of storing data; it is responsible for the latest quality data dissemination to enrich the effectiveness of the anti-phishing solutions.

Phisherman [36], [35] is the only solution in the literature with the same aim as PhishRepo. However, PhishRepo is conceptually superior to Phisherman in many design aspects. Some examples include a deduplication filter, a crowdsourcing-based verification process, malicious submission detection, and the ability to report objections. PhishRepo generally benefits from automated submission architecture, and its design allows it to access a variety of information sources in raw format. Furthermore, the deduplication filter ensures diverse data collection and the elegant verification process used in PhishRepo results in high-quality data. The objection reporting helps to maintain the quality even more. Moreover, the innovative data distribution structure is purposely designed in PhishRepo to attract users, primarily autonomous anti-phishing tools. These tools could integrate PhishRepo more thoughtfully to handle the constantly changing phishing attacks. Further, the proposed network architecture and TAP strategies are critical for the solution's smooth operation from a security perspective.

PhishRepo is a phishing data repository that is accessible online. As a result, anyone interested in the solution could gain access to it and obtain the final benefit, the data. The primary audience for PhishRepo is anti-phishing researchers. They can use this solution effectively to eliminate the phishing data hassle. Since the repository includes multi-modal features, the researcher could use PhishRepo to take their research in a new direction. Furthermore, the raw format in PhishRepo supports

representation learning approaches such as deep learning to design differentiated anti-phishing solutions. It is further intended to support reinforcement learning (RL) environments because PhishRepo includes an interactive feedback facility. With this facility, an implemented RL environment could submit its actions for specific observations and receive quality feedback from PhishRepo. Therefore, the data collected by PhishRepo could aid anti-phishing researchers in various ways, allowing them to conduct more effective research. In addition to such, PhishRepo is an excellent solution for data drifting, which mainly affects machine learning models' performance [21], [22]. Therefore, the latest data collected by PhishRepo could be used to retrain existing models to retain their performance in the fast-evolving nature of phishing.

Moreover, PhishRepo is the first study to examine the tendency of data leakage in a phishing dataset in the anti-phishing domain. It found that the deduplication filter introduced in this study causes no data leakage. The experiments conducted to demonstrate the efficacy of the PhishRepo data have also demonstrated that the data are diverse and do not contain duplicate data, which could lead to a data leakage problem. Furthermore, PhishRepo has been compared with two recently used public datasets using three anti-phishing solutions. There also, PhishRepo outperformed other datasets by achieving high detection accuracy, f1-score and low FNR by showing the strength of the proposed solution.

However, the reliability of PhishRepo is primarily determined by the submissions it receives. Therefore, reporters are essential to the proposed architecture, and corporate reporters are critical because PhishRepo encourages real-time submissions rather than manual ones. Another essential role in PhishRepo is the editor, particularly the crowd user, who is always critical to the success of the beta verification process. However, alpha eliminates the need for a beta. Therefore, PhishRepo assumes that few editors can manage beta verification in the early stages. However, the contributions of the reporters and editors are critical for PhishRepo to continue its process and achieve its ultimate goal.

As a general limitation of the solution, the third-party services' availability is critical in PhishRepo, and the failure of some may affect the solution's continuity. Therefore, PhishRepo expects a collaborative effort against phishing rather than individual combat. Further, a few more anti-phishing communities could be integrated into the alpha verification process to strengthen the alpha process and reduce the human workload in the verification. Moreover, archiving some erroneous pages (e.g., 403 pages, 404 pages, and content not found pages) impacts the PhishRepo data quality. Therefore, additional work will be required in the future to automatically detect erroneous or unwanted pages via web page screenshots and remove such data points from the repository. Then, the quality of the PhishRepo data could be improved further, providing researchers with significantly less noisy data.

VII. Conclusion

While machine learning methods are gaining popularity in phishing detection, the lack of labelled data limited the viability of machine learning-based anti-phishing solutions. Large-scale, diverse data sources are essential in phishing detection in today's context, and it helps researchers have effective machine learning models to combat phishing in the future. PhishRepo comes under these circumstances, and it is an online phishing data repository that collects, verifies, disseminates, and archives real-time phishing data. PhishRepo uses a tactical approach from collection to dissemination. Therefore, it always guarantees the quality of data it saves. Further, automated submission, deduplication filtering, automated verification, crowdsourcing-based human interaction, objection reporting window, and security considerations outperform PhishRepo over similar solutions in the phishing domain.

However, the proposed gap-filling solution's reliability depends on its submissions. Although it is a limitation, PhishRepo identifies its importance and promotes specific tactics to bind users to the solution. Therefore, PhishRepo will be an essential service to provide quality labelled multi-modal feature-based phishing data to detect phishing attacks effectively in the future.

Acknowledgment

The authors acknowledge the support received from the Center for Information Technology Services (CITeS) of the University of Moratuwa, Sri Lanka and Dr Chamath Keppitiyagama of the University of Colombo School of Computing, Sri Lanka.

References

[1] ENISA, *ENISA threat landscape report 2018: 15 top cyber threats and trends.* Publications Office, 2019. [Online]. Available: https://data.europa.eu/doi/10.2824/622757

[2] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing attacks: A recent comprehensive study and a new anatomy," *Frontiers in Computer Science*, vol. 3, Mar. 2021. [Online]. Available: https://doi.org/10.3389/fcomp.2021.563060

[3] W. D. Yu, S. Nargundkar, and N. Tiruthani, "A phishing vulnerability analysis of web based systems," in *2008 IEEE Symposium on Computers and Communications*. IEEE, Jul. 2008. [Online]. Available: https://doi.org/10.1109/iscc.2008.4625681

[4] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," *Expert Systems with Applications*, vol. 106, pp. 1–20, Sep. 2018. [Online]. Available: https://doi.org/10.1016/j.eswa.2018.03.050

[5] H. Huang, S. Zhong, and J. Tan, "Browser-side countermeasures for deceptive phishing attack," in *2009 Fifth International Conference on Information Assurance and Security*. IEEE, 2009. [Online]. Available: https://doi.org/10.1109/ias.2009.12

[6] 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 Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2797–2819, 2017. [Online]. Available: https://doi.org/10.1109/comst.2017.2752087

[7] APWG, "Phishing activity trends report: 4th quarter 2020," *Anti-Phishing Working Group. Retrieved February, 09*, p. 13, 2021.

[8] N. C. R. L. Y. Teraguchi and J. C. Mitchell, "Client-side defense against web-based identity theft," *Computer Science Department, Stanford University. Available: http://crypto. stanford. edu/SpoofGuard/webspoof. pdf*, 2004.

[9] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, "Anti-phishing phil," in *Proceedings of the 3rd symposium on Usable privacy and security - SOUPS '07*. ACM Press, 2007. [Online]. Available: https://doi.org/10.1145/1280680.1280692

[10] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in *2010 Proceedings IEEE INFOCOM*. IEEE, Mar. 2010. [Online]. Available: https://doi.org/10.1109/infcom.2010.5462216

[11] M. Baslyman and S. Chiasson, ""smells phishy?": An educational game about online phishing scams," in *2016 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, Jun. 2016. [Online]. Available: https://doi.org/10.1109/ecrime.2016.7487946

[12] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. Gonzalez, "Classifying phishing URLs using recurrent neural networks," in *2017 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, Apr. 2017. [Online]. Available: https://doi.org/10.1109/ecrime.2017.7945048

[13] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, vol. 2017, pp. 1–20, 2017. [Online]. Available: https://doi.org/10.1155/2017/5421046

[14] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27–39, May 2019. [Online]. Available: https://doi.org/10.1016/j.future.2018.11.004

[15] W. Wang, F. Zhang, X. Luo, and S. Zhang, "PDRCNN: Precise phishing detection with recurrent convolutional neural networks," *Security and Communication Networks*, vol. 2019, pp. 1–15, Oct. 2019. [Online]. Available: https://doi.org/10.1155/2019/2595794

[16] S. Ariyadasa, S. Fernando, and S. Fernando, "Detecting phishing attacks using a combined model of LSTM and CNN," *International Journal of ADVANCED AND APPLIED SCIENCES*, vol. 7, no. 7, pp. 56–67, Jul. 2020. [Online]. Available: https://doi.org/10.21833/ijaas.2020.07.007

[17] C. Opara, B. Wei, and Y. Chen, "HTMLPhish: Enabling phishing web page detection by applying deep learning techniques on HTML analysis," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2020. [Online]. Available: https://doi.org/10.1109/ijcnn48605.2020.9207707

[18] C. Opara, Y. Chen, and B. wei, "Look before you leap: Detecting phishing web pages by exploiting raw url and html characteristics," 2020.

[19] J. Feng, L. Zou, O. Ye, and J. Han, "Web2vec: Phishing webpage detection method based on multidimensional features driven by deep learning," vol. 8, pp. 221 214–221 224, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.3043188

[20] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013. [Online]. Available: https://doi.org/10.1109/surv.2013.032213.00009

[21] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious url detection using machine learning: A survey," 2019.

[22] A. E. Aassal, S. Baki, A. Das, and R. M. Verma, "An in-depth benchmarking and evaluation of phishing detection research for security needs," *IEEE Access*, vol. 8, pp. 22 170–22 192, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.2969780

[23] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight URL-based phishing detection," *Future Internet*, vol. 13, no. 6, p. 154, Jun. 2021. [Online]. Available: https://doi.org/10.3390/fi13060154

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[25] 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, Mar. 2019. [Online]. Available: https://doi.org/10.1016/j.eswa.2018.09.029

[26] L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. Nguyen, "An efficient approach for phishing detection using single-layer neural network," in *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*. IEEE, Oct. 2014. [Online]. Available: https://doi.org/10.1109/atc.2014.7043427

[27] E.-S. M. El-Alfy, "Detection of phishing websites based on probabilistic neural networks and k-medoids clustering," *The Computer Journal*, vol. 60, no. 12, pp. 1745–1759, Apr. 2017. [Online]. Available: https://doi.org/10.1093/comjnl/bxx035

[28] W. Chen, W. Zhang, and Y. Su, "Phishing detection research based on LSTM recurrent neural network," in *Communications in Computer and Information Science*. Springer Singapore, 2018, pp. 638–645. [Online]. Available: https://doi.org/10.1007/978-981-13-2203-7_52

[29] M. Chatterjee and A.-S. Namin, "Detecting phishing websites through deep reinforcement learning," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, Jul. 2019. [Online]. Available: https://doi.org/10.1109/compsac.2019.10211

[30] S. Bell and P. Komisarczuk, "An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank," in *Proceedings of the Australasian Computer Science Week Multiconference*. ACM, Jan. 2020. [Online]. Available: https://doi.org/10.1145/3373017.3373020

[31] V. Zeng, S. Baki, A. E. Aassal, R. Verma, L. F. T. D. Moraes, and A. Das, "Diverse datasets and a customizable benchmarking framework for phishing," in *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*. ACM, Mar. 2020. [Online]. Available: https://doi.org/10.1145/3375708.3380313

[32] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 2015–2028, Apr. 2018. [Online]. Available: https://doi.org/10.1007/s12652-018-0798-z

[33] A. Orunsolu, A. Sodiya, and A. Akinwale, "A predictive model for phishing detection," *Journal of King Saud University - Computer and Information Sciences*, Dec. 2019. [Online]. Available: https://doi.org/10.1016/j.jksuci.2019.12.005

[34] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15 196–15 209, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2892066

[35] G. Tally, "Phisherman: A phishing data repository," in *2009 Cybersecurity Applications & Technology Conference for Homeland Security*. IEEE, Mar. 2009. [Online]. Available: https://doi.org/10.1109/catch.2009.24

[36] G. Tally, D. Sames, T. Chen, C. Colleran, D. Jevans, K. Omiliak, and R. Rasmussen, "The phisherman project: Creating a comprehensive data collection to combat phishing attacks," *Journal of Digital Forensic Practice*, vol. 1, no. 2, pp. 115–129, Jul. 2006. [Online]. Available: https://doi.org/10.1080/15567280601015564

[37] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," in *2012 International Conference for Internet Technology and Secured Transactions*. IEEE, 2012, pp. 492–497.

[38] 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. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/lin

[39] K. Beck and J. Zhan, "Phishing using a modified bayesian technique," in *2010 IEEE Second International Conference on Social Computing*. IEEE, Aug. 2010. [Online]. Available: https://doi.org/10.1109/socialcom.2010.100

[40] E. Buber, O. Demir, and O. K. Sahingoz, "Feature selections for the machine learning based detection of phishing websites," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, Sep. 2017. [Online]. Available: https://doi.org/10.1109/idap.2017.8090317

[41] J. C. Chang, S. Amershi, and E. Kamar, "Revolt: Collaborative crowdsourcing for labeling machine learning datasets," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, May 2017. [Online]. Available: https://doi.org/10.1145/3025453.3026044

[42] L. Zhao, G. Sukthankar, and R. Sukthankar, "Incremental relabeling for active learning with noisy crowdsourced annotations," in *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*. IEEE, Oct. 2011. [Online]. Available: https://doi.org/10.1109/passat/socialcom.2011.193

[43] A. Drutsa, V. Farafonova, V. Fedorova, O. Megorskaya, E. Zerminova, and O. Zhilinskaya, "Practice of efficient data collection via crowdsourcing at large-scale," 2019.

[44] T. Aitamurto, A. Leiponen, and R. Tee, "The promise of idea crowdsourcing–benefits, contexts, limitations," *Nokia Ideasproject White Paper*, vol. 1, pp. 1–30, 2011.

[45] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Applied Statistics*, vol. 28, no. 1, p. 20, 1979. [Online]. Available: https://doi.org/10.2307/2346806

[46] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *Proceedings of the ACM SIGKDD Workshop on Human Computation - HCOMP '10*. ACM Press, 2010. [Online]. Available: https://doi.org/10.1145/1837885.1837906

[47] D. L. Hansen, P. J. Schone, D. Corey, M. Reid, and J. Gehring, "Quality control mechanisms for crowdsourcing," in *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13*. ACM Press, 2013. [Online]. Available: https://doi.org/10.1145/2441776.2441848

[48] C. Eickhoff, "Cognitive biases in crowdsourcing," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, Feb. 2018. [Online]. Available: https://doi.org/10.1145/3159652.3159654

[49] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," 2010.

[50] D. T. Nguyen, F. Alam, F. Ofli, and M. Imran, "Automatic image filtering on social networks using deep learning and perceptual hashing during crises," 2017.

[51] R. Gowtham and I. Krishnamurthi, "A comprehensive and efficacious architecture for detecting phishing webpages," *Computers & Security*, vol. 40, pp. 23–37, 2014.

[52] S. Ariyadasa, S. Fernando, and S. Fernando, "phishrepo-dataset," 2022. [Online]. Available: https://data.mendeley.com/datasets/ttmmtsgbs8/3

[53] R. M. Verma, V. Zeng, and H. Faridi, "Data quality for security challenges," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Nov. 2019. [Online]. Available: https://doi.org/10.1145/3319535.3363267

[54] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, "Urlnet: Learning a URL representation with deep learning for malicious URL detection," *CoRR*, vol. abs/1802.03162, 2018. [Online]. Available: http://arxiv.org/abs/1802.03162

[55] S. Ariyadasa, S. Fernando, and S. Fernando, "Phishing websites dataset," 2021. [Online]. Available: https://data.mendeley.com/datasets/n96ncsr5g4/1