future internet

MDPI

Article

A Novel Logo Identification Technique for Logo-Based Phishing Detection in Cyber-Physical Systems

Padmalochan Panda [1,†], Alekha Kumar Mishra [1,†] and Deepak Puthal [2,3,*,†]

[1] Department of Computer Science and Engineering, National Institute of Technology Jamshedpur, Jharkhand 831014, India
[2] Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi 127788, United Arab Emirates
[3] Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), University of Technology Sydney, Ultimo, NSW 2007, Australia
* Correspondence: deepak.puthal@uts.edu.au
† These authors contributed equally to this work.

Abstract: The first and foremost task of a phishing-detection mechanism is to confirm the appearance of a suspicious page that is similar to a genuine site. Once this is found, a suitable URL analysis mechanism may lead to conclusions about the genuineness of the suspicious page. To confirm appearance similarity, most of the approaches inspect the image elements of the genuine site, such as the logo, theme, font color and style. In this paper, we propose a novel logo-based phishing-detection mechanism that characterizes the existence and unique distribution of hue values in a logo image as the foundation to unambiguously represent a brand logo. Using the proposed novel feature, the detection mechanism optimally classifies a suspicious logo to the best matching brand logo. The experiment is performed over our customized dataset based on the popular phishing brands in the South-Asia region. A set of five machine-learning algorithms is used to train and test the prepared dataset. We inferred from the experimental results that the ensemble random forest algorithm achieved the high accuracy of 87% with our prepared dataset.

Keywords: phishing; phishing detection; logo-based detection; hue value ratio; pixel hue density distribution

check for updates

Citation: Panda, P.; Mishra, A.K.; Puthal, D. A Novel Logo Identification Technique for Logo-Based Phishing Detection in Cyber-Physical Systems. *Future Internet* **2022**, *14*, 241. https://doi.org/10.3390/fi14080241

Academic Editor: Tinghuai Ma

Received: 8 July 2022
Accepted: 11 August 2022
Published: 15 August 2022

1. Introduction

Customer-oriented e-commerce solutions have brought forward a number of challenges in Cyber-Physical Systems. It is now common to conduct substantial transactions online for making basic purchases, internet banking, paying regular bills, acquiring a mortgage, vehicle loan, paying taxes and many more [1]. These online activities opens space for the hackers to exploit the vulnerabilities of the existing transaction service and launch cyberattacks in order to achieve financial gain or bring down the reputation of a person or organization.

Among the list of common threats, phishing is a threat of major concern in IoT-based Cyber-Physical Systems [2,3]. Phishing is a deception tactic that uses impersonated webpages, emails, calls or sms to extract classified data from individuals, such as their usernames, secret passwords, credit-card-related information and bank account details [4]. Phishing begins with a spoofed email, and then the victim is redirected towards the falsified websites to obtain the required information [5].

Phishing threats are widespread these days, and researchers are contributing various types of solutions for phishing detection. It is reported that fraudsters employ the optical characteristics stolen from genuine websites, particularly the logo, in their phishing websites in order to visually convince a victim to believe that he/she is using the genuine site.

Moreover, they use a domain name close to the original domain with a minor mutation, such as facebook.com changed to facbook.com or faceboook.com. In order to detect a phishing site, the first task is to conclude that it is visually similar to the genuine site with a major or minor inconsistency observed in the URL and other related parameters. The most important elements that make a fake webpage similar to the original one are logo images. Therefore, logo-based visual-similarity-based phishing detection approach lays the initial steps for an accurate and sophisticated phishing-detection technique.

In a logo-based phishing detection approach, simple pixel-by-pixel matching is not effective to solve a phishing problem. This is because an attacker may slightly change a pixel's intensity to result in a lower matching score while not affecting the visual image quality. That is why feature extraction and selection is an important part of a logo-based detection approach. Several logo-based phishing-detection mechanisms have been reported in the literature utilizing a wide range of image features to compare and compute the similarity of a forged logo with the genuine ones. We inferred from the reported literature that most of the detection mechanisms work only with images of homogeneous dimension. In order to apply their classification and detection mechanism, the logo images require transformation. In this process, sometimes the actual characteristics of a logo are lost to a considerable extent.

In this paper, we propose a logo-based phishing-detection mechanism to assess the identity consistency between the real and projected identities of a website using a hybrid technique including image-based similarity and machine learning (ML)-based approaches. The proposed mechanism uses the existence of unique set of hue values and their distribution in images irrespective of size to unambiguously identify the characteristics and use it to find the similarity between the logos. The advantage of the proposed mechanism is that it can detect and classify a logo image irrespective of its size. The contributions of this paper are as follows:

- We propose a novel logo-identification mechanism for logo-based phishing detection.
- The proposed mechanism uses the hue value ratio and the pixel density distribution in a logo that uniquely defines its feature and identifies well-known brands.
- A number of 21 brands with 48 different classes of logos are used for training the ML model.
- The detection accuracy of ensemble random forest algorithm is found to be the best with an accuracy of 87%.

The rest of this article is organized as follows: Section 2 provides an overview and enlists the categories of phishing detection. Section 3 identifies the approach used for logo-based detection. Section 4 provides a survey of all phishing-detection techniques reported thus far along with the findings. Section 5 presents the contribution to identify the brand logos with the proposed novel features. Section 6 provides the experiment setup details for training ML models. Section 7 summarizes the results of various ML algorithms using the prepared dataset of branded logos followed by our conclusion of the work in Section 8.

2. Overview of Phishing Attacks

Phishing attacks often begin with a malicious link that convinces victims to visit a deceptive website where they are duped into giving critical information (e.g., secret codes, data regarding the financial account and security numbers). This confidential data can then be used against the victim in the future. Through imitating genuine electronic communications, phishing attacks are frequently performed with the goal of stealing user passwords associated with their financial data. E-commerce, banking and informational sites are the most common targets of phishing attacks.

According to Das et al. [6], a phishing attempt consists of three parts: To begin, the mailer uses botnets to send out a large number of phishing emails with enticing subject lines and malicious attachments, photos and links. When a user clicks on these links or attachments in the email, they are redirected to a fake website, called a collector, where users are requested to submit personal information, such as their login passwords and bank and card information. Finally, the cashier applies the gathered credentials to complete a financial transaction from the victim's account.

Email phishing [7], often known as misleading phishing, is the most popular type. An attacker sends hundreds of spam emails in this type of attack. Even a single successful attack can result in a large amount of sensitive data. To make the message appear official, the phishing email uses the same language, typefaces, logos and signatures as real emails. They frequently compel users to act by using language, such as "expiring of card, password or membership" to create a sense of urgency. Although the URL inside the email appears to be close to the original site, it usually reflects a domain name or additional sub-domains that contain one or more misspelled characters.

Spear phishing [8,9] is a more advanced type of phishing that targets a specific individual or company. It necessitates a thorough understanding of a company—particularly its privileged personnel hierarchy. At the administrative level, perceiving a specific individual's username and password can aid in the initiation of an assault on the entire business. Whaling refers to spear phishing attacks launched at the top of an organization's staff hierarchy.

Classification of Various Phishing Attack Detection Mechanisms

Figure 1 depicts a broad classification of phishing-detection mechanisms [2,10]. The working principle of each detection mechanism category is provided briefly as follows:

- URL-scan-based approach [11]: These mechanisms scan the suspicious URL to parse numerous aspects of interest in order to evaluate their pattern for detecting anomalies related to phishing sites. This approach discovers possible phishing URLs by constructing various combinatorial URLs from current legitimate URLs and determining whether they exist and are involved in phishing-related activity over the internet.
- Content-based approach: The content-based technique identifies keywords and patterns in the requested email text and/or Uniform Resource Locator (URL) of the entire website, its components and document object model. The extracted contents of the web-page are examined and used for detection.
- Heuristic-learning-based approach [2]: This approach is based on the detection of anomalies in a website using a set of heuristics developed over a long-term observation. If a reported website has one or more anomaly patterns, then existing heuristics are used to detect them. These techniques extract a set of elements, such as text, images and URL-specific information. Finally, the anomaly is detected using the set of defined heuristics and applying the rules or thresholds obtained from the assimilating algorithms.
- Visual-similarity-based approach [12,13]: This approach uses a visual similarity score between web-pages. The score of similarity between these sites is used to detect phishing. Logos, photos, font size and type, alignment, text location, etc. are examples of the elements used for calculating the similarity score. The goal is to determine whether a reported site and a related popular/legitimate site have visual similarity.
- Blacklisting approach [14]: These approaches maintain a database of previously detected phishing sites/links, URLs and domains. The URLs reported by user comments, and trusted third parties are used to create the blacklist. A freshly arrived link or URL is checked against the blacklisted sites for a better match. When a match with a higher similarity ratio is identified, it is flagged to be a phishing web-site and added to the blacklist.
- Machine Learning (ML) and Hybrid Approach [15]: To enhance efficiency, these strategies integrate one or more of the aforementioned techniques as well as other detection techniques. ML-based hybrid techniques are becoming more popular as a result of their capacity to infer new prospective phishing patterns from existing ones. These strategies make use of a variety of algorithms to fine-tune classification and increase phishing detection precision.

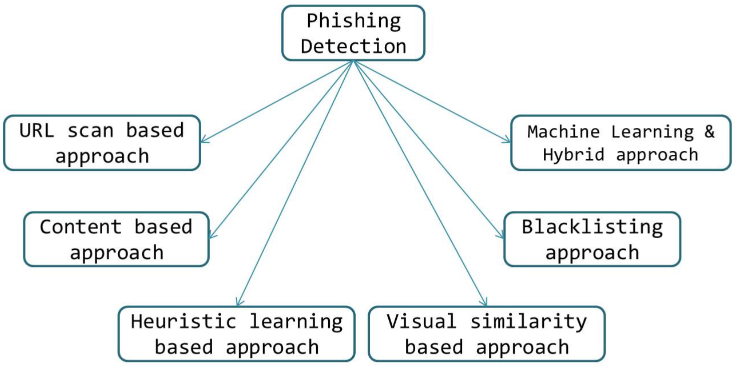

Phishing Detection
URL scan based approach
Content based approach
Heuristic learning based approach
Visual similarity based approach
Blacklisting approach
Machine Learning & Hybrid approach

Figure 1. Categorization of phishing attack detection [2].

3. Logo-Based Phishing Detection

The visual-similarity-based approach is the most challenging approach, yet the most effective one to detect phishing. This is because an attacker needs to achieve visual perfection in its trap in order to convince a victim. That is why they have to place all user trusted elements in the right place. Since the logo is the most visually verified element to authenticate a popular brand, most of the researchers in the field of visual-similarity-based approach for phishing detection focus on checking for a genuine logo of a popular brand on a suspicious page.

Another reason is that reputed firms also embed unique pattern in their logo to differentiate it from other brands in the same domain. Currently, the use of logo features along with one or more ML algorithm to test and train the features is the topmost approach practiced in this category. Figure 2 shows an example of a phishing form and a genuine real form for Google login. The attacker is using the old version logo of Google to make the page similar to original one. In this case, a logo-based detection mechanism can efficiently identify the use of the Google logo in the page and help to detect the phishing page.

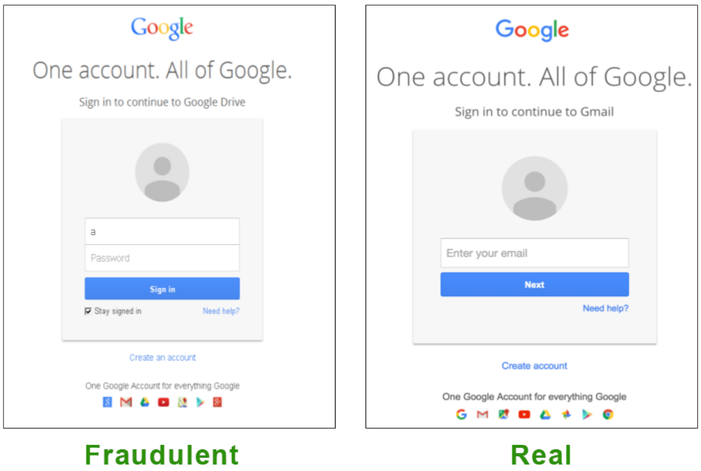

Google
One account. All of Google.
Sign in to continue to Google Drive
a
Password
Sign in
Stay signed in
Need help?
Create an account
One Google Account for everything Google
Google
One account. All of Google.
Sign in to continue to Gmail
Enter your email
Next
Need help?
Create account
One Google Account for everything Google
Fraudulent
Real

Figure 2. Example of phishing of the Google login page (source: https://images.google.com/, accessed on 10 August 2022).

Figure 3 shows a generic approach of a logo-based phishing-detection mechanism. Initially, the logo is identified and extracted using various boundary-detection mechanisms [16]. Then, various features of the logo, such as the color value, width and length, are extracted. Using the underlying detection algorithm, the logo-based phishing mechanism would respond whether the page is a phishing or genuine page. If the page is found to be a phishing page, it is blocked and added to the blacklist.

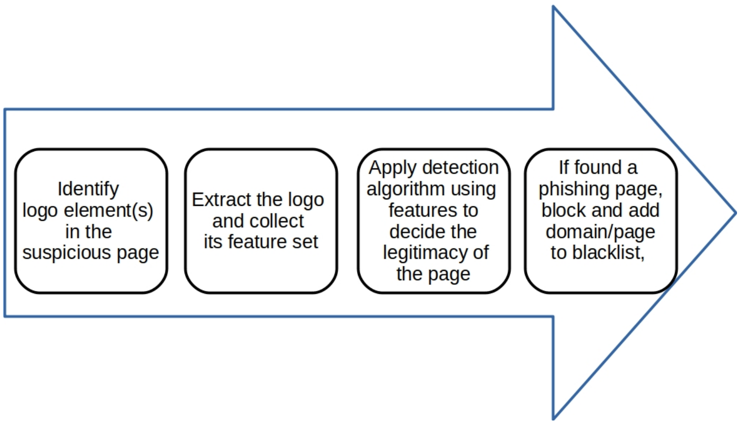

Identify logo element(s) in the suspicious page
Extract the logo and collect its feature set
Apply detection algorithm using features to decide the legitimacy of the page
If found a phishing page, block and add domain/page to blacklist,

Figure 3. Basic-logo-based phishing web-page detection.

4. Literature Survey

A huge number of contributions have been made in the field of detection mechanisms in each category of phishing detection. The major ones are the URL-scan-based approach along with the use of ML algorithms. In this section, we cover a few of the recent works of all the categories as reported in the literature.

Das et al. [6] inferred that the hybrid form or combination of multiple phishing-detection techniques outperform traditional approaches in terms of accuracy. Despite computational overhead, ML-based approaches achieve better accuracy and lower false positive rates.

Chiew et al. [17] used basic color characteristics to represent a logo image and verified the identity consistency between the true and the projected identity of a website to detect phishing.

Varshney et al. [5] analyzed various novel strategies and significant work proposed in the area of phished website detection and depicted their respective flaws and virtue.

Biancho et al. [18] proposed a phishing-detection technique using deep-learning model. They first utilized a logo region proposal for logo identification followed by applying Convolutional Neural Network (CNN) for logo classification. They claimed to achieve 96% accuracy with their proposed model.

Yao et al. [19] proposed a Faster Region-based Convolutional Neural Network (Faster R-CNN)-based method for small-scale logo recognition along with URL-based code for phishing attack detection.

Peng et al. [20] proposed a method that focuses on the attack's natural language text and semantic analysis of the text using Natural Language Processing (NLP) techniques to detect incorrect remarks that are plausible for a phishing attack.

Ding et al. [21] proposed a compound technique consist of keyword search, applying heuristics, followed by using a logistic regression classifier to identify phishing URL. First, a keyword search is performed for the suspicious page in the baidu search engine and the top ten domains are matched. If matching fails, then the heuristics are used to check for genuineness of the page. If the page is still not categorized as genuine, it is fed to a logical regression classifier to classify it as a phishing site or as genuine.

Rao and Pais [22] proposed a detection mechanism called Jail-Phish. Jail-Phish aims to improve the detection accuracy of search engine for detecting a phishing site on a compromised server. The detection procedure includes the comparison of the suspicious site and the genuine sites with matched domain. The matching results in a similarity score between the suspicious site and the domain, and this score is used to detect the phishing page.

Bozkir and Aydos [16] proposed LogoSENSE, which is a Histogram Orient Gradient (HOG)-based logo detection method for phishing web page recognition. They used HOG to obtain visual representations of target brand logos in a scale invariant fashion. They used max-margin loss-equipped Support Vector Machine (SVM) classifier for classification of a phishing logo.

Azeez et al. [23] provided an automated white-list technique for identifying phishing attacks. Here, the visible link and the actual link are analyzed to determine the white-list. The similarities of the known legitimate site are calculated by binding the domain name with the contents of the whitelist. Next, these are matched with the IP address to obtain the final similarity score. This information helps to decide the membership of a hyperlink in the whitelist. The authors claimed to achieve an average accuracy of 96.17%.

Lin et al. [24] proposed a hybrid deep-learning system, Phishpedia, which addresses identity logo recognition and matching logo variants to detect phishing logos. This method provides better accuracy in phishing logo detection but is prone to false-positives in many scenarios.

Butnaru et al. [25] proposed a light-weight URL-based phishing-detection mechanism. They claim to use a unique combination of features to train and detect phishing attack using ML algorithms. The important point is they used only the URL to extract all the features for phishing detection.

Gupta et al. [26] provided an anti-phishing solution for real-time environments. They used a small number of features (nine) to achieve a higher detection accuracy, which makes the technique suitable for resource-constrained devices. They used four ML-based classifiers to train and test the selected features and claimed to have nearly 99% accurary.

Moedjahedy et al. [27] combined three correlation methods: Spearman, Power Predictive Score (PPS) and Maximal Information Coefficient (MICe) for feature selection and recursive feature elimination method to minimize the number of features to select only the use features for phishing detection without compromising the accuracy. They used a number of ML algorithms to verify the accuracy of selected features under different scenarios.

Ruofan et al. [28] proposed a mechanism called PhishIntention, which extracts the intention of phishing in a web-page by visually extracting brand intention and interacts with the web-page to gather its intention regarding phishing. They extracted the abstract web-page layout and determined the phishing intention from it. They used a heterogeneous system of a deep-learning-vision-model to implement their work. The summary of the phishing-detection techniques is tabulated in Table 1. The list of review articles referenced for our study and analysis of phishing-detection techniques are provided in Table 2.

Table 1. Summary of phishing-detection techniques.

Sl. No.	Title	Authors	Year	Keynotes
1	Utilization of website logo for phishing detection	Chiew et al. [17]	2015	Content-based image retrieval feature from Google image database and used image similarity for phishing detection
2	Deep learning for logo recognition	Biancho et al. [18]	2017	CNN-based Deep Learning model for logo detection
3	Detecting Phishing Attacks Using NLP and ML	Peng et al. [20]	2018	NLP-based approach to analyse text and content of webpages to detect phishing attack.
4	Deep Learning for Phishing Detection	Yao et al. [19]	2018	Used both URL and logo to propose a phishing-detection mechanism
5	Jail-Phish: An improved search-engine-based phishing detection system	Rao and Pais [22]	2019	Website contents, such as the logo, favicon, images and text, for phishing detection
6	A keyword-based combination approach for detecting phishing webpages	Ding et al. [21]	2019	A compound approach of search engine, heuristics rules and LR classifier is used to detect a phising site
7	LogoSENSE: A companion HOG-based logo detection scheme for phishing brand recognition	Bozkir and Aydos [16]	2020	Used Histogram Oriented Gradients to obtain visual representation of logo image and hence recognize phishing web pages

Table 1. *Cont.*

Sl. No.	Title	Authors	Year	Keynotes
8	Adopting automated whitelist approach for detecting phishing attacks	Azeez et al. [23]	2021	Came up with an automated whitelist-based approach to detect phishing web-pages by using hyperlink parameters
9	Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages	Lin et al. [24]	2021	Presented a hybrid deep-learning system that matches logo images of the same brand with the website logo to detect phishing
10	Towards Lightweight URL-Based Phishing Detection	Butnaru et al. [25]	2021	Used only URL-based features to train and detect phishing using ML algorithms.
11	A novel approach for phishing URLs detection using lexical-based machine learning in a real-time environment	Gupta et al.	2021	Used nine features of an URL to train and detect a phishing URL using ML algorithms
12	CCrFS: Combine Correlation Features Selection for Detecting Phishing Websites Using Machine Learning	Moedjahedy et al. [27]	2022	Used combine approach of correlation and recursive feature elimination process to limit the number of features to detect phishing detection.

Table 2. Summary of survey works on phishing detection.

Sl. No.	Title	Authors	Year	Keynotes
1	A survey and classification of web phishing detection schemes	Varshney et al. [5]	2016	Presented a comprehensive analysis of phished website detection and outlining advantages and disadvantages
2	Phishing Detection: Analysis of Visual Similarity Based Approaches	Jain et al. [13]	2017	Presented a comprehensive and comparative analysis of phishing attacks and detection using visual-similarity-based approaches
3	Systematization of Knowledge (SoK): A Systematic Review of Software-Based Web Phishing Detection	Dou et al. [29]	2017	Systemized study of phishing-detection techniques and analysis.
4	SoK: A Comprehensive Reexamination of Phishing Research From the Security Perspective	Abhisha Das et al.	2020	Re-examined phishing research works, categorizing the existing works based on attack vectors and examination of properties and features for phishing detection
5	Phishing Attacks Survey: Types, Vectors and Technical Approaches	Alabdan et al. [30].	2020	A comprehensive survey of attack properties and various detection mechanisms.
6	Exquisite Analysis of Popular Machine Learning–Based Phishing Detection Techniques for Cyber Systems	Meenakshi Das et al. [6]	2021	Performed an exquisite analysis various machine-learning-based phishing-detection techniques, which includes analysis and taxonomy used in various methods
7	Phishing Website Detection With Semantic Features Based on Machine Learning Classifiers: A Comparative Study	Almomani et al. [31]	2022	Classified the ML-based detection techniques and performed an experiment based accuracy comparison of the ML-based techniques
8	A survey of phishing attack techniques, defence mechanisms and open research challenges	Jain et al. [32]	2022	Emphasized on distribution procedure of phishing attack, highlighted the consequences of phishing threats and enlisted various challenges involved in phishing detection

5. Proposed Work

In this section, we present the proposed logo-based phishing-detection mechanism. Figure 4 shows two different logos of Google used in different periods of time. Even though both logos differ in their font style, the color combinations of both the logo images is nearly same. Moreover, in addition to color combinations, the distribution of these colors in the logo is also similar to Google.

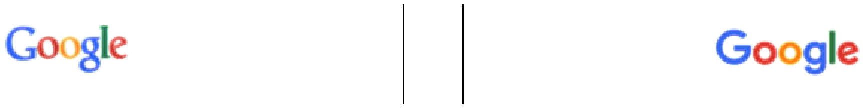


Figure 4. Fraudulent vs. genuine google logo.

The proposed logo-detection mechanism uses this concept to extract the features and identify a logo. The novelty of the proposed mechanism is the use of hue value density and the average relative distance of each pixel with a given hue value from their centroid position as the features for detection. These features can uniquely represent any logo because the logos are generally designed by the combination of various colors and shapes in it.

If coincidentally, two or more logo have the same combination of hue values and density, then it is the distribution of the pixels with the same hue value in the image that distinguish one logo from another. The overall work of detection of phishing logos involves a number of steps, such as logo image extraction from web-page or email, classifying the logo as one of the brand logo to confirm the email or web-page as phishing. A number of standard the logo extraction [16] and post detection blacklisting mechanism are already in use.

Therefore, in this paper, we mainly focus on improving the precision of detection of the logo and pivot the attention for extracting the features of the logo image for classifying the phishing logo. The proposed phishing-detection mechanism have two phases as shown in Figure 5: (i) Training Phase and (ii) Detection Phase. The *Training Phase* begins with extraction of the logo from the website and then it is converted to Hue, Saturation and Value (HSV) format. In the next step, the hue value centroid position and pixel ratio is calculated for each distinct hue value.

Now, the nearest hue values are clustered to approximate the analysis of presence of different colors in the logo. For each hue cluster, the hue density distribution is computed by obtaining the pixel compactness with respect to the relative distance from the hue centroid. Finally, the dataset is prepared from the retrieved image features, and ML algorithms are applied on it to train the model for detecting phishing logos. In the *Detection Phase*, the logo is extracted from the provided web-page or email. The logo image is then pre-processed as provided in the training phase to obtain hue-based information from the image. Then, an ML algorithm is used to find the best match. On obtaining a best match, it is detected as a phishing logo.

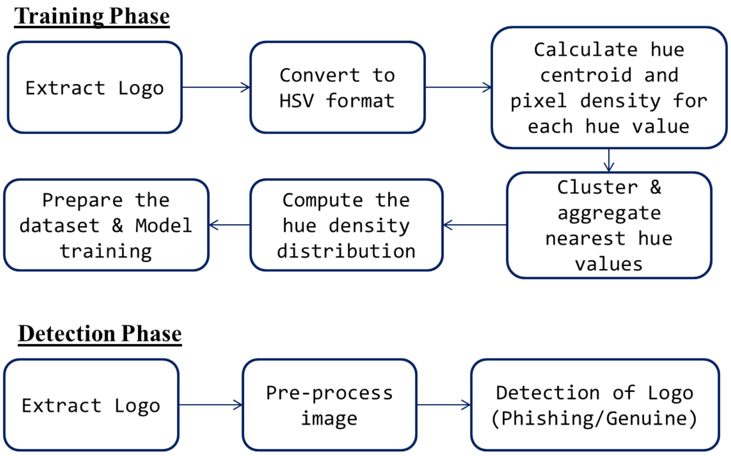

Training Phase
Extract Logo
Convert to HSV format
Calculate hue centroid and pixel density for each hue value
Cluster & aggregate nearest hue values
Compute the hue density distribution
Prepare the dataset & Model training
Detection Phase
Extract Logo
Pre-process image
Detection of Logo (Phishing/Genuine)

Figure 5. Flow diagram of the proposed approach.

Hue refers only to the pure spectrum color names found on the color wheel: red, orange, yellow, green, blue and violet [33]. The hue value representation uses a range of 0–360 integer value to represent the original pigmentation of a pixel as shown in Figure 6. A single integer representation of a color empowers us to extract and use other associated features of the logo image for efficient detection. Once an image I is transformed to its equivalent hue, saturation and value (HSV) format, say I_{hsv}, the list of distinct hue values $HL_{I_{hsv}}$ present in the image is obtained. For each hue value h_i in the $HL_{I_{hsv}}$, the pixel ratio and the centroid position of all the pixels sharing the hue value h_i is computed.

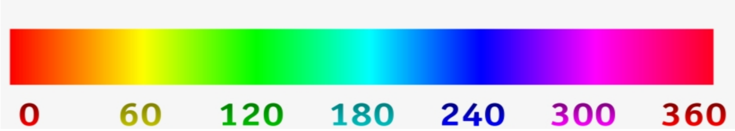


Figure 6. Representation scale of hue values.

Definition 1. *The **pixel ratio** of a hue value h is defined as the ratio of the number of pixels having hue value h in the image I_{hsv} and the total number of pixels in the image I_{hsv}*

Definition 2. *If $p_1, p_2, \ldots, p_m$ are the m number of pixels having same hue value h in an image I_{hsv}, then the centroid of h is given by $\left(C_x^h, C_y^h\right)$, where*

$$\begin{aligned} C_x^h &= \frac{x_{p_1} + x_{p_2} + \cdots + x_{p_m}}{m}, \textit{ and} \\ C_y^h &= \frac{y_{p_1} + y_{p_2} + \cdots + y_{p_m}}{m} \end{aligned} \tag{1}$$

In the next step, the relative Euclidean distance of each pixel p_i is calculated from its hue centroid.

Definition 3. *The relative Euclidean distance between a pixel p and the centroid, C^h is defined as the ratio of the Euclidean distance between p and C^h and length of the side of the image that incident of the $line(p, C^h)$ with the smallest angle.*

It is clear from the Definition 3 that the relative distance ranges between 0 and 1. The reason behind using relative distance is that the proposed mechanism considered logos with different dimensions. The images are not transformed to the equivalent image of a given size to preserve the ratio of color and their shape in the original logo. The relative distance of a pixel from its hue centroid represents the relative position of the pixel in an image irrespective of its dimension.

Therefore, our proposed mechanism has the ability to extract this unique feature from logo images of different size. Since the relative distance of any pixel from its hue centroid ranges between 0 and 1, we created approximately 10 groups of relative distance $\{0.1, 0.2, \ldots, 0.9, 1.0\}$ to maintain the count number of pixels in each group. The count of pixels in each group reflects the positional distribution of the pixels for a given hue value. The final representation of the hue features of a given image is defined in the form of a dictionary data structure as follows:

$$\begin{gathered} \{h : \{\text{``}pixelratio\text{''} : pixelratioval, \text{``}C\text{''} : (x, y), \\ 0.1 : value_1, 0.2 : value_2, \ldots, 0.9 : value_9, 1.0 : value_{10}\}\} \end{gathered} \tag{2}$$

where $value_1$, $value_2$, …, $value_{10}$ are the count of pixels with a given hue value having relative distance of 0.1, 0.2, …, 1.0, respectively.

Handling individual hue values for realistic logos would be cumbersome because the logos contains large range of color values. However, majority of them are nearby gradient patterns and visually identical to the same color. Therefore, the nearby hue values were clustered with magnitude difference of small scale to represent gradient patterns in a logo and group nearly similar colored hue values to reduce the randomness in the data. For this purpose, we used Agglomerative Hierarchical Clustering [34]. The algorithms for feature selection and training ML models is provided in Algorithms 1 and 2, respectively.

Algorithm 1: The algorithm for logo feature extraction.

```
Input: Set of logo images $\mathbf{I}$
Output: Set of feature dictionary $\mathbf{F}$
for each $I$ in $\mathbf{I}$ do
    $I_{hsv}$=BGRtoHSV($I$);
    $\mathbf{I_{hsv}} = \mathbf{I_{hsv}} \bigcup I_{hsv}$ ;
for each $I_{hsv}$ in $\mathbf{I_{hsv}}$ do
    $HL_{I_{hsv}} = extractHueValues(I_{hsv})$;
    $F$=createFeatureDict();
    for each hueval in $HueList_{I_{hsv}}$ do
        add hueval to $F$; $C$=computeCentriod(hue);
        add $C$ to $F$; for each pixel p with hueval do
            reldist=computeRelativeDistance($p$,$C$);
            add reldist to $F$;
    add $F$ to $\mathbf{F}$;
return $\mathbf{F}$;
```

Algorithm 2: The algorithm for training the ML models.

```
Input: Set of feature dictionary $\mathbf{F}$
Output: Set of ML training models $\mathbf{M}$
for each $F$ in $\mathbf{F}$ do
    $F$=AHClustering(F);
    $F_{vector}$=createFeatureVector($F$);
    add $F_{vector}$ to dataset LogoData;
for each mlalgo in MLAlgoList do
    m=TrainModel(mlalgo,$LogoData$);
    add $m$ to $\mathbf{M}$;
return $\mathbf{M}$;
```

To cite a simple example, we chose the logo of axis bank as shown in Figure 7. The distinct hue values, *distinctHue*, present in this logo are

```
distinctHue: [0, 5, 6, 15, 30, 90, 98, 99, 100, 101, 102, 103, 104, 105,
              106, 107, 150]
```

Grouping the nearby hue values to a single cluster results in the following clustered hue values.

```
hueCluster: repHue  -> hueMembers
                4    ->  [0, 5, 6],
                15   ->  [15],
                30   ->  [30],
                90   ->  [90],
                102  ->  [98, 99, 100, 101, 102, 103, 104, 105, 106, 107],
                150  ->  [150]
```

For a hue value of 102, the relative distance distribution of pixels is given by

```
hueValue: 102
pixratio: 16.0
pixels: 596

relativeDistanceDistribution: [0.0: 0,  0.1: 0,  0.2: 19, 0.3: 30,
                               0.4: 21, 0.5: 20, 0.6: 2,  0.7: 4,
                               0.8: 2,  0.9: 2,  1.0: 0]
```

Here, *hueValue* represents the cluster aggregated hue value, *pixratio* represents the percentage of pixels of the total logo image depicted by the *hueValue*, *pixels* represents the number of pixels in the image that is depicted by the *hueValue*. The *relativeDistanceDistribution* represents the percentage of pixels at relative distance of 0.0 to 1.0 from the hue value centroid. Using these above features, the dataset is prepared for phishing logo detection.

Figure 7. Example: Axis Bank logo.

6. Experiment Overview

We used a system with Intel(R) Core(TM) i7-6200U CPU @ 2.30 GHz processor with 8 GB of RAM. The operating system used is Windows 10. The major python libraries that are used for the experiment are numpy, cv2 and sklearn. Initially, we gathered logo images from popular phishing prone websites. For pre-processing of an image, first the image was converted to its HSV representation. We extracted the hue values, and the nearby hue values were clustered with magnitude difference of 5 to represent gradient patterns in a logo.

In the next step, the centroid point in the logo image for each hue value is calculated along with the percentage of pixels for that hue value. After derivation of the centroid, the pixel distribution for each hue value is computed based on its relative distance from the centroid, which signifies the pattern of the hue color distribution from hue centroid in a particular logo. Finally, the feature dataset is prepared and selected ML algorithms are applied on the data to generate a model.

Dataset Description & Analysis

For a better efficiency and remarkable phishing detection, we considered most popular and prominent phishing prone websites from a South-Asian perspective. These include popular banking websites, social networking websites and other popular phishing websites. Some sample logos of the dataset are highlighted in Figure 8.

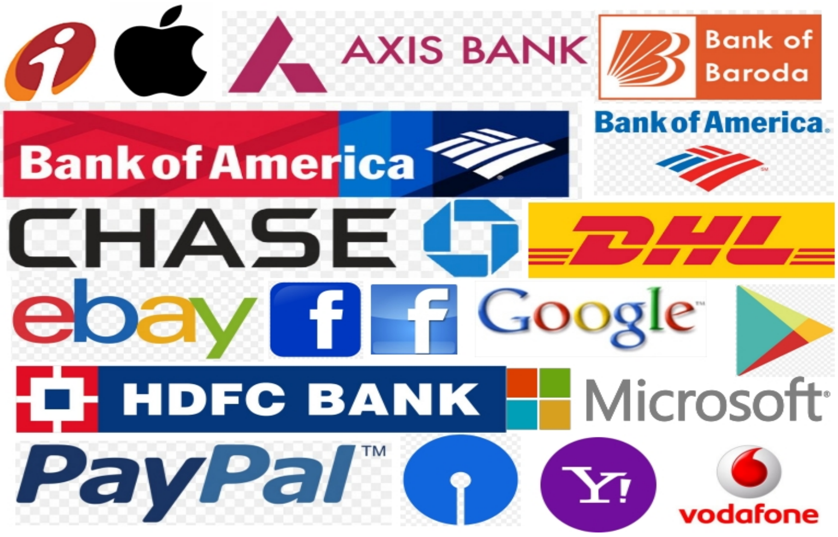


Figure 8. Various logos used in the dataset.

The dataset collection includes 21 distinct popular websites (as seen in Figure 9) with an aggregate of *48 different class* of logo images and a total 538 as logo count. For every distinct logo class, we have at least five logo images to reduce bias in the data.

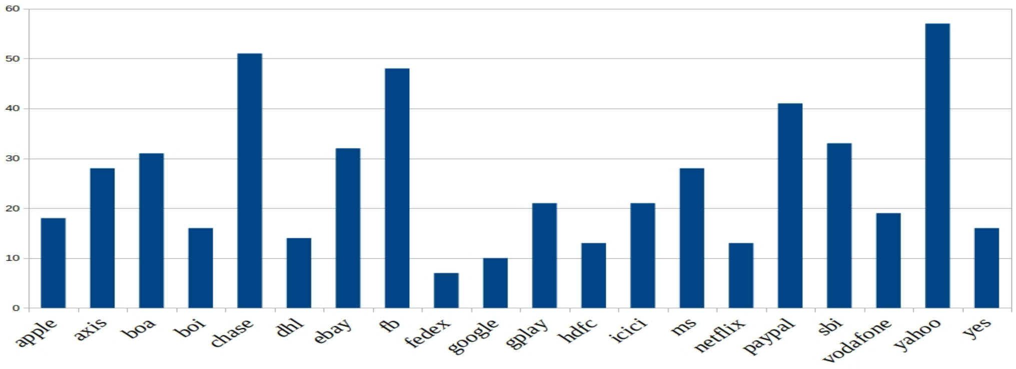


Figure 9. Class-wise logo image distribution.

A crucial decision is aroused during dataset generation is that how to standardize the number of distinct hue values per logo, i.e., the number of unique hue values to consider for each image representation in the dataset. It is observed from Figure 10 that the hue values per image range from 2 distinct hue values to 25 distinct hue values. Therefore, we considered 6 to 15 distinct hue values as standard for hue-representation of an image. The Agglomerative Hierarchical Clustering [34] is applied in bottom-up manner to cluster the distinct hue values until the considered hue value range is achieved, where the hue values with a magnitude difference of 5 are clubbed together to form a cluster.

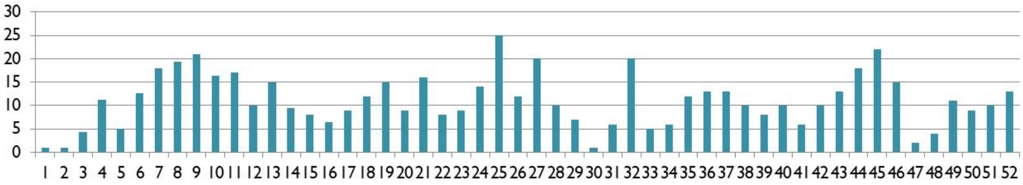


Figure 10. Number of distinct hue values per logo.

7. Results

The generated feature dataset was evaluated with two different types of data formation from the same set of images. The first dataset was generated by considering the each variant of logos of the same brand as separate class without grouping into a single class. The second dataset is generated by grouping all variant classes of the same brand into a single class. We considered five best performing multi-class classifiers. These were Decision Tree(D Tree), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Gaussian Naive Bayes (GNB) and Ensemble Random Forest (ERF) multi-class classifiers [35,36]. We varied the considered hue values from 6 to 15 clustered prominent values. For achieving the classification accuracy, we considered the randomized train-test split size to be 75:25 and the mean of the accuracy performed over 100 evaluation rounds.

Figure 11 compares the detection accuracy of ML techniques without clubbing. The ERF multi-class classifier and D Tree were the best performer in every scenario clocking around average accuracy 85%, with peak accuracy of 87.21% for 8, 11 and above 13 prominent hue color values. This performance was marginally degraded with the reduction of the hue values. The D Tree classifier and KNN have marginally lower accuracy than ERF classifier. The SVM and GNB classifiers were unable to perform better for this dataset, where GNB multi-class classifier has the lowest accuracy. These three classifiers performed in a similar manner when ranging from 6 to 12 hue values and randomly deteriorated for values above the range.

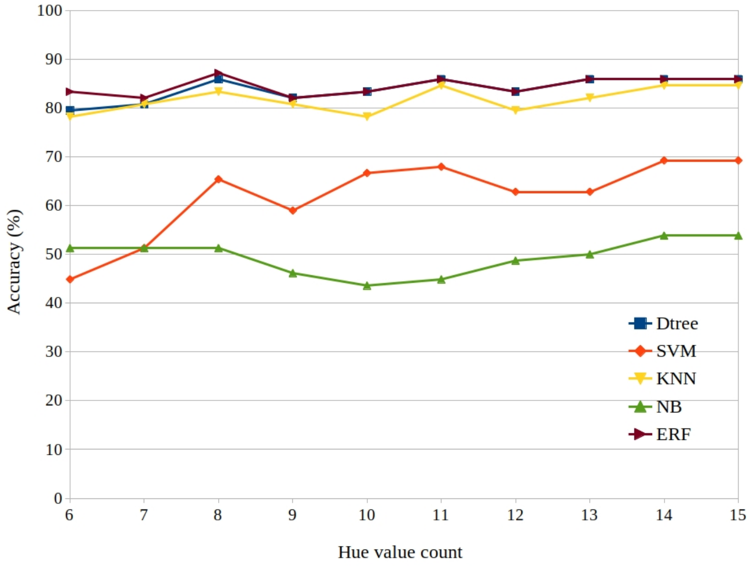


Figure 11. Accuracy without clubbing.

Figure 12 shows the accuracy of the ML algorithms with clubbing. The accuracy of all the classifiers increased except for GNB. The ERF was still the best performer with a steady average accuracy of 86%. Moreover, the highest accuracy of our experiment is achieved by ERF with 88% for 7, 8 and 12 hue value counts. Surprisingly the performance of KNN was lower compared to no clubbing dataset, whereas the SVM accuracy has increased significantly for this dataset. The GNB remained as the least performer for both variants of the dataset.

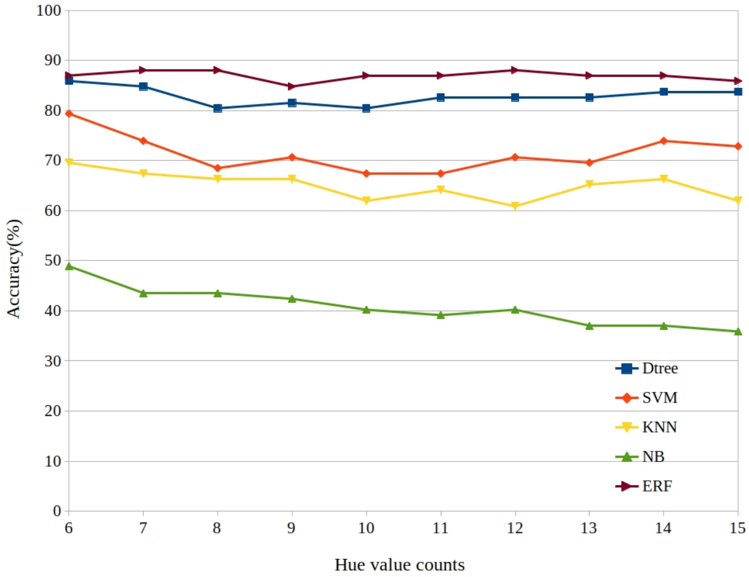


Figure 12. Accuracy with clubbing.

The overall accuracy comparison is provided in Figure 13. To summarize, the least performer was GNB with approximately 40% accuracy. The ERF multi-class classifier was the steady performer with approximately 87% accuracy on aggregate. This figure is equally comparable with that of the LogoSense detection mechanism by Bozkir et al. [16], which has achieved 93.5% accuracy considering only an individual class of images for logo-based phishing detection.

From the confusion matrix shown in Figure 14, nearly all the classes are correctly classified. The diagonal value represents the correct classification. The axes represent the actual and predicted class IDs, where the classes can be determined using Table 3. We infer from the confusion matrix that some of the *Yahoo* logo images were wrongly classified as *Facebook* logos due to similarity in both color and pattern.

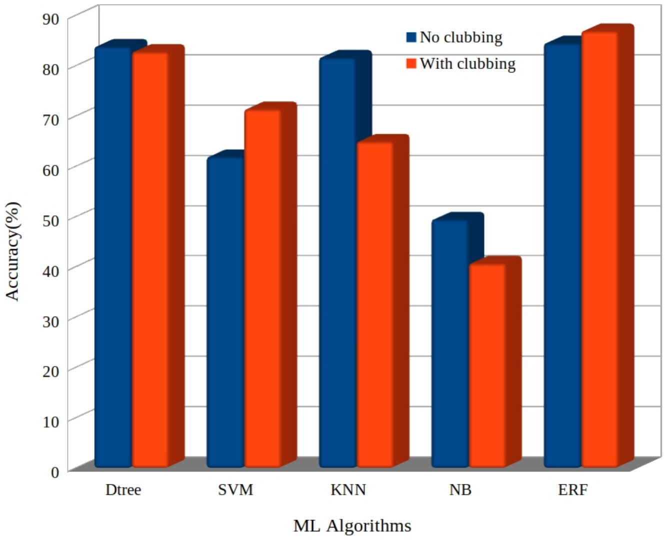
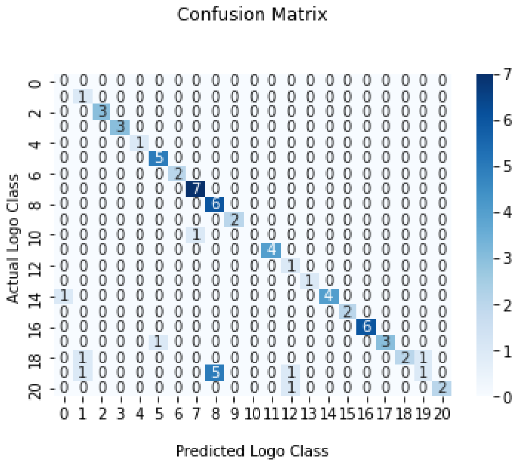


Figure 13. Overall accuracy performance.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
18	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0
	0	1	0	0	0	0	0	0	5	0	0	0	1	0	0	0	0	0	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2

Figure 14. Confusion matrix for the best performance.

Table 3. Class name and class identifiers.

Class ID	0	1	2	3	4	5	6
Class Name	apple	axis	boa	bob	boi	chase	dhl
Class ID	7	8	9	10	11	12	13
Class Name	ebay	fb	fedex	google	gplay	hdfc	icici
Class ID	14	15	16	17	18	19	20
Class Name	ms	netflix	paypal	sbi	vodafone	yahoo	yes

Table 4 compares the performance metrics and the accuracy of proposed work with logoSENSE [16]. logoSENSE used almost higher range parameters, such as dataset size and epochs, and achieved better accuracy compared to the proposed work. The parameters where the proposed work had an advantage over logoSENSE are the number brands and number of classes used for the experiment. In the proposed work, we used a higher variation of logo images for a single brand leading to 48 classes for a small dataset compared to logoSENSE, which used only 16 classes for a large dataset of size around 3800.

Table 4. Comparison of the proposed work with LogoSENSE [16].

Parameters	LogoSENSE	Proposed Work
OS	Ubuntu 18.04	Windows 10
Processor	Intel i7	Intel i5
RAM size	24 GB	8 GB
Dataset	Phishtank, Phishbank	Own
Training Dataset Size	3060	432
Testing Dataset Size	864	106
No. of epochs	1000	500
Number of brands	15	21
Number of classes	16	48
Overall accuracy	93.5%	87%

8. Conclusions and Future Work

In this paper, we proposed a novel logo-based phishing detection approach that uses hue values and density distribution of hue values from each hue value centroid of an image as features to improve the precision of detection of genuine logo from the phishing logo. We used 21 distinct brand of logos and determined the accuracy to be 87% in correctly classifying the genuine logos. For performance adjustment and finding the optimal number of hue values, we varied the number of hue values from 6 to 15.

The best accuracy was achieved using the Ensemble Random Forest multi-class classifier. We also found that the relative distribution of hue value density from the hue centroid helped in more precise detection of the genuine logo. Overall, the Ensemble Random Forest classifier proved to be the most accurate performer followed by the D Tree classifier. For our prepared dataset, the K-Nearest Neighbor, Support Vector Machine and Gaussian Naive Bayes classifiers did not perform well.

For further continuation of this work, a multi-layer phishing-based detection application can be made that can combine various strategies for phishing attack detection along with logo-based detection in a pipeline for precision. A multi-layer logo-based detection system can be made using these novel hue values, where the first layer segregates and classifies to which logo it belongs, and the next subsequent layers determine the similarity percentage of that logo with the trained genuine logo images for better precision. Another approach would be to use various other parameters of the websites/e-mails along with the hue values of the logo images to improve the overall detection accuracy.

Author Contributions: Conceptualization, A.K.M., D.P.; Methodology, A.K.M.; software, P.P., A.K.M.; validation, P.P., D.P.; formal analysis, A.K.M. investigation, D.P.; resources, P.P., A.K.M.; data curation, P.P.; writing–original draft preparation, P.P.; writing—review and editing, A.K.M. and D.P.; visualization, D.P.; supervision, A.K.M.; project administration, A.K.M., D.P.; funding acquisition, NA. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ramzan, Z. Phishing attacks and countermeasures. In *Handbook of Information and Communication Security*; Stavroulakis, P., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 433–448.
2. Mishra, A.K.; Tripathy, A.K.; Saraswathi, S.; Das, M. Prevention of Phishing Attack in Internet-of-Things based Cyber-Physical Human System. In *High Performance Vision Intelligence*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 15–32.
3. Sahoo, B.; Rath, S.; Puthal, D. Energy efficient protocols for wireless sensor networks: A survey and approach. *Int. J. Comput. Appl.* **2012**, *44*, 43–48.
4. Bhatt, P.; Thakker, B. A novel forecastive anomaly based botnet revelation framework for competing concerns in internet of things. *J. Appl. Secur. Res.* **2021**, *16*, 258–278. [CrossRef]
5. Varshney, G.; Misra, M.; Atrey, P.K. A survey and classification of web phishing detection schemes. *Secur. Commun. Netw.* **2016**, *9*, 6266–6284. [CrossRef]
6. Das, M.; Saraswathi, S.; Panda, R.; Mishra, A.K.; Tripathy, A.K. Exquisite Analysis of Popular Machine Learning–Based Phishing Detection Techniques for Cyber Systems. *J. Appl. Secur. Res.* **2021**, *16*, 538–562. [CrossRef]
7. Gangavarapu, T.; Jaidhar, C.; Chanduka, B. Applicability of machine learning in spam and phishing email filtering: Review and approaches. *Artif. Intell. Rev.* **2020**, *53*, 5019–5081. [CrossRef]
8. Halevi, T.; Memon, N.; Nov, O. Spear-phishing in the wild: A real-world study of personality, phishing self-efficacy and vulnerability to spear-phishing attacks. *Phishing-Self-Effic. Vulnerability Spear-Phishing Attacks* **2015**, *2015*. [CrossRef]
9. Bullee, J.W.; Montoya, L.; Junger, M.; Hartel, P. Spear phishing in organisations explained. *Inf. Comput. Secur.* **2017**, *25*, 1–21. [CrossRef]
10. Zuraiq, A.A.; Alkasassbeh, M. Phishing detection approaches. In Proceedings of the 2019 Second International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; pp. 1–6.
11. Almeida, R.; Westphall, C. Heuristic Phishing Detection and URL Checking Methodology Based on Scraping and Web Crawling. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–10 November 2020; pp. 1–6.
12. Medvet, E.; Kirda, E.; Kruegel, C. Visual-similarity-based phishing detection. In Proceedings of the fourth International Conference on Security and Privacy in Communication Netowrks, Istanbul Turkey, 22–25 September 2008; pp. 1–6.
13. Jain, A.K.; Gupta, B.B. Phishing detection: Analysis of visual similarity based approaches. *Secur. Commun. Netw.* **2017**, *2017*, 1–21. [CrossRef]
14. Hara, M.; Yamada, A.; Miyake, Y. Visual similarity-based phishing detection without victim site information. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Cyber Security, Nashville, TN, USA, 30 March–2 April 2009; pp. 30–36.
15. Kumar, A.; Chatterjee, J.M.; Díaz, V.G. A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 486. [CrossRef]
16. Bozkir, A.S.; Aydos, M. LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition. *Comput. Secur.* **2020**, *95*, 101855. [CrossRef]
17. Chiew, K.L.; Chang, E.H.; Tiong, W.K. Utilisation of website logo for phishing detection. *Comput. Secur.* **2015**, *54*, 16–26. [CrossRef]
18. Bianco, S.; Buzzelli, M.; Mazzini, D.; Schettini, R. Deep learning for logo recognition. *Neurocomputing* **2017**, *245*, 23–30. [CrossRef]
19. Yao, W.; Ding, Y.; Li, X. Deep learning for phishing detection. In Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, VIC, Australia, 11–13 December 2018; pp. 645–650.
20. Peng, T.; Harris, I.; Sawa, Y. Detecting phishing attacks using natural language processing and machine learning. In Proceedings of the 2018 IEEE 12th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 31 January–2 February 2018; pp. 300–301. [CrossRef]
21. Ding, Y.; Luktarhan, N.; Li, K.; Slamu, W. A keyword-based combination approach for detecting phishing webpages. *Comput. Secur.* **2019**, *84*, 256–275. [CrossRef]
22. Rao, R.S.; Pais, A.R. Jail-Phish: An improved search engine based phishing detection system. *Comput. Secur.* **2019**, *83*, 246–267. [CrossRef]
23. Azeez, N.A.; Misra, S.; Margaret, I.A.; Fernandez-Sanz, L.; Abdulhamid, S.M. Adopting automated whitelist approach for detecting phishing attacks. *Comput. Secur.* **2021**, *108*, 102328. [CrossRef]
24. Lin, Y.; Liu, R.; Divakaran, D.M.; Ng, J.Y.; Chan, Q.Z.; Lu, Y.; Si, Y.; Zhang, F.; Dong, J.S. Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Virtual Event, 11–13 August 2021; pp. 3793–3810.

25. Butnaru, A.; Mylonas, A.; Pitropakis, N. Towards Lightweight URL-Based Phishing Detection. *Future Internet* **2021**, *13*, 154. [CrossRef]
26. Gupta, B.B.; Yadav, K.; Razzak, I.; Psannis, K.; Castiglione, A.; Chang, X. A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Comput. Commun.* **2021**, *175*, 47–57. [CrossRef]
27. Moedjahedy, J.; Setyanto, A.; Alarfaj, F.K.; Alreshoodi, M. CCrFS: Combine Correlation Features Selection for Detecting Phishing Websites Using Machine Learning. *Future Internet* **2022**, *14*, 229. [CrossRef]
28. Liu, R.; Lin, Y.; Yang, X.; Ng, S.H.; Divakaran, D.M.; Dong, J.S. Inferring Phishing Intention via Webpage Appearance and Dynamics: A Deep Vision Based Approach. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Virtual Event, 11–13 August 2021.
29. Dou, Z.; Khalil, I.; Khreishah, A.; Al-Fuqaha, A.; Guizani, M. Systematization of Knowledge (SoK): A Systematic Review of Software-Based Web Phishing Detection. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2797–2819. [CrossRef]
30. Alabdan, R. Phishing Attacks Survey: Types, Vectors, and Technical Approaches. *Future Internet* **2020**, *12*, 168. [CrossRef]
31. Almomani, A.; Alauthman, M.; Shatnawi, M.T.; Alweshah, M.; Alrosan, A.; Alomoush, W.; Gupta, B.B.; Gupta, B.B.; Gupta, B.B. Phishing Website Detection With Semantic Features Based on Machine Learning Classifiers: A Comparative Study. *Int. J. Semant. Web Inf. Syst. (IJSWIS)* **2022**, *18*, 1–24. [CrossRef]
32. Jain, A.K.; Gupta, B. A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterp. Inf. Syst.* **2022**, *16*, 527–565. [CrossRef]
33. Ahn, J.S.; Lee, Y.K. Color distribution of a shade guide in the value, chroma, and hue scale. *J. Prosthet. Dent.* **2008**, *100*, 18–28. [CrossRef]
34. Bouguettaya, A.; Yu, Q.; Liu, X.; Zhou, X.; Song, A. Efficient agglomerative hierarchical clustering. *Expert Syst. Appl.* **2015**, *42*, 2785–2797. [CrossRef]
35. Qian, B.; Su, J.; Wen, Z.; Jha, D.N.; Li, Y.; Guan, Y.; Puthal, D.; James, P.; Yang, R.; Zomaya, A.Y.; et al. Orchestrating the development lifecycle of machine learning-based IoT applications: A taxonomy and survey. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–47. [CrossRef]
36. Rajora, S.; Li, D.L.; Jha, C.; Bharill, N.; Patel, O.P.; Joshi, S.; Puthal, D.; Prasad, M. A comparative study of machine learning techniques for credit card fraud detection based on time variance. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1958–1963.