International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0509
e ISSN: 2584-2854
Volume: 03
Issue: 11 November 2025
Page No: 3216 - 3222

PHISHSIM: Phishing Website Detection

Nivyashree R[1], Bhoomika S R[2], Chiranth H M[3], Bhavan N Gowda[4], Chakram Janya H U[5]
[1]Asst. Professor, Department of Computer Science & Engineering, Malnad College of Engineering
[2,3,4,5]Department of Computer Science & Engineering, Malnad College of Engineering
Email ID : *rns@mcehassan.ac.in[1], bhoomikasr17@gmail.com[2], chiruchiranthhm@gmail.com[3], bhavanng@gmail.com[4], chakramjanya69@gmail.com[5]*

Abstract
In this paper, we introduce a powerful new approach for detecting phishing websites that is entirely feature-free. Our method, called PhishSim, uses the Normalized Compression Distance (NCD), a technique that requires no specialized parameters. NCD works by measuring the similarity of two websites through compression, eliminating the time and effort typically needed for feature extraction. We classify suspicious pages by comparing their HTML content to a database of known phishing sites. To keep our database efficient, we employ the Furthest Point First (FPF) algorithm to extract "prototypes"—representative examples of phishing webpage clusters. Furthermore, we integrate an incremental learning algorithm to make the system continuously adaptable, ensuring detection remains sharp even as attack methods evolve (concept drift). Tested on a large, realistic dataset, PhishSim significantly outperforms previous methods, achieving an outstanding AUC score of 98.68% , a high true positive rate (TPR) of about 90% , and a remarkably low false positive rate (FPR) of 0.58%. By using prototypes, we avoid storing large amounts of historical data, making the system practical for real-world deployment with a fast processing time of approximately 0.3 seconds.
Keywords: HTML , Webpage , Prototype Extraction , Website Similarity , Compression , dthreshold (Distance Threshold) , Quality of Clustering (QC) metric

1. Introduction
Phishing is a major cybersecurity threat, defined as a social engineering attack that tricks people into giving up sensitive information like passwords or credit card numbers. The most common form of phishing on the web uses convincing, professional-looking fake websites to lure victims. This problem is getting worse because phishing toolkits and free hosting are readily available. These resources let attackers launch large campaigns quickly. The ongoing changes in these attacks make it very hard to create reliable detection systems. Historically, research has focused on two main approaches. Feature-Based Methods rely on traditional Machine Learning or Deep Learning to spot specific harmful traits, such as URL structure or form types. While these methods were initially effective, they quickly become ineffective when attackers change their tactics. This issue is known as concept drift.
Historically, research has followed two main paths:

1. **Feature-Based Methods:** These rely on traditional Machine Learning or Deep Learning to identify specific malicious characteristics (e.g., URL structure or form types). While initially effective, they are easily broken when attackers change their tactics—a problem known as **concept drift**.
2. **Similarity-Based Methods:** They are good for filtering large numbers of sites, but they often require converting the website into a complex format, like DOM trees or Doc2Vec vectors. Our work, PhishSim, is based on the key insight that 90% of confirmed phishing sites are simply variations or replicas of older attacks. We propose a new method that does not use features and is based on similarity. This approach avoids the limitations of previous methods by comparing the raw structural content of the websites themselves.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0509
e ISSN: 2584-2854
Volume: 03
Issue: 11 November 2025
Page No: 3216 - 3222

1.1. Key Contributions

- Our main contributions to the field are: - Introducing a systematic method for measuring website similarity, especially for detecting nearly identical phishing websites using the Normalized Compression Distance (NCD). [1]
- Developing PhishSim, a tool without parameters that uses NCD and prototype-based learning to identify new phishing sites generated from existing templates.
- Designing a thorough, feature-free system framework suitable for use in corporate intranets or cloud environments.
- Integrating an incremental learning framework that allows the system to continuously change and improve over time without the need to retrain entire models..

1.2.Concepts and Definitions

- **Normalized Compression Distance (NCD) for Similarity:** NCD is a unique distance metric that requires no parameters. It is based on information theory and approximates the theoretically ideal Normalized Information Distance (NID**).** [2]
- **The basic principle of NCD is straightforward.** If two files share information, compressing them together will be much more efficient than compressing them separately**.**The fundamental principle of NCD is simple: **if two files share information, compressing them together will be much more efficient than compressing them separately**. NCD calculates this relationship using a standard compression algorithm C (we found **LZMA** works best):

$$NCD(x,y)=max\{C(x),C(y)]C(xy)-min\{C(x),C(y)\}$$

[cite: 643]

Here, x and y are the two files (HTML contents), C(x) is the compressed size of x, and C(xy) is the size of their combined compressed form. A value close to 0 indicates high similarity, meaning they compress well together. A value close to 1 indicates low similarity, meaning they are distinct. We apply NCD to the website's HTML content because phishing kits create structurally similar pages. To make our method strong against common tricks used by attackers, we first remove all text and HTML comments. This leaves only the structural HTML tags that the browser renders. This approach protects us from code obfuscation and the addition of hidden, invisible elements. [3]

1.3.Prototype-Based Learning:

To handle the large number of potential phishing sites, we use a prototype-based clustering approach. A prototype is simply an actual data point, a phishing website chosen to represent an entire group of similar sites.

- **Prototype Extraction: The FPF Algorithm** We adapted the O(nk)-time **Furthest Point First (FPF)** algorithm.
- The process begins by selecting any data point as the first prototype.
- The next prototype is chosen as the point that is **furthest** from all previously selected prototypes.
- This continues until every data point is within a specific distance (dthreshold) of a chosen prototype.

NCD-Based Classification When a new website x is checked, we calculate its NCD against every prototype z.

- If NCD(x,z)<dthreshold
  for any z∈prototypes, classify x as phishing[cite: 173, 179]

Incremental Learning Our system is designed for continuous learning. When new, legitimate phishing samples are detected (those that the current model *fails* to classify), we extract *new* prototypes from them using FPF and add them to the main Prototype DB. This ensures the system constantly learns from new attacks without a complete, time-consuming overhaul.

2. PhishSim System Overview

PhishSim is proposed as a centralized, server-based solution deployable in corporate networks, by ISPs, or on cloud platforms.

2.1.Phishing Website Classification

1. **URL Input:** The system intercepts the URL a user attempts to visit.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0509
e ISSN: 2584-2854
Volume: 03
Issue: 11 November 2025
Page No: 3216 - 3222

2. **HTML DOM Acquisition:** We use the open-source **Chromium** engine to render the page and obtain the HTML DOM, simulating a user's web browser experience.
3. **Core Classification:** The system performs the NCD-based comparison against the prototypes. [4]
4. **Action:** A phishing prediction results in a warning page for the user.

2.2.Phishing Prototype Database Update

This is a critical, periodic process to combat the fleeting nature of phishing sites (average lifetime is only a few hours).

1. **New Data Inflow:** We receive new phishing URLs from blacklists (like PhishTank) or user feedback.
2. **Prototype Generation:** The new data is passed to the **Prototype Extraction** module, where FPF selects new representative prototypes.
3. **DB Update:** These new prototypes are added to the Prototype DB, ensuring the classifier remains current. Figure 1 shows Classification Using Prototypes Figure 2 shows Furthest Point Algorithm Figure 3 shows Process Algorithm Figure 4 shows System Diagram of the Feature – free Detection Framework

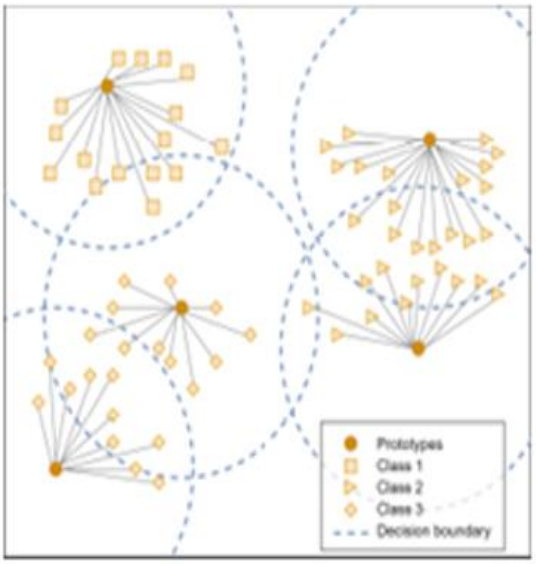

Figure 1 Classification Using Prototypes

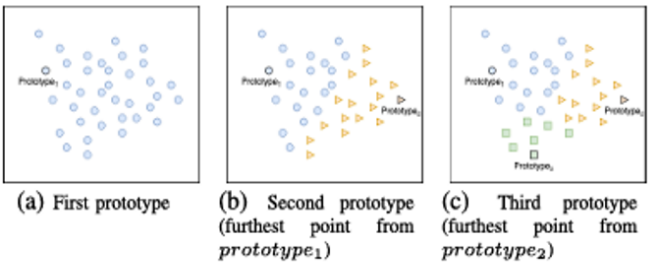

Fig. 2. Furthest Point First Algorithm.

Figure 2 Furthest Point Algorithm

Algorithm 1 Prototype Extraction

$prototypes \leftarrow \emptyset$
for all $x \in data$ **do**
  $distance[x] \leftarrow \infty$
  $cluster[x] \leftarrow \emptyset$
end for
while $max(distance) > d_{threshold}$ **do**
  $z \leftarrow \arg\max_{x \in data} distance[x]$
  for $x \in data$ **do**
    if $distance[x] > NCD(x, z)$ **then**
      $distance[x] \leftarrow NCD(x, z)$
      $cluster[x] \leftarrow z$
    end if
  end for
  $prototypes \leftarrow prototypes \cup \{z\}$
  $data \leftarrow data \setminus \{z\}$
end while

Figure 3 Process Algorithm

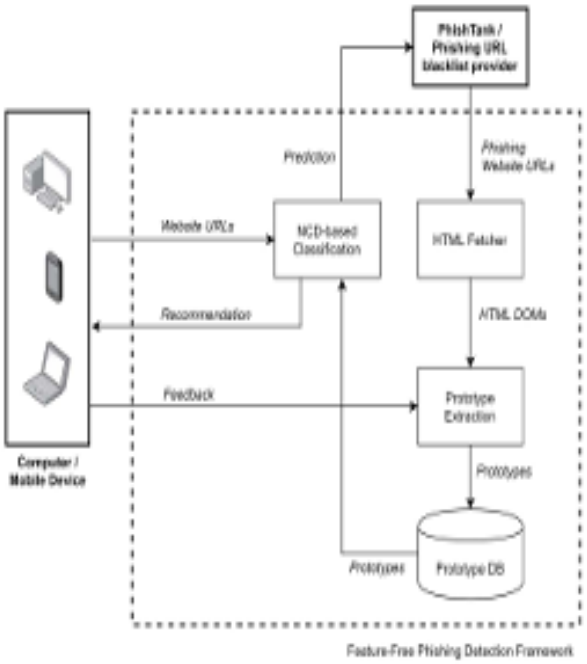

Figure 4 System Diagram of the Feature – free Detection Framework

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0509
e ISSN: 2584-2854
Volume: 03
Issue: 11 November 2025
Page No: 3216 - 3222

The flow diagram illustrates how user requests (Website URLs) go into the NCD-based Classifier (left side), which uses Prototypes from the Prototype DB and generates a Recommendation. Simultaneously (right side), new phishing URLs from Blacklist providers are fed through an HTML Parser, then to Prototype Extraction, which updates the Prototype DB. This closed loop enables continuous, incremental learning. Figure 6 shows Phishing Websites with Similar HTML Contents (Cluster 1) Figure 7 shows Netflix Legitimate Website [5]

(a) P_NTF_52 (b) P_NTF_60

Figure 5 Phishing Websites with Similar HTML Contents (Cluster 1)

Figure 6 Netflix Legitimate Website

- Visual Appearance: These two websites have a very distinctive design. For example, P_NTF_52 features a hero image with the Daredevil character and a small, centered sign-in box, while P_NTF_60 features a full-bleed grid of movie posters and a larger, darker sign-in box.
- HTML Analysis (NCD Result): Despite their visual differences, these two websites were found to be highly similar based on the Normalized Compression Distance (NCD) calculation on their HTML DOM files, resulting in an NCD value of 0.04.
- Conclusion: This high similarity suggests that the websites' HTML DOMs are almost identical, likely because they were built using the same phishing kit. This demonstrates that the NCD method can group attacks from the same toolkit even if the attackers change the visual styling. [6]
- Context: This image serves as the true target for comparison with the phishing websites shown in Figure 6 and is used in the Similarity Analysis (Section V).
- Comparison to Phishing Sites (Figure 6): When the legitimate site (Figure 7) was analyzed using NCD on its screenshot image, it was not detected as similar to any of the phishing websites, despite some similar element styles. The paper notes that frequent updates to the background image (e.g., latest movie or TV series) may contribute to this visual difference. Figure 7 shows Website Clusters

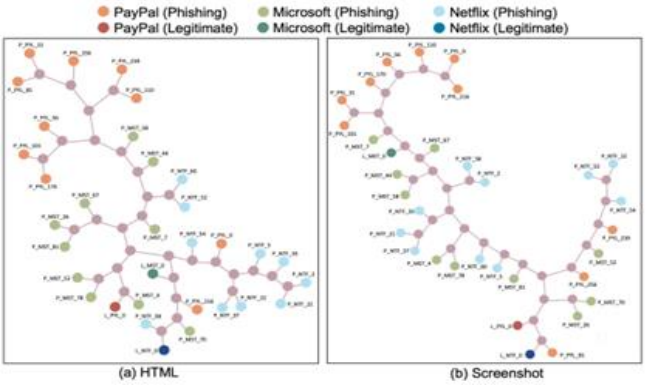

Figure 7 Website Clusters

3. Performance Evaluation

We evaluated PhishSim against two leading similarity detection techniques: Proportional Distance and the Doc2Vec model. Our methodology uses a temporal split for training and testing, which is considered superior to cross-validation as it prevents overestimation of performance by training on "future" data. Crucially, the testing environment uses a highly imbalanced class ratio of 1 phishing site to 140 legitimate sites, closely mimicking real-world traffic. [7]

3.1.PhishSim Performance

The optimal dthreshold was determined to be 0.251 by optimizing the Quality of Clustering (QC) metric.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0509
e ISSN: 2584-2854
Volume: 03
Issue: 11 November 2025
Page No: 3216 - 3222

Table 1 Overall Metrics

Performance Metrics (Default Threshold)	PhishSim (NCD)	Proportional Distance [18]
TPR (True Positive Rate)	89.75%	84.08%
FPR (False Positive Rate)	0.58%	0.10%
G-mean (Geometric Mean)	94.47%	91.65%
AUC Score	0.9868	0.9863
Performance Metrics (Default Threshold)	PhishSim (NCD)	Proportional Distance [18]
TPR (True Positive Rate)	89.75%	84.08%
FPR (False Positive Rate)	0.58%	0.10%
G-mean (Geometric Mean)	**G-mean** (Geometric Mean)	**G-mean** (Geometric Mean)
94.47%	94.47%	94.47%
91.65%	91.65%	91.65%
94.34%	94.34%	94.34%

- Overall Performance: PhishSim achieved the best overall classification ability, demonstrated by the highest AUC score of 0.9868. [8]
- Real-World Reliability: The G-mean (a better measure for imbalanced data) of 94.47% was the highest, confirming PhishSim's superior balance between detection success (TPR) and false alerts (TNR).
- FPR vs. TPR: While Doc2Vec had a slightly higher TPR, its incredibly high FPR (6.95%) would make it unusable in a real-world settings experience constant false alarms. PhishSim offers a strong combination of nearly 90% true positive rate (TPR) with a low 0.58% false positive rate (FPR).

3.2. Incremental Learning

- In the incremental experiment, the model was updated weekly for 39 weeks.
- Detection Rate (TPR): PhishSim consistently kept a TPR close to 90%, usually outperforming the proportional distance method.
- False Alarm Stability: Importantly, PhishSim maintained a low and steady FPR of under 0.8% throughout the iterations. This is a sharp contrast to the Doc2Vec-based method, which had a higher FPR and showed a tendency to increase over time. Figure 8 shows TPR in Incremental Learning Setting Figure 9 shows FPR in Incremental Learning Setting

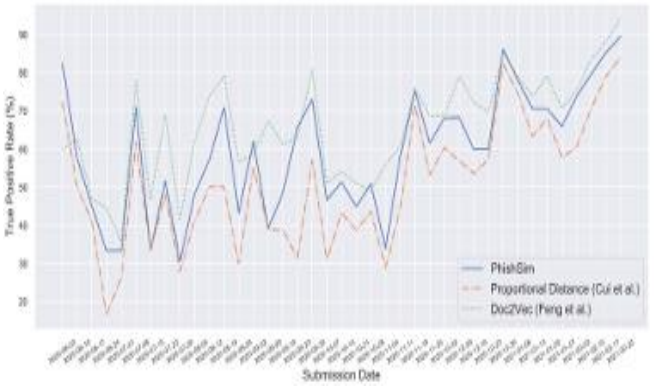

Figure 8 TPR in Incremental Learning Setting

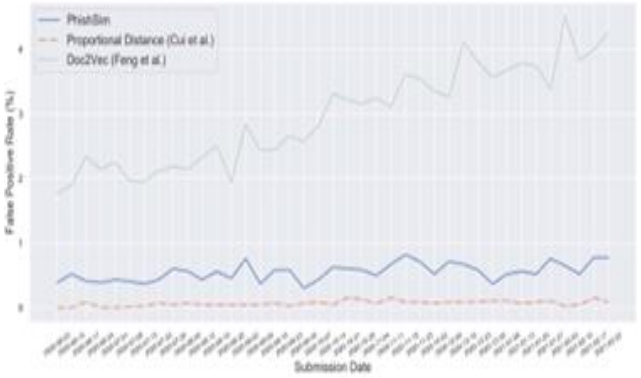

Figure 9 FPR in Incremental Learning Setting

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0509
e ISSN: 2584-2854
Volume: 03
Issue: 11 November 2025
Page No: 3216 - 3222

Setting. Description: Figure 15 plots the True Positive Rate over the weekly iterations, showing PhishSim's consistent performance near the 90% mark. Figure 16 plots the False Positive Rate, visually demonstrating that PhishSim (dark line) maintains a significantly lower and more stable FPR (below 0.8%) than the other methods, making it robust against concept drift over time. [9]

4. Run-Time and Memory Analysis

4.1.Run-Time Performance

The runtime is dominated by the NCD calculation, which involves compressing the concatenation of the website with each prototype.

- Query Time: With 1,366 prototypes in the database, the average time to process a single website is approximately 0.3 seconds.
- Model Update Time: PhishSim takes the least amount of time to update its model in successive iterations compared to the baseline methods (which require clustering from scratch). This is a major advantage for a continuously operating system. Figure 10 shows Total Process Duration 1st to 5th Iteration

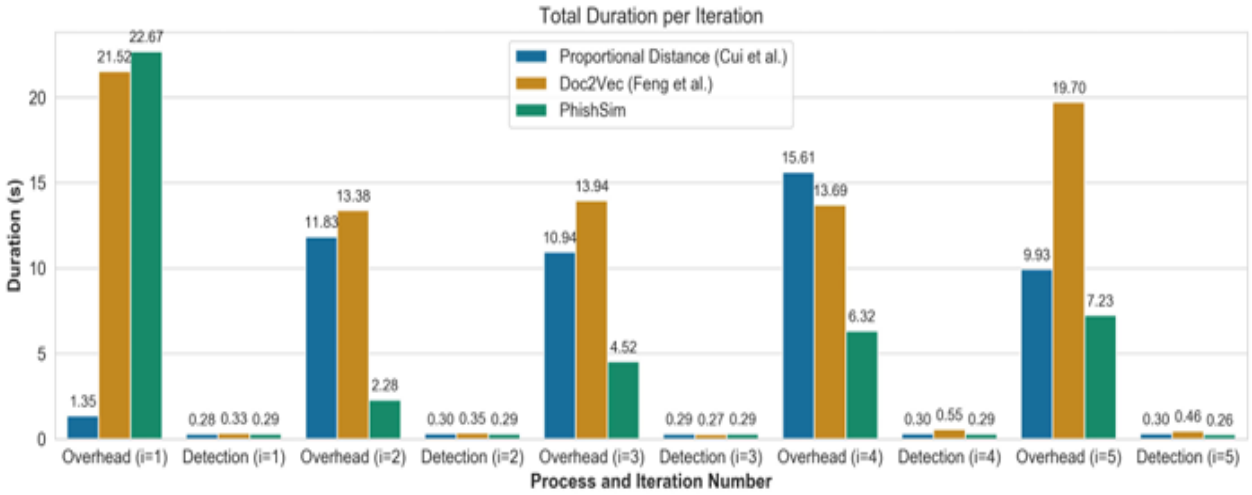

Figure 10 Total Process Duration 1st to 5th Iteration

Description: Overall Duration of Process from 1st to 5th Iteration. The bar chart illustrates the time taken for the Overhead (model update/preparation) and Detection activities. Stages for the three approaches across five cycles. PhishSim features a quick Detection time (approximately 03s), and importantly, its Overhead duration reduces at a faster rate,following the initial iteration, signifying a significantly quicker incremental learning progression.

4.2. Memory Requirements

PhishSim is highly efficient in memory usage because of its prototype extraction.

- Data Reduction: The system attained a compression ratio of 0.15, indicating that 1,366 prototypes embodied 9,034 fraudulent websites.
- Storage Size: Considering the average HTML DOM size of 727 B, saving all 1,366 prototypes necessitates only 0947 MB of data capacity.Prototyping completely removes the necessity of keeping a vast amount of complete historical data.

Conclusion

We presented PhishSim, an efficient, feature-less method for detecting phishing that employs the Normalized Compression Distance (NCD) and a prototype-oriented incremental learning framework. Measuring structural similarity through rendered HTML makes our approach resilient to new, developing assaults (concept shift) that would normally disrupt feature-oriented detection systems. The assessment of PhishSims using a substantial, authentic dataset validated its superiority, resulting in

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0509
e ISSN: 2584-2854
Volume: 03
Issue: 11 November 2025
Page No: 3216 - 3222

an AUC score of 98.68% a high G-mean that guarantees dependable performance in unbalanced real-world situations. The system's efficiency is equally impressive, needing under 1 MB of storage for its detection model and categorizing a website in approximately 03 seconds .For future endeavours, we propose creating a technique for the ongoing upkeep of the prototype collection specifically by eliminating outdated prototypes to enhance storage efficiency and run-time performance.

Acknowledgements

Any views, results, and conclusions or suggestions presented in this ,the opinions expressed in the paper belong to the authors and may not align with the perspectives of the scholarship provider.

References

[1]. ALEXA. TOP 500 SITES IN EACH COUNTRY. ACCESSED: MAY 11, 2020. [ONLINE].AVAILABLE:HTTPS://WWW. ALEXA.COM/TOPSITES/COUNTRIES
[2]. ATTACKERS USE MORSE CODE, OTHER ENCRYPTION METHODS IN EVASIVE PHISHING CAMPAIGN. ACCESSED: DEC. 17, 2021. [ONLINE]. AVAILABLE:
[3]. COMMON CRAWL. ACCESSED: MAR. 25, 2021. [ONLINE]. AVAILABLE: HTTPS:// COMMONCRAWL.ORG/
[4]. GOOGLE CUSTOM SEARCH. ACCESSED: MAR. 4, 2019. [ONLINE]. AVAILABLE:
[5]. PHISHERS' FAVORITES: AFTER FIVE QUARTERS, MICROSOFT IS UNSEATED BY PAYPAL. ACCESSED: APR. 12, 2020. [ONLINE]. AVAILABLE: HTTPS://WWW. VADESECURE.COM/EN/PHISHERS-FAVORITES-Q3-2019/
[6]. PHISHERS' FAVORITES: IT'S LONELY AT THE TOP: MICROSOFT REMAINS THE #1 IMPERSONATED BRAND IN PHISHING ATTACKS. ACCESSED: JAN. 28, 2020. [ONLINE]. AVAILABLE:
[7]. PHISHERS' FAVORITES: MICROSOFT HOLDS ONTO THE #1 SPOT BUT FACE BOOK PHISHING SURGES. ACCESSED: APR. 12, 2020. [ONLINE]. AVAILABLE:
[8]. PHISHERS' FAVORITES: PAYPAL LEADS, NOTE PHISHING INCREASES, AND SMALLER BANKS BECOME BIGGER TARGETS. ACCESSED: APR. 12, 2020. [ONLINE]. AVAILABLE:
[9]. PHISHTANK: AN ANTI-PHISHING SITE. ACCESSED: OCT. 1, 2018. [ONLINE]. AVAILABLE: https://www.phishtank.com/