



LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition

Ahmet Selman Bozkir, Murat Aydos

Department of Computer Engineering, Hacettepe University, Ankara, TURKEY

ARTICLE INFO

Article history:

Received 25 December 2019

Revised 5 April 2020

Accepted 21 April 2020

Available online 8 May 2020

Keywords:

Phishing

Logo Detection

Computer Vision

HOG

Machine Learning

ABSTRACT

With the advent of Internet and opportunities in e-commerce, a visual perception oriented cyber-attack so-called phishing has become one of the tremendous problems of the cyber world since it aims to access user credentials in order to gain illegal financial profit and steal sensitive personal data. In order to fight with this security threat, various studies using a different source of information such as URL, text content, DOM trees or visual features belonging to web pages have been utilized. Apart from other works, we propose a companion scheme to recognize brands of “zero hour” phishing web pages by localizing and classifying the target brand logos involved in page screenshots by solely use of computer vision methods in object detection manner. For this purpose, the features of Histogram of Oriented Gradients (HOG) have been employed to obtain visual representations of target brand logos in scale invariant fashion. In addition, throughout the classification, a max-margin loss equipped SVM classifier has been used in order to work with a low number of training images and to decrease the number of false positives. Moreover, we prepared a publicly available dataset having a total of 3060 training and 1979 unique phishing and legitimate web page/e-mail snapshots along with their bounding box annotations for evaluation and further academic usage. Detailed experiments show that, at the best configuration, our schema named “LogoSENSE” is able to achieve 93.50% precision and of 77.94% recall score along with obtaining F1 score of 85.02%. The experiments show that the proposed approach outperforms SIFT based detection and presents comparative results against a state-of-art deep learning based object detection method. As a result, LogoSENSE serves promising results in terms of detection accuracy and run-time efficiency, yielding a companion tool that can be used as a brand recognition mechanism for phishing web pages and emails.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Significant progress in e-commerce and the growing number of online services (e.g., e-mail, cloud data, online payment systems) have eased not only life but also led an increase in a special kind of cybercrime named phishing. By definition, phishing is a security threat that attempts to deceive innocent users into capturing their confidential information (e.g., username, password, credit card details and social security number) by combining social engineering and web site spoofing techniques (Chiew et al., 2015; Rao and Ali, 2015). The lifecycle of a typical phishing attack starts with receiving a fake e-mail, SMS or instant message from scammers which attempts to make users think and believe that it is coming from a legitimate source. It should be noted that these messages usually involve compelling statements and a link pointing to scammer's fake web page that mimics the target brand's legitimate web page.

Once the user inputs his/her credential, the life cycle of the attack eventually ends by sending the sensitive information to phishers in order to be used for various purposes such as online fraud or exploit of private data.

According to the phishing trend reports of Anti-Phishing Working Group (APWG, 2018), the total financial losses due to the attacks in 2014 and 2015 were 4.5 and 4.6 billion dollars respectively (Jain and Gupta, 2017). Moreover, as of 2013, approximately 450,000 phishing attacks were recorded. Apart from financial losses, in recent years, APWG has also pointed out a new emerging threat, so-called spear-phishing – a variant of phishing – which targets specific users or companies to obtain their private contents (e.g. business secrets, personal repositories) instead of financial profit. Further, APWG's Q3 (third quarter) report of 2017 highlights the increasing number of phishing attacks targeting the cryptocurrency sector.

Although it has been studied for almost two decades, phishing is still not a fully solved problem since there exists an arms race

E-mail address: selman@cs.hacettepe.edu.tr (A.S. Bozkir).

between phishers and researchers. According to (Jain and Gupta, 2017), the main reason for the vulnerabilities of phishing threats is the lack of user knowledge. The main trick of phishers is to create identical or visually very similar web pages which mimic their legitimate target counterparts. Moreover, as stated in (APWG, 2018), scammers have started to purchase SSL certificates in recent years in order to keep this illusion. Dhamija et al. (Dhamija et al., 2006) have conducted an experiment aiming to reveal to what extent participants distinguish legitimate web pages from fake ones. Their results show that 90% of the users were unable to discriminate between these two kinds of web pages. Another interesting finding is that 23% of the users do not even notice the address bar of web sites (Jain and Gupta, 2017). Due to these facts, in order to present a software-based solution, researchers have developed a variety of anti-phishing mechanisms that can be grouped under three main bases: (1) heuristic, (2) black or white lists, and (3) visual similarity (Jain and Gupta, 2017). We should note that one another classification of these mechanisms has been accepted in literature based on whether they are target-dependent or not Corona et al. (2017).

In general, *heuristic* based approaches aim to build a rule engine or prediction model based on various features extracted from structural assets of web pages such as URL (Rao and Ali, 2015; Moghimi and Varjani, 2016; Sathish and Thirunavukarasu, 2015), textual content (Rao and Ali, 2015; Ramanathan and Wechsler, 2012; Kauser et al., 2014; Moghimi and Varjani, 2016) or document object model (Xiang et al., 2011; Rosiello et al., 2007). Although they have shown a significant amount of success in detection, heuristic based approaches have some drawbacks as follows. First, due to the arms race, phishers are learning new features and forge them to modify their pages according to these techniques so that they are able to circumvent the latest heuristic based mechanisms. Second, according to Varshney, Misra and Atrey (Varshney et al., 2016), machine learning methods employed in heuristic based approaches have high-computational cost and require large datasets which make them impractical to deploy as a client-side application.

As another approach, *blacklisting* or *whitelisting* based solutions focus on keeping phishing or legitimate web page URLs respectively. These URLs are compared against the web page URL that user inputs to browser Varshney et al. (2016). Today, several contemporary browsers such as Google Chrome (Google Chrome Browser, 2020) and Mozilla Firefox (Firefox Browser, 2020) employ built-in blacklist databases which are being kept updated. For instance, Google Safe Browser API is being used by Chrome and Firefox browsers. The data source of these types of databases and APIs is based on either user feedbacks or reports of third party institutions such as APWG Varshney et al. (2016). Although they are relatively faster to execute, compared to other methods, they exhibit an indispensable limitation. Actually, a substantial amount of time is required to discover newly launched phishing webpages (Jain and Gupta, 2017). Nonetheless, the mean lifetime of a phishing web page is around 61 h (Moore and Clayton, 2007) and some of the web pages are shut down before they were identified. Further, as reported in (Sheng et al., 2009), 47% to 83% of the phishing URLs are added to blacklists within the first 12 h. Therefore, obtaining an up-to-date blacklist database is a very challenging task and users are becoming vulnerable to attacks if the URLs they visit do not exist in the blacklist database. In addition to the limitations as mentioned earlier, phishing web pages exhibit following frauds by which a robust anti-phishing mechanism also has to deal with:

- 3 Implementing the legitimate web page's HTML content in a way that will produce the same visual appearance with a completely different DOM organization.
- 4 Retrieving image-based contents via JavaScript.
- 5 Inserting SSL certificates in phishing pages.

In the light of these observations, with a growing number of visual similarity based phishing detection methods have been proposed in the literature. As stated by Jain and Gupta (Jain and Gupta, 2017), visual similarity based schemas use a visual signature created by utilizing: (1) DOM tree, (2) Cascading Style Sheet (CSS) similarity, (3) pixel-based similarity and (4) hybrid approaches. This is not surprising since the main point of phishing attacks is to exploit users' visual perception. Furthermore, as stated by Bozkir and Sezer (Bozkir and Akcapinar Sezer, 2016), recent years have witnessed an increasing tendency in the use of computer vision techniques for phishing detection. Apart from the other visual similarity based studies that rely on using structural information (e.g., DOM tree), pure vision based studies work on rendered web page images so that they become robust to spoofs listed above. It should also be noted that pure computer vision based approaches are considered as proactive solutions that are robust to zero-hour attacks (Bozkir and Akcapinar Sezer, 2016).







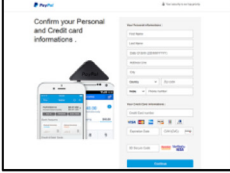
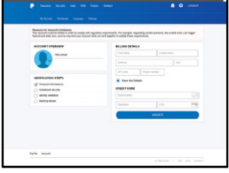
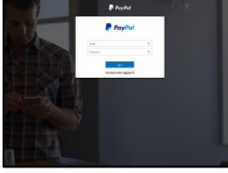
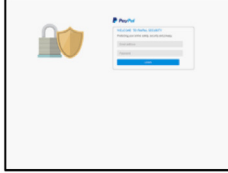





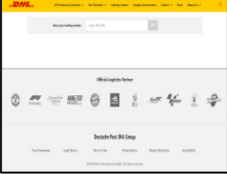


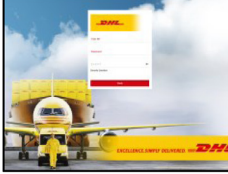



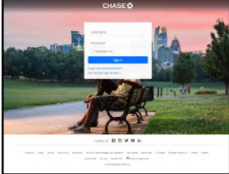


When the literature is reviewed, several studies (Corona et al., 2017; Bozkir and Akcapinar Sezer, 2016; Chen et al., 2009; The-Chung et al., 2014; Dalgic et al., 2018) which attempt to match partial or whole page screenshots with legitimate counterparts can be found. Although these studies show promising and successful detection rates, they may suffer from the problem of phishing page polymorphism in which phishers often use different layout and color schemes in order to evade visual similarity based countermeasures. This situation has been illustrated by the example snapshots given in Table 1 in which legitimate and phishing web pages of three different highly phished brands are presented. As can be seen, phishers may prepare both visually similar and unrelated phishing web pages by using phishing toolkits. These phishing variants may be entirely irrelevant for their legitimate counterparts yielding cases hard to detect via computer vision based recognition systems. The main reason for this argument is that phishing pages involve high visual variance along with a lower number of common elements compared to their legitimate counterparts. At this point, the use of modern methods such as deep convolutional neural networks (DCNN) could be considered as a helpful method since their high generalization capability. However, a brand new zero-hour phishing web page could remain too difficult to be correctly classified because of (1) the aforementioned high variance and (2) small training datasets.

Another critical aspect of phishing attacks is that the scammers usually insert target brand logos into their phishing web pages in order to gain the trust of users (Chiew et al., 2015). This observation constitutes the primary motivation of this study since the existence of the logo has a significant impact on site credibility in addition to representing the identity of the targeted legitimate web page. In their work, Dhamija et al. (Dhamija et al., 2006) and Alsharnouby et al. (Alsharnouby et al., 2015) approve the brand logo as a vital visual trust cue in company web sites. Further, Dhamija et al. (Dhamija et al., 2006) define the successful phisher as the one who presents convincing content causing the victim to fail to notice security indicators exhibited by anti-phishing tools. On top of that, the brand logos could remain unchanged or slightly altered components of phishing contents even in the case where scammers modify legitimate web page/mailling design entirely. Based on 78-day observation on phishtank.com, Geng, Lee and Zang (Geng et al., 2015) reported that 86.2% of the phishing web pages contain brand logos and stated that "brand entities are powerful weapons of phishers to trick users". In line with this observation, Jain and

- 1 Substituting of text blocks with other types of contents such as images, ActiveX or Java Applets which generate the same visual appearance (Jain and Gupta, 2017).
- 2 Phishing page polymorphism: Changing the page layout or textual contents of legitimate web pages (Lam et al., 2009).

Table 1

Web page screenshots for some brands and their phishing variants.

Legitimate page snapshots	Several phishing web page snapshots for different brands			
				
				
				
				
				

Gupta (Jain and Gupta, 2018) pointed out that phishing e-mails involve the company logo and mimic the legitimate mailing templates. To our knowledge, there exists a limited number of studies in the literature (Chiew et al., 2015; Wang et al., 2011; Xiang, 2013) that attempt to detect phishing web pages by capturing the target brand logos. These approaches mentioned above usually employ local descriptors such as SIFT (Lowe, 2004) or SURF (Bay et al., 2008) and try to determine the visual similarity by keypoint matching. To sum up, it is a critical trade-off for a phisher to include or avoid logo placement in phishing contents. Including brand logo brings the risk of detection whereas lack of it causes loss of a significant trust cue.

In this work, we propose and evaluate a pure vision based logo detection and classification schema (i.e., LogoSENSE) in order to identify the target brands involved in detected phishing web pages (See Fig. 1). In contrast to several studies which focus on phishing web page identification, we focus on recognition of the target brand(s) of the phishing webpage or e-mail. Instead of keypoint matching, we suggest using HOG descriptors (Felzenszwalb et al., 2010) in order to detect target brand logos in multi-scale object detection fashion. Throughout the study, we have employed max-margin object detection (MMOD) framework (King, 2015) addressing max-margin loss in SVM. As stated in (King, 2015), MMOD

is an efficient sliding window based object detection method that employs all sub-windows rather than sub-sampling. This schema, moreover, enables training with few positive samples while avoiding negative example mining. It should be noted that another advantage of MMOD is to reduce false predictions. We have also collected a publicly available annotated training and evaluation data set at <https://web.cs.hacettepe.edu.tr/~selman/logosense/>. The training dataset involves 3060 web page snapshots covering 15 highly phished brands. For the evaluation dataset, we collected 1979 unique snapshots, including unique 979 phishing and 1000 legitimate web page/e-mail screenshots which include logos involved in the training dataset. According to the results of the comparative study, the proposed schema surpasses local keypoint based methods in terms of efficiency and accuracy.

The main contributions of this paper are summarized as follows:

- 1 Regarding the methodology, we treat the problem in terms of pure vision based object detection and recognition in a data-driven manner.
- 2 We propose to employ MMOD – a fast, efficient, accurate and few positive samples requiring framework – for the first time in the phishing page logo detection and recognition domain.

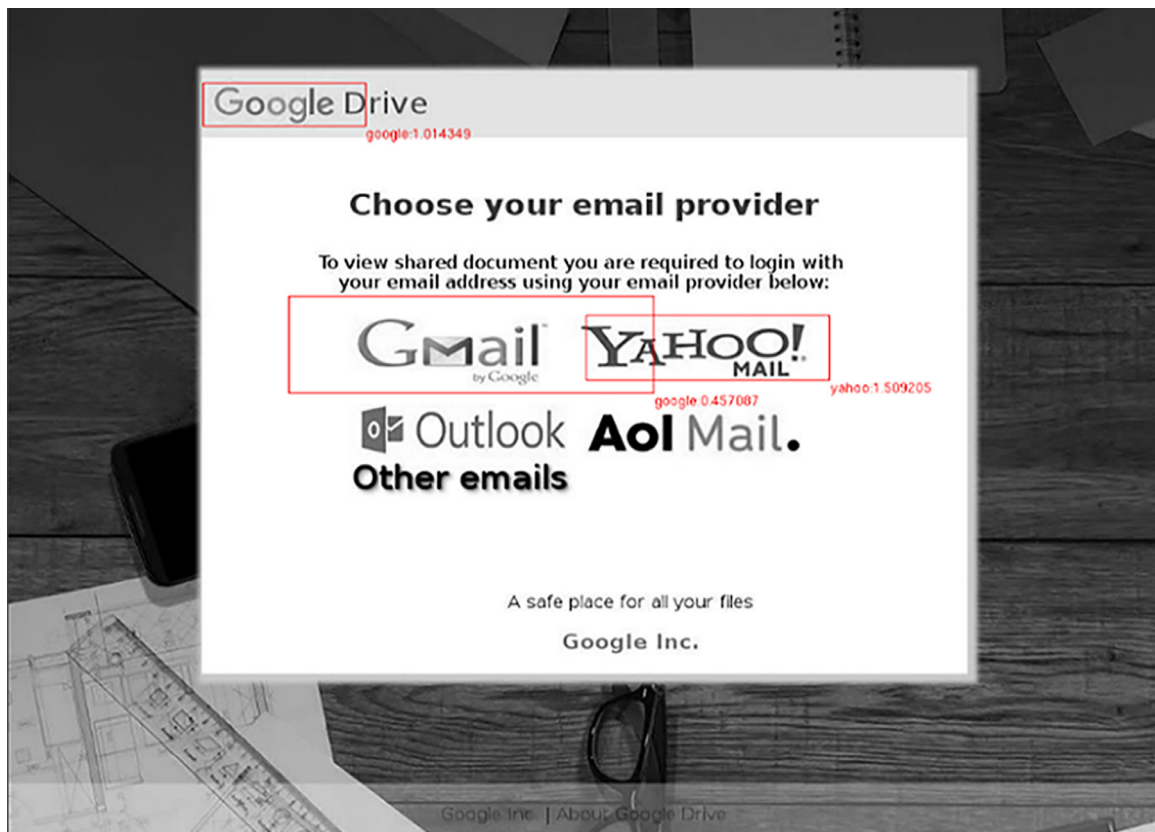


Fig 1. A sample detection and recognition result of the proposed scheme.

- 3 We released a bounding box annotated and a publicly available dataset which will hopefully be useful for further research on this field.
- 4 The proposed model is only using web page/e-mail snapshots as the source of information which is invariant to underlying HTML source code and the language of the web page/e-mail. Moreover, it is robust to change of page layout and color schemas used in suspicious content.
- 5 The proposed system can be used as a middleware for brand recognition purposes in both phishing web pages and emails without any change.

The rest of this paper is organized as follows. Section 2 overviews the visual similarity based studies in the literature. Section 3 demonstrates the methodologies used in this work. Section 4 presents details about the collected data set. Then we give details of the system in terms of architecture and processing. In Section 5, the design and workflow of the proposed approach are explained. Then, we present the results obtained from comprehensive experiments along with a comparative study in Section 6. In Section 7, we have discussed some drawbacks of the suggested scheme. Finally, this paper is concluded in Section 8.

2. Related work

In recent years, due to the difficulties as mentioned above, there has been a tendency to employ visual similarity based approaches in the detection of phishing pages. Visual similarity can be captured by the use of features extracted via either utilizing structural information or employing pure computer vision methods.

In their work, Fu and Wenyan (Fu and Wenyan, 2006) have employed color-based features which were extracted from low-

resolution web page screenshots, and they computed the dissimilarity between these signatures via Earth Mover's Distance.

In order to combat with polymorphic phishing web pages, Lam et al. (Lam et al., 2009) have used image processing techniques and utilized some heuristic rules to reveal the layout of the webpage blocks. To achieve this goal, they have searched "minimum vertical and horizontal inter-space between the blob and all of its neighbors on the right-hand side" (Lam et al., 2009). Next, they divided the image into nonoverlapping blocks for the next process of block-level matching with the help of some predefined rules. In their work, so-called "DeltaPhish", Corona et al. (Corona et al., 2017) have combined both structure and vision based information to detect phishing pages in target-independent fashion. Hence, they extracted HTML specific features (e.g., URL, style information) along with layout related visual features such as color and HOG.

As another work which relies on page layout matching, Chen, Huang and Chen (Chen et al., 2009) have proposed to use a modified version of Contrast Context Histogram (CCH) descriptor. Moreover, they have clustered the strong matching key points to identify matching regions. Bozkir and Akcapinar Sezer (Bozkir and Akcapinar Sezer, 2016) have developed a system that generates a HOG based visual signature on representing the square sized top section of a web page. The generated signatures are then compared via histogram intersection kernel and authors have reported that the similarity of 75% or higher indicates a strong phishing alarm.

In (Jain and Gupta, 2018), Jain and Gupta proposed a phishing web page detection scheme based on two modules: (1) a pre-filter with a login form detector and whitelisting and (2) a rule-based classifier employing features sourcing from keyword, hyperlink, reference domain, and Cascaded Style Sheets gathered from suspicious pages. Their work also uses Google Search Engine API to query detected significant textual contents such as keywords, page title and meta-tags. In addition, they computed the visual

layout similarity of two pages by only considering CSS selector-value pairs. Though they have achieved higher true positive rate (i.e. 99.72%) with the less false positive rate (i.e. 1.49%), one of their main assumptions which the phishing web pages always mimic the legitimate counterparts is doubtful and our current observations indicate that this assumption should be argued and discussed.

In contrast to other visual similarity based studies, several works aim to identify phishing attacks by detecting target brand logos in phishing web pages. By definition of Wang et al. (Wang et al., 2011), “the premise behind this fairly simple idea is that a critical element in virtually all fraudulent sites is the brand mark, or logo, of the institution being imitated”. Indeed, in order to deceive users, the key element for phishers is the brand logo of the targeted web page. According to our observations, accurate detection of logos could provide a more robust anti-phishing scheme since the polymorphism on logos is less dramatic than the other visual features of web pages such as layout, style and color.

In their work, to detect phishing web pages, Geng, Lee and Zhang (Geng et al., 2015) have proposed to employ favicon, brand logo and copyright information as the source of information. To locate candidate brand logos in suspicious web pages, they have used an HTML based web page segmentation approach and compared their Hu moment features with a curated legitimate logo set. Besides, in order to verify whether the detected web page is phishing, they have utilized some other features such as name server, resolution IP, redirection state and incoming link information. Based on the conducted experiments with machine learning methods by using individual and combined feature sets, authors showed that the features solely extracted via Hu moments over brand logos establish more discriminative power among the others during the classification stage. Further, they obtained 0.989 true positive rate (TPR) while reducing the false positive rate to 0.042 (FPR). Though they achieve high TPR along with relatively low FPR, there exist some inherent problems about the selection of candidate logo images. First of all, phishers have a general tendency to embed logos in a much larger image file that sometimes might even cover the whole background. This situation causes high noisy content to be considered during signature generation. Second, authors assume that all phishing web pages have their brand logos in their upper left part. Nevertheless, this case is not always correct. There exist several instances in which brand logos may be located in the upper right part due to the target language the scammer focuses on.

Dunlop et al. (Dunlop et al., 2010) have suggested a browser-based plug-in so-called “GoldPhish” which extracts the logo from the suspicious web page and apply optical character recognition (OCR) to reveal its text content. The extracted text is submitted to a search engine “Google” in order to check the existence of the suspicious page’s domain in top results. As highlighted in (Jain and Gupta, 2017), though this scheme is capable of protecting users from “zero-hour” attacks, the robustness of the system is highly dependent on the underlying OCR mechanism. As another browser plug-in based attempt, Wang et al.’s work (Wang et al., 2011) named as “Verilogo” employs SIFT features to detect phishing pages. In order to achieve this, Verilogo keeps a database of SIFT keypoint descriptors belonging to top phished brands and seeks intense pairwise matches between individual logos and stripes extracted from suspicious web page screenshot. Furthermore, the authors have reported that the detection of the first logo takes approximately 4 s. According to the results of our comparative study that we present in Section 6, Verilogo achieves very high precision scores while obtaining relatively low recall values.

The study of Chiew et al. (Chiew et al., 2015), on the other hand, proposes a two-stage approach that downloads all images of the suspicious web page and recognizes them by use of 10 different pixel-based features such as mean, energy and entropy. At

the second stage, they verify the domain name of the positively classified web page by checking it on “Google Image Search”. The comparative study which was presented in (Chiew et al., 2015) reported that their approach is superior to the work of Dunlop et al. (Dunlop et al., 2010) in terms of false positive and true negative rate.

Apart from conventional approaches relying on hand-crafted features, the proliferation of deep learning (DL) has revolutionized the way of machine learning in computer vision and other fields. Beyond its superior generalization capability, DL has also enabled machines to learn in an end-to-end manner. Similarly, this revolution has also impacted and improved studies related to logo detection and recognition. As such, Bianco et al. (Bianco et al., 2017) have suggested a convolutional neural network (CNN) based logo recognition scheme which is working on images in the wild. In order to achieve this goal, they first built a custom and basic CNN architecture which aims to classify input images. For selecting candidate image patches, they have used the *selective search* mechanism in order to localize logo like regions. In their study, they have utilized the “Flickr-Logos32” and “Logos-plus32” datasets and evaluated the performance gain of synthetic versus real data augmentation along with image processing techniques (Bianco et al., 2017). Further, by applying techniques such as class balancing, contrast normalization and weight sampling, they have reached up to scores of 0.989, 0.906 and 0.946 in terms of precision, recall and F1 respectively. In another study, Indola et al. (Indola et al., 2015) have modified the well-known “GoogLeNet” deep neural network architecture to improve the logo recognition performance at “Flick-Logos32” dataset. Authors of (Indola et al., 2015) have reported that they have achieved an accuracy of 89.6% during the recognition. In another study carried out by Oliveira et al. (Oliveira et al., 2016), brand logos in natural images have been localized and classified by utilizing transfer learning techniques on an object detection scheme named as “Fast Regional Convolutional Neural Networks” (F-RCNN) suggested by Girshick (Girshick, 2015). Oliveira et al. (Oliveira et al., 2016) have tested and assessed two different selective search modes: (a) fast and (b) quality. During the study, they have employed VGG architecture and applied different threshold values in order to evaluate the overall detection performance. Based on the experiments carried on “Flick-Logos32” dataset, they have reached up to state-of-art scores of 0.955, 0.908 and 0.931 in terms of precision, recall and F1 respectively.

More recently, Su et al. (Su et al., 2018) proposed a new incremental learning scheme called “Scalable Logo Self-co-Learning (SL²)” which enables the self-discovery of training images from noisy web environment in an automated manner. Instead of working with a limited number of manually bounding box annotated images, authors of (Su et al., 2018) introduced a scheme to reveal and retrieve potential logo images from web data by progressively developing the model discrimination power via exploiting iterative joint self-learning and synthetic context augmentation. As a result, they have collected an extensive “WebLogo-2M” dataset including more than 2.1 million images of 194 brands. Moreover, their approach outperforms all other state-of-art object detection methods in terms of mAP (46.9). As can be seen, the studies employing DL methods generally focus on detecting logos in natural images. According to our best knowledge, none of them has investigated the use of DL in the anti-phishing field.

3. Methodology

This study aims to detect logos of target phished brands in suspicious web pages by assuming them as objects which can be semi rigidly represented with HOG descriptors. To do so, we have utilized the novel max-margin object detection approach developed by King (King, 2015). The remaining of this section briefly intro-

duces key concepts about HOG and MMOD. For further exploration, the interested reader is referred to those referenced sources in the literature.

3.1. Histogram of oriented gradients

Invented by Dalal and Triggs (Dalal and Triggs, 2005), Histogram of Oriented Gradients is a powerful computer vision method that is employed to represent local object appearance in a semi-rigid way by making use of distribution of intensity gradients and edge directions. To date, HOG descriptors were used in various fields such as vehicle brand recognition (Llorca et al., 2013) and moving object detection (Liang and Yuang, 2015). For the following reasons, HOG descriptors are well suited to the task of logo detection: (i) being capable of capturing regional visual cues of logos; (ii) providing a semi-rigid representation of objects by having a certain degree of rotational and translational invariance; (iii) generating compact vectors which reduces the complexity of further computations.

As stated in (Bozkir and Akcapinar Sezer, 2016; Dalal and Triggs, 2005), extraction of HOG descriptors employs 3 stages: (i) gradient computation, (ii) orientation binning and (iii) block normalization. First, the image or region of interest is divided into a grid of equal-sized cells. Next, for each pixel, orientation bins are computed and assigned according to the angle ranges by converting the gradient vectors to an angle. As the third stage, in order to avoid illuminance variations and to obtain more robust representation, the normalization process is carried out on grouped cells (blocks). Eventually, normalized histograms are concatenated and a vector for the descriptor is generated. For further reading on HOG descriptors, Dalal and Trigg's paper (Dalal and Triggs, 2005) can be studied.

3.2. Max margin loss object detection

Developed by King (King, 2015), max-margin object detection (MMOD) is an object detection schema, which incorporates maximum margin loss in a structured learning framework which is then used to learn rigid object appearances via HOG features. In general, several object detection schemas are based on application of binary classifiers on sub-windows of an image, followed by removing overlapped detections via non-maximum suppression (King, 2015). However, this technique is not very efficient since it presents a high computational cost to cover and compute all sub-windows. Thus, this situation leads to a sub-optimal solution due to the utilization of only a subset of whole training data. To overcome this limitation and related problems, MMOD approach suggests an optimizer working on all windows along with high-performance object detection in terms of missed and false detections (King, 2015). Furthermore, MMOD provides a multi-scale detection mechanism since the sliding windows are formed and processed by a coarse-to-fine pyramidal fashion. Besides, with the use of max-margin loss, MMOD constitutes a schema that is trainable with few positive samples.

Given that, r denotes a rectangular area of an image while \mathfrak{R} represents whole valid labeled rectangles such that they do not overlap each other and will be scanned by the algorithm. At this point, regarding the rectangles of r_1 and r_2 , we define the property of "non-overlapping" as given in (1):

$$\frac{\text{Area}(r_1 \cap r_2)}{\text{Area}(r_1 \cup r_2)} < 0.5 \quad (1)$$

That is, if the intersection area of two rectangles is greater than half of their union, this implies that these two rectangles (i.e. r_1 and r_2) are overlapping. At this stage, if \mathcal{R} denotes all valid labeled rectangles then the process of object detection can be defined as

presented in (2), where f indicates the window scoring function and i represents the target image:

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{r \in \mathcal{Y}} f(i, r) \quad (2)$$

In other words, (2) describes a set of sliding window positions having the highest scores yet do not overlap each other (King, 2015). This procedure, in general, is accomplished by utilizing a sorting method incorporating non-maximum suppression. However, as pointed out by King (King, 2015), an ideal scheme would use a window scoring function that jointly minimized the number of false alarms and missed detections produced.

If we rewrite the (2) in a detailed way we arrive at (3) where \mathbf{w} is a parameter vector and ϕ represents the feature extractor from the sliding window located at r .

$$f(i, r) = \langle \mathbf{w}, \phi(i, r) \rangle \quad (3)$$

As pointed out in (King, 2015), denoting the sum of window scores for a set of rectangles, y , as $F(i, r)$ yields to (4):

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} F(i, r) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{r \in y} \langle \mathbf{w}, \phi(i, r) \rangle \quad (4)$$

At this step, it is evident that we need an optimized parameter vector \mathbf{w} that generates the lowest number of detection faults. Thus, our aim eventually becomes scoring the correct label of i_m so that it becomes the largest among the other incorrect labelings when it is given an image and label pair $(i_m, y_m) \in I \times \mathcal{Y}$ (King, 2015). Note that, more the condition (5) occurs, less the mistakes we have.

$$F(i_m, y_m) > \underset{y \neq y_m}{\max} F(i_m, y_m) \quad (5)$$

According to the initial problem definition, MMOD is provided with a set of images $\{i_1, i_2, \dots, i_n\} \subset I$ along with assigned labels $\{y_1, y_2, \dots, y_n\} \subset \mathcal{Y}$ and its main objective is to seek a "good" \mathbf{w} . At this point, given in (6), the max-margin approach comes into the picture in order to predict the correct label of rectangle r_i with a large margin.

$$\underset{\mathbf{w}}{\min} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{s.t. } F(i_m, y_m) \geq \underset{y \in \mathcal{Y}}{\max} [F(i_m, y) + \Delta(y, y_m)], \forall i \quad (6)$$

Where $\Delta(y, y_m)$ indicates the loss for faulty predictions where the correct label of y is y_m . In his work (King, 2015), King has designed the loss $\Delta(y, y_m)$ as given at (7):

$$\Delta(y, y_m) = L_{\text{miss}} \cdot (\# \text{ of missed detection}) + L_{\text{fa}} \cdot (\# \text{ of false alarms}) \quad (7)$$

Here the L_{miss} and L_{fa} are being used to parameterize the importance of obtaining high recall and precision. Since the regime applied in (6) is a hard-margin formulation, King (King, 2015) has transformed it into a soft-margin form as given in (8)

$$\underset{\omega, \varepsilon}{\min} = \frac{1}{2} \|\omega\|^2 + \frac{c}{n} \sum_{m=1}^n \varepsilon_m \quad (8)$$

$$\text{s.t. } F(x_m, y_m) \geq \underset{y \in \mathcal{Y}}{\max} [F(x_m, y) + \Delta(y, y_m)] - \varepsilon_m, \quad \forall m \\ \varepsilon_m \geq 0, \quad \forall m$$

By this conversion, it is aimed that the convex optimization problem turns out to be more robust against noisy and not perfectly separable data. As explained in (King, 2015), the rest of the MMOD converts the *hard margin* formulation denoted in (6) to a *soft margin* setting and obtains the \mathbf{w} by using the cutting plane method (Mao et al., 2017) which computes an increasingly more accurate lower bounding approximation constructed from tangent planes. For further reading, the work of King (King, 2015) could be read since the paper also touches the strategy involved in solving the MMOD optimization.

Table 2

Number of screenshots included for each brand in training and validation set.

	Alibaba	AOL	Apple	BOA	Chase	DHL	Dropbox	Facebook
# Train/Test Images for each fold	(136/68)	(136/68)	(136/68)	(136/68)	(136/68)	(136/68)	(136/68)	(136/68)
# Validation Images	32	25	63	54	66	64	50	124
	Google	Microsoft	Office	Orange	Paypal	Wellsfargo	Yahoo	Total
# Train/Test Images for each fold	(136/68)	(136/68)	(136/68)	(136/68)	(136/68)	(136/68)	(136/68)	3060
# Validation Images	48	47	27	72	78	72	42	864

The aforementioned design of the max-margin objective function has been preferred in this study because of the reasons listed below:

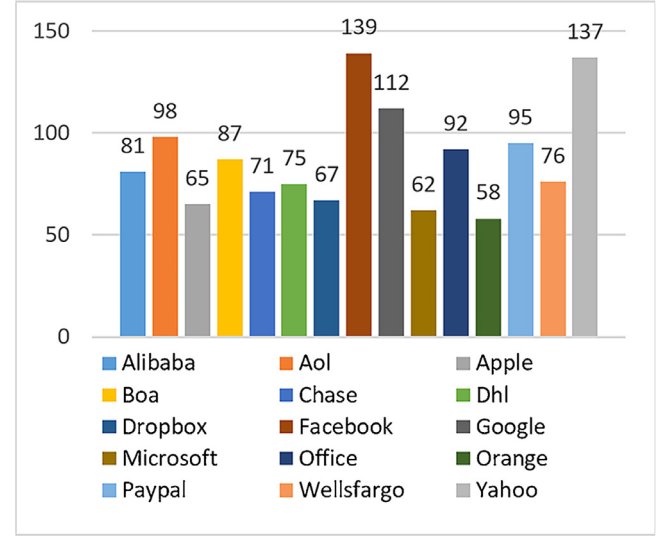
- Phishing web pages or emails usually involves logos that generally mimic the original brand logos and the alterations are carried out on a small scale. For this reason, the robustness of MMOD against alterations in semi-rigid objects enables us to use it efficiently on the problem we address.
- Max-margin object detection framework requires only “positive” truth boxes to be trained. There is no need to define negative samples. All image patches except the annotated logos are being accepted as negative samples. Therefore, this makes MMOD a suitable solution to detect and recognize logos on screenshots.
- The structure and design of the loss function and the strategy used to solve the required quadratic SVM makes it possible to run MMOD with less training “positive” samples. This is a benefit for the phishing web logo detection since the in-class variations among the brand logos are reasonably low in the problem domain.

4. Dataset

As mentioned before, this study aims to find corresponding logos in phishing web pages by using object detection methods. At this point, the need for bounding box annotations of screenshots has emerged. By definition, bounding boxes describe the coordinates (i.e. x_{top} , y_{top} , $width$, $height$) of ground truth locations of the logos to be detected. To our best knowledge, in the literature, there exists no such kind of phishing data set that was annotated with ground truth boxes of phished brand logos. For this purpose, we have first picked 15 well known and highly phished brands as follows: Alibaba, AOL, Apple, Bank of America, Chase Bank, DHL, Dropbox, Facebook, Google, Microsoft as the brand itself, Office as the office products of Microsoft, Orange, Paypal, Wellsfargo and Yahoo. Note that, the curation of these brands is based on a long-term daily-basis observation of phishing page listings served on www.openphish.com. For this selection, we observed and recorded the number of phishing alerts between 1.9.2018 – 3.3.2019. Further, we have collected two distinct datasets including these brands for different purposes. While the first corpus (i.e., *training set*) has been used to train each individual brand logo detector, the second and larger one (i.e., *wild-set*) was collected in order to evaluate the system performance in the wild. It is noteworthy that snapshots involved in the *wild-set* were curated in a way that they have much more significant variance in terms of visual design than the training set.

Throughout the data collection, we have both utilized the web sites of “Phishtank” (Phishtank, 2020a) and “Phishbank” (Phishbank, 2020b). Upon gathering the snapshots, we have used a GUI based “imglab” tool shipped with “dLib-ml” framework (King, 2009) to annotate ground truth bounding boxes.

For the training dataset, we collected 1530 (102 unique snapshots \times 15 brands) unique phishing + 102 legitimate web page

**Fig 2.** Distribution of brand logos to be detected in wild-set corpus.

screenshots curated from different sectors and languages. Meanwhile, the term *unique* here refers to denote visually distinct web pages. As a result, we have used 204 snapshots in order to create a detector for each brand. While half of these samples (102) were selected from phishing web pages of the corresponding brand, we randomly collected the remaining images from legitimate web pages sourcing from different contexts (e.g., E-trade, porn, education). Note that, we copied these 102 legitimate web page snapshots to all brand logo set folders located in separate folders. As a result, we obtained 3060 image samples for the training set in total. Moreover, in order to validate the individual brand logo detection performance of the models created by using the training set, we have collected a relatively small but unique validation-set. The number of snapshots included in *training* and *validation* sets for respective brands has been listed in Table 2.

We have manually annotated the snapshots with ground-truth boxes by drawing surrounding boxes of the logos. As stated before, we have used imglab tool for this purpose. Imglab is a C++ based application equipped with a graphical user interface and serves many features for object detection tasks. Since our task is to detect related logos in their correct positions (i.e., overlapping ratio > 0.5), we carefully annotated snapshots located under each brand logo folder.

In order to evaluate the robustness of the detector performance for each brand detector in real-world scenarios, we have constructed the *wild-set* dataset via sampling different phishing samples for each brand. For this purpose, we have collected 979 phishing e-mail and web page screenshots having a varying number of phished brand logos. Note that, while we were curating web page snapshot during the collection of training and validation sets, varied sized (i.e. width and height) e-mail snapshots were also ap-

Table 3
Several brand logo examples in training/validation set. It can be easily seen that, some brands show high variances in their logo design because of several reasons such as: (1) brand logo evolution, (2) distortions made by phishers intentionally. .

Brand Name	Extracted logo patches on training snapshots					
Alibaba						
AOL						
Apple						
BOA						
Chase						
DHL						
Dropbox						
Facebook						
Google						
Microsoft						
Office						
Orange						
Paypal						
Wellsfargo						
Yahoo						

pended to our wild-set. We have also sampled 1000 unique legitimate web pages in order to assess the false positive performance of the proposed scheme. Consequently, for the wild-set, we gathered 1979 (1000 legitimate + 979 phishing) screenshots in total. The distribution of the snapshots of respective brands involved in *wild-set* has been plotted in Fig 2. The aforementioned 1000 legitimate web page snapshots were curated from Alexa top 100 web pages in addition to randomly selected ones covering different languages and sectors.

For academic purposes, we serve the proposed dataset for free at <https://web.cs.hacettepe.edu.tr/~selman/logosense/>. Several example logos for each brand have been illustrated and listed in Table 3. As can be easily seen, there exist two types of variance in logos: (1) evolution of the brand logo in years and (2) visual distortions made by phishers intentionally. These factors affect the visual structure (a.k.a. transformations in terms of scale and orientation) of the located logos and constitute challenges for logo detection and recognition processes.

5. The proposed approach

5.1. Design of the application

Our system, named LogoSENSE, has been implemented in C++ by employing OpenCV (OpenCV, 2020) and dlib machine learning (King, 2009) libraries. Due to the source code transportability of used libraries and our implementation, LogoSENSE can be compiled and used on any platform including Linux, Windows and Android. During the compilation, we have enabled the compiler to use AVX instruction sets in order to achieve faster execution time. As our preliminary experiments show that, the use of AVX instructions on Intel CPUs shipped later than 2011 improves detection speed by factor 10× (i.e. 4 s → 0.4 s). Our implementation expects either a single image file or multiple files specified by a folder name as a startup argument. In addition, users are enabled to specify minimum detection probability threshold (i.e. [0–1]) and image zoom factors (i.e. 1×, 1.5×, 2×).

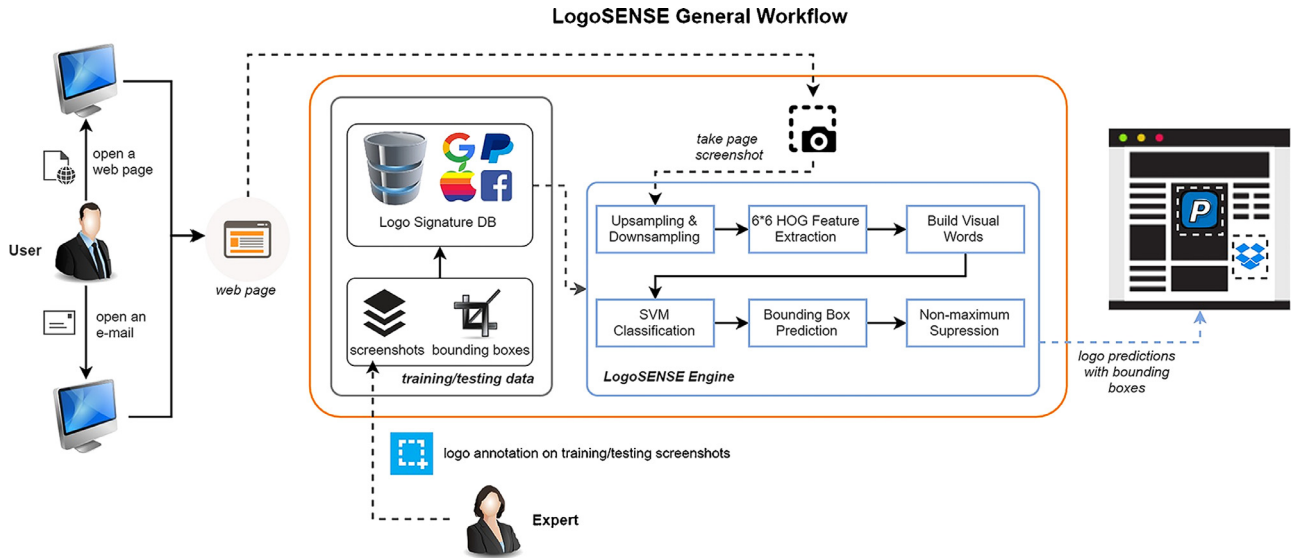


Fig 3. General workflow of the proposed system.

5.2. General workflow

In Fig. 3 the workflow of the proposed scheme has been illustrated. As shown in Fig. 3, LogoSENSE requires a training procedure for the brand logos of interest. Therefore, it is crucial to have a human curator and annotator at the training stage. Once the training of the model of a specific logo is finished, the weights of the SVM model is being stored to be further used. For this reason, LogoSENSE has two modes: (1) *training* and (2) *detection*. During the *training* stage, our application reads the XML file containing paths of images along with their corresponding ground truth boxes. Given a directory path, in *detection* mode, it outputs a plain text file that lists the image filenames and the labels of the detected logos along with their coordinates and probability values.

5.3. Feature extraction and prediction

According to our observations, it is a common fact that phishers deliberately place very large or small logos to evade logo based anti-phishing mechanisms. Therefore we have provided an option to upscale input images (i.e., 1.5x, 2x) so that small logos can be detected. During the feature extraction step, as stated in (King, 2015), each sliding window is divided into 6×6 grid which will then be used to extract a 2048 bin histogram of visual words. MMOD's original HOG schema computes 36-dimensional HOG vectors for each 10×10 grayscale pixel blocks which involve 5×5 cells. As mentioned before, a robust detector must detect all the objects of interest regardless of its scale. This situation is also valid for phishing web pages. In order to cope with this, King's MMOD (King, 2015) employs spatial pyramid bag-of-words schema (Lazebnik et al., 2006) in which the number of pyramids can be set easily. One another problem is that each sliding window produces long concatenated feature vectors leading to computational cost for quantization. Therefore, MMOD employs the method of random projection based locality sensitive hashing which eventually generates an 11-bit number to resemble the visual word's bin (King, 2015). Next, the method of non-maximum suppression – a well-known post-processing technique for eliminating weak candidate detections – has been used to localize the most robust sliding window.

Finally, it is noteworthy that King's MMOD (King, 2015) allows training in a binary fashion, yet it can be used as a multi-class detector. Since the dlib toolkit (King, 2009) enables us to classify

a window by multiple trained classifiers, it is possible to detect more than one type of logo per image.

6. Experiments and results

In order to evaluate the efficiency and effectiveness of our proposed schema, we have followed a three folded experimental procedure briefly explained below. First, we have trained each brand logo detector by using respective training set and evaluated in 3 folded cross-validation setting. Second, by utilizing whole training set, we have trained logo detector for each brand and tested against the sample screenshots included in respective validation set. It should be kept in mind that, during these steps, we have always trained and evaluated brand-specific detectors. Finally, in order to assess the overall performance of the proposed scheme in multi-class fashion, we employed the wild-set corpus in which single and multiple logo involved snapshots exist. Moreover, the phishing cases curated for the generation of wild-set corpus involve both phishing web pages and email snapshots. Out of 979 phishing cases, 18 samples belong to fake email screenshots.

As our methodology is based on binary detection for a brand in different sized image patches, we followed up a very simple strategy to increase recognition speed. Yet, each image patch is being cropped once and tested by all available detectors.

Following to evaluation of LogoSENSE scheme, we made a comparative study by using the approach “Verilogo” suggested by Wang et al. (2011). According to our observations, there exist not very much study in the literature that we could make a fair benchmark by using the same dataset we proposed. The main reasons for this argument are as follows: (1) unlike the others which focus on image identification (Bozkir and Akcapinar Sezer, 2016; Fu and Wenjin, 2006; Dalgic et al., 2018), our study tackles the problem as an object detection task, (2) though there are some datasets (e.g. Phish-IRIS dataset (Dalgic et al., 2018)) in the literature, these dataset(s) cannot be directly used for comparison purposes since they involve no logo containing screenshot instance. Moreover, the logo recognition system in (Geng et al., 2015) is also not comparable since their scheme analyzes images extracted from HTML content by directly wrapping the content. However, our study relies on full screenshots. Because of these reasons, we have limited our comparative study only with “Verilogo” Wang et al. (2011).

During the conduction of experiments, an Ubuntu 18.04 installed computer equipped with Intel Core i7 8750 K processor and 24 GB of system memory has been employed.

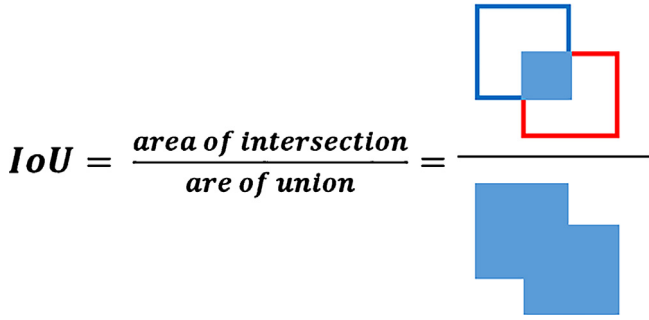


Fig 4. Graphical illustration of intersection over union (IoU) metric formulation.

6.1. Evaluation metrics

Throughout the study, we have used some evaluation metrics in order to assess the performance of the proposed scheme. In this section, we present details and meanings of these metrics which are used both in object detection stage and evaluation of the suggested scheme.

In a classification system, in order to express the performance, three metrics have been widely used (a) precision, (b) recall and (c) F1-measure (Imran et al., 2017). The formulas of precision, recall and F1-measure are given in (9, 10) and (11), respectively.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positive} + \text{False Positives}} \quad (9)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positive} + \text{False Negatives}} \quad (10)$$

$$\text{F1-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

At this point, the computation of precision and recall should be explained since the terms of *true positive*, *false positive* and *false negative* have some semantic differences compared to well-known binary classification schemes. To be informative, first, we need to define the metric of IoU (i.e. Intersection over Union) which has been briefly given in (1). An illustration of IoU is presented in Fig. 4. To this end, IoU metric denotes whether the detected logo's bounding box hits the ground truth box within a specific overlapping threshold t .

In line with these definitions, the aforementioned terms can be expressed as follows:

- True Positive (TP): A correct and in-place detection where $\text{IoU} \geq \text{threshold value}$.
- True Negative (TN): A wrong detection where $\text{IoU} < \text{threshold value}$.
- False Negative (FN): A ground truth box could not be detected

Following these definitions, the precision score indicates how many of the detections are correct since $\text{precision} = \text{TP/all detections}$. In contrast, recall score shows the ability of a classification model to capture all the relevant ground truth boxes since $\text{recall} = \text{TP/all ground truths}$. The behavior of a detection system could be managed in favor of either accuracy or relevancy via these two metrics and a threshold value. If one seeks a balance between these two metrics, F1-measure comes into prominence. Meanwhile, as can be seen, F1-measure is a geometric mean of precision and recall.

In addition to the precision score, average precision (AP) indicates the average precision value when the recall score is investigated from 0 to 1.

6.2. Experiments with the training dataset

In this first phase experiment, we have aimed to evaluate the target logo detection performance of each brand detector by applying 3-fold cross-validation on the *training* dataset. Further, the evaluation procedure took into account whether the detected logos "overlap" with ground truth boxes. This experiment aims to reveal whether each MMOD based brand detector is capable of detecting annotated ground truth boxes when a limited amount of training data is supplied. Note that, this experiment covers the performance of the models created for each brand detector by using only 136 training samples and testing on 68 remaining cases in 3 fold cross-validation regime. The detailed results of this experiment are provided in Table 4. The rows for each metric have been computed by taking the average of the scores collected from 3 folds.

According to the results, overall precision has been found as 0.9702 while overall recall has been computed as 0.8731. Moreover, the overall AP was eventually calculated as 0.8659. The overall scores show that our logo detectors work well in a general manner and precision scores are always better than recall values. In terms of the precision score, "Alibaba", "AOL", "Chasebank", "BOA" and "Wellsfargo" are ranked as the 5 topmost brands. Similarly, "Paypal", "DHL", "BOA", "Facebook" and "Yahoo" constitute the top 5 successful brands in terms of the recall score. Further, the best prediction success has been observed on "Paypal" and "DHL" brands when AP and recall scores are considered. This finding is not a surprise since our visual inspection over the logo instances of these brands reveals that the intra-class variation belonging to these successfully detected brands is relatively lower than the others.

On the other hand, "Orange" detector has presented relatively worst recall and AP scores (0.7818 and 0.7643 respectively) compared to others. We believe that the reason behind this finding is that the logo structure of this brand includes much less discriminative visual features. This specific finding was also discovered during the conducted experiments in Wang et al. (2011). Although our approach differs from theirs' in terms of feature extraction methodology (i.e., they have used SIFT features) edge and corner structures of the image patches still play a key role in order to obtain HOG based discriminative feature descriptors. This is obvious due to the fact that it is mostly affected by the directions of the image gradients. Thus, we can conclude that less the logo patches involve details, worse the recall scores we can achieve.

If the logo instances given in Table 3 are investigated, it can be seen that visual differences among the logos of a brand may differ from one to another. In other words, the level of visual consistency between the brands that we encounter show differences. Some brands have different versions of their logos involving either (1) only symbol, (2) only text, (3) combination of symbol and text and (4) different line options. This fact causes a challenge for the generalization capability of LogoSENSE since it heavily relies on templating a gradient map for characterizing the brand logo silhouette from training images. Although HOG features show rotational invariance to some extent, significant visual changes still constitute relatively small drops in recall scores.

6.3. Experiments with validation dataset

In this experiment, we have trained each brand detector by using whole training set and evaluated their individual precision, recall and AP scores by using *validation* dataset. As stated before, validation dataset consists of different phishing web page snapshots that do not exist in the training dataset. The results of this assessment are presented in Table 5. We also provided a chart in Fig. 5 regarding these scores.

According to the listed results, the best detector has been found as the one detecting the "Chase" brand. Meanwhile, the detectors of "Facebook" and "Google" have shown worse results in terms

Table 4

Average prediction scores of each brand detector when 3 fold cross validation has been applied over the *training* dataset. The best detectors have been marked with bold characters. .

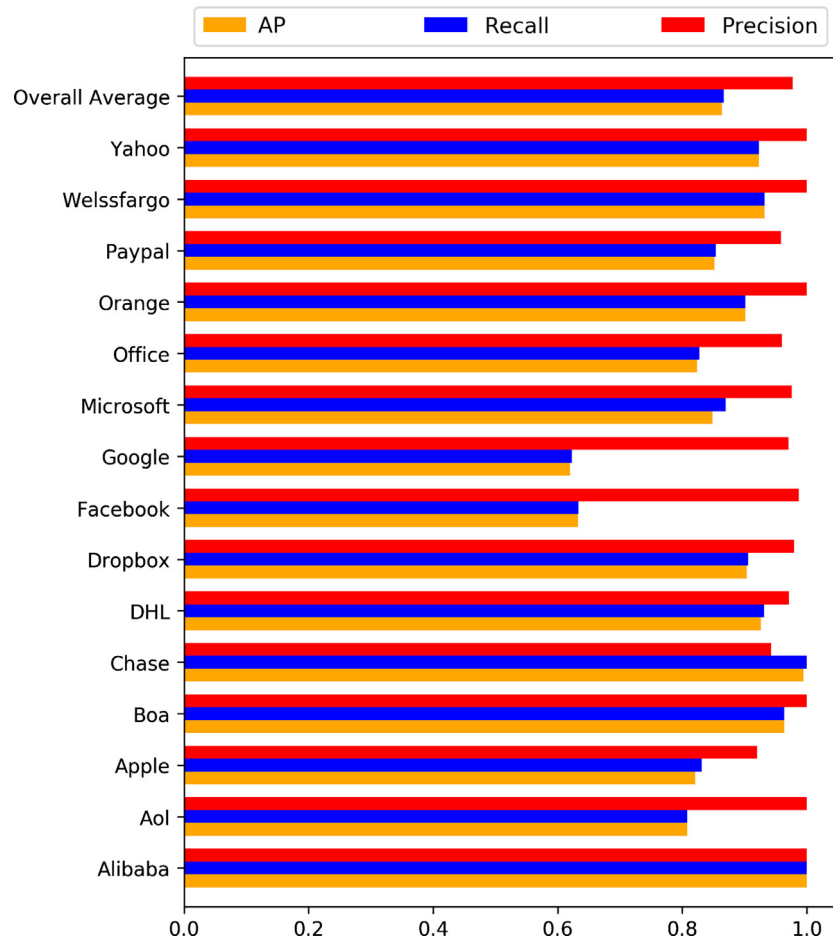
	<i>Alibaba</i>	<i>AOL</i>	<i>Apple</i>	<i>BOA</i>	<i>Chase</i>	<i>DHL</i>	<i>Dropbox</i>	<i>Facebook</i>
Precision	1	1	0,9493	1	1	0,9716	0,9340	0,9797
Recall	0,8823	0,8136	0,8752	0,9333	0,9019	0,9411	0,8321	0,9226
AP	0,8755	0,8136	0,8653	0,9304	0,9000	0,9411	0,8297	0,9117
	Google	Microsoft	Office	Orange	Paypal	Wellsfargo	Yahoo	Overall Score
Precision	0,9456	0,9616	0,9791	0,9438	0,9507	0,9907	0,9480	0,9702
Recall	0,8256	0,8850	0,8202	0,7818	0,9568	0,8127	0,9127	0,8731
AP	0,8121	0,8825	0,8193	0,7643	0,9487	0,8127	0,8819	0,8659

Table 5

Prediction scores of each brand detector on *validation* dataset. The best detector has been marked with bold characters while the worst ones are underlined.

	<i>Alibaba</i>	<i>AOL</i>	<i>Apple</i>	<i>BOA</i>	<i>Chase</i>	<i>DHL</i>	<i>Dropbox</i>	<i>Facebook</i>
Precision	1	1	0,9200	1	0,9428	0,9714	0,9795	0,9870
Recall	1	0,8076	0,8313	0,9636	1	0,9325	0,9056	<u>0,6633</u>
AP	1	0,8076	0,8209	0,9636	0,9947	0,9261	0,9037	<u>0,6326</u>
	Google	Microsoft	Office	Orange	Paypal	Wellsfargo	Yahoo	Overall Score
Precision	0,9705	0,9756	0,9600	1	0,9589	1	1	0,9777
Recall	<u>0,6226</u>	0,8695	0,8275	0,9014	0,8536	0,9324	0,9230	0,8689
AP	<u>0,6198</u>	0,8488	0,8234	0,9014	0,8514	0,9324	0,9230	0,8632

Individual and overall performance of detectors on validation data

**Fig. 5.** Individual and overall detection performance of detectors on validation dataset.

of recall and AP compared to the previous evaluation (underlined in Table 5). These two distinct cases show how the intra-class variation affects the recall performance of the proposed scheme. Nonetheless, one another outcome of these experiments is that precision scores still remain high in case of a decrease in recall values. Combining with the results of the previous experiment, we can conclude that the LogoSENSE scheme behaves in favor of precision rather than recall by using default settings.

At this point, one may ask a natural question of “Can we change this behavior in favor of recall?”. The answer to this question actually lies behind the loss function used in SVM classification module. Recalling the (7), there exist two coefficients namely L_{miss} (missed detection) and L_{fa} (false alarm) for computing the overall loss throughout the learning process. In our default settings, we use the same coefficient value of 1. Incorporating a smarter and adaptive loss function design is capable of changing the behavior of the algorithm in favor of either precision or recall.

In order to test this argument, we have incorporated focal loss (Lin et al., 2017) into the loss function. Invented by Lin et al. (Lin et al., 2017), the focal loss is a loss function designed for object detection tasks and equipped with the estimated detection probability into loss function design rather than setting a fixed cost for false detections. The formal definition of focal loss is given in (12) where p_t denotes the probability of detection and a modulating factor of $-(1 - p_t)^y$ has been added as a tunable *focusing* parameter where $y \geq 0$ (Lin et al., 2017)

$$FL(p_t) = -(1 - p_t)^y \log(p_t) \quad (12)$$

It is noteworthy that it decreases the accumulated effect of easily classified negatives on overall loss with a varying parameter of y . According to our preliminary experiments, by use of focal loss, recall scores can increase up to 8.3% along with the decrease of precision scores up to 6.4%. Thus, we eventually concluded that to find out an optimal point, conducting further rigorous and comprehensive experiments are needed, and we have left it as future work of this study.

6.4. Experiments. with wild-set dataset

In this experiment, we evaluated our proposal against a larger and “mixed” wild-set corpus involving 984 phishing and 1000 legitimate web page/e-mail snapshots. In other words, we tried to observe whether the proposed approach is robust in real-world scenarios. Further, we assess the performance of built detectors whether they generate low false alarms (i.e., false positives) while achieving precision and recall scores as high as possible. The metric of F1 has been mainly considered since it computes a geometric mean of precision and recall values.

Generally speaking, it is a known observation that phishers often resize the brand logos yielding much smaller or bigger logo or logo like images that exist in phishing web pages. These cases create a challenge for LogoSENSE even it analyses the whole image in terms of cropped windows for different scale-spaces. Therefore, instead of evaluating the proposed scheme against the wild-set image dataset within a single fixed zoom factor, we also attempted to test it on snapshots which were upsampled by $1.5\times$ and $2\times$ zooming factors. In other words, we have resized each test image in wild-set by these factors and have evaluated the system on these configurations.

As another important point, we also assessed the precision-recall-F1 scores by applying different detection threshold values ranging from 0 to 1 with 0.1 step size. By increasing the threshold value, we impose the detectors to be stricter on deciding whether a candidate image patch is a logo. In other words, the threshold parameter enables us to filter out weak predictions.

In line with these definitions, we have conducted 11 (threshold of 0.0 to 1.0) \times 3 (upsampling factors of 1, 1.5 and 2) = 33 experiments in order to reveal precision, recall and F1 scores for each threshold and upsampling parameters. Training of each detector has been carried out by using the training dataset of each brand. According to the results, we have plotted precision, recall and F1-score curves in Fig. 6. According to the experiments, the following findings have been revealed:

- If we assume the F1-score as the metric of success since it computes a balanced score between precision and recall, among the others, the best F1-score has been obtained as 0.8502 (precision: 0.9350, recall: 0.7794) by setting threshold value as 0.3 and up-sample factor of 2.
- Higher the zoom factor (ZF) by applying (1 \rightarrow 1.5 \rightarrow 2), stricter the threshold values we need in order to make precision scores that reach up to 1. In contrast, having higher upsampling factor causes better recall scores with less threshold value. This finding implies that there exist small logos that LogoSENSE could not detect when ZF=1 (original size). However, upsampling enables LogoSENSE to capture more *little* brand logos.
- Compared to the experimental results achieved from previous phases, it can be concluded that LogoSENSE produces slightly worse results when it is exposed to a more diverse real-world scenario. If we compare the best setting of this experiment (precision: 0.9350, recall: 0.7794) with the overall mean scores achieved in previous experiments, it can be inferred that the decrease in precision is less than the decrease in recall score. This finding is not a surprise because of LogoSENSE's default behavior sourcing from its default loss function design that causes it to be oriented to focus on precision rather than recall.

Apart from prediction capability, in terms of running time, mean detection duration for ZF=1 snapshots requires 0.4 s while ZF=2 images need 0.86 s. On average, 0.61 s is required when we set ZF as 1.5. Despite the increasing duration, as can be seen from Fig. 6, upsampling the input image yields better precision and recall scores along with better filtering out incorrect detections. Moreover, with upsampling, a trade-off between precision and recall should be considered.

6.5. Comparative study

In order to better understand whether the proposed scheme is efficient and effective, we first compared LogoSENSE with OpenCV based implementation of Wang et al.'s work (i.e., Verilogo (Wang et al., 2011)) which employs SIFT features to describe the key points of the target brand logos. Since they do not provide public source code or executable, we have implemented their work on C++ by sticking to their design and explanations. As an alternative, we also compared LogoSENSE with a state-of-art deep convolutional neural network based object detection and image segmentation approach named as Mask R-CNN He et al. (2017).

Verilogo must be equipped with a set of ground truth logos having different versions in order to build a corpus in a manner of whitelist. Next, it utilizes SIFT feature descriptors to be matched among the suspicious and ground truth brand logo images. If the strong matches are captured, Verilogo alerts the user. Nevertheless, picking strong matches requires fine-tuning of the parameters regarding SIFT based feature extraction phase. Thus we have first set the hessian parameter to 1000 to capture significant feature points. Meanwhile, it is noteworthy to point out that Verilogo works on horizontal overlapping image stripes extracted from whole content. To this end, we have cropped and processed overlapping windows having row heights of 200, 300, 400 and 500 pixels. Since most of the logo patches are higher than 100 pixels, we did not attempt

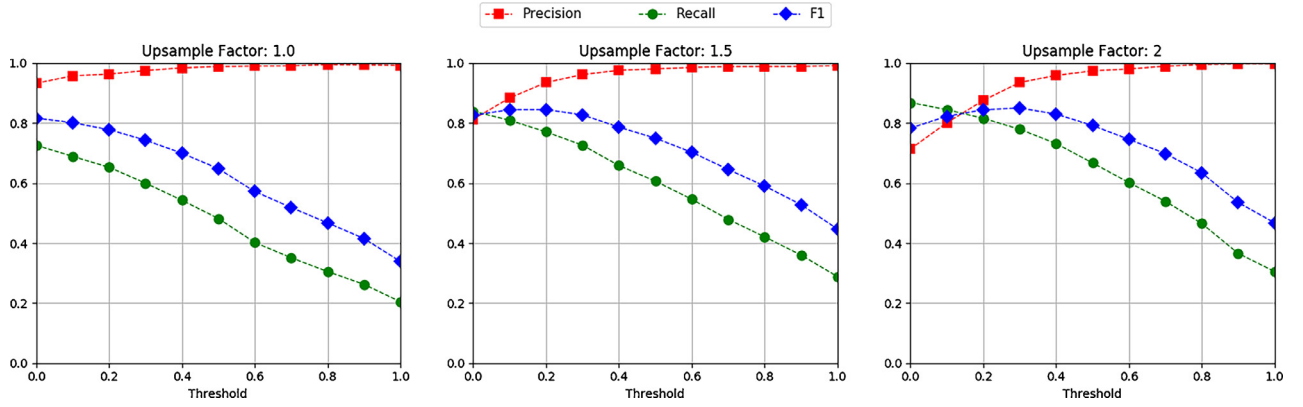


Fig. 6. Precision-recall and F1 charts for varying threshold values along with different snapshot up sampling ratios.

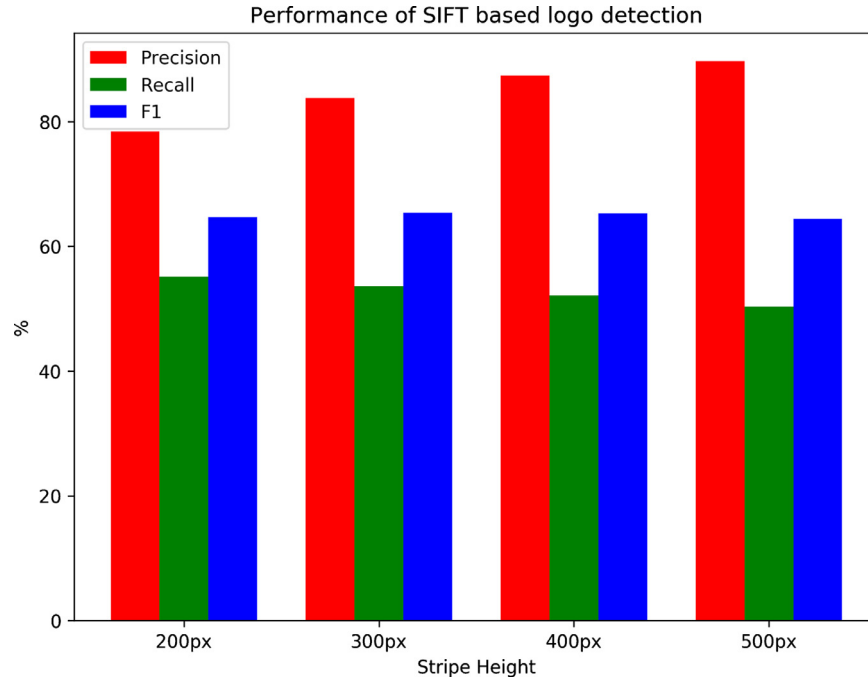


Fig. 7. Precision, recall and F1 scores of Wang et al. (Wang et al., 2011)'s work on wild-set corpus.

Table 6

Performance metrics of various stripe heights for Wang et al.'s work (Wang et al., 2011) on wild-set corpus.

Stripes	Precision	Recall	F1-score	Duration
200 pixels	0.8938	0.5522	0.6826	8.73 s
300 pixels	0.9233	0.5368	0.6789	8.73 s
400 pixels	0.9361	0.5222	0.6704	8.65 s
500 pixels	0.9583	0.5036	0.6602	8.9 s
Our best	0.9350	0.7794	0.8502	0.86 s

to set stripe height as 100 pixels. During the comparative study, we have employed the wild-set corpus. According to the directives and parameters, the outcome of the conducted experiments is presented in Table 6 and Fig. 7.

According to the results given in Table 6, the first finding is that though Verilogo presents competitive results in terms of precision scores, the proposed scheme outperforms Verilogo in terms of recall and F1 scores. Moreover, it can be easily seen that larger stripes affect the precision score in a positive way while reducing

the recall value. One of the reasons for this finding might possibly be related to providing Verilogo with more feature space causing it to detect irrelevant feature points along with yielding misdetections. In contrast, narrower stripes enable Verilogo to be more consistent during the matching stage.

Regarding the duration for detection given in Table 6, LogoSENSE requires much less time to complete detection reaching up 10× faster computation. Note that, given values refer to the average time taken to analyze input image. At this point, the benefit of the proposed approach comes into prominence. Since Verilogo aims to seek and match robust feature points extracted from all legitimate logo instances involved in a whitelist, its scalability is limited. On the other hand, LogoSENSE offers a more scalable solution in terms of running time since it only needs required number of detectors which will operate on the same extracted patch.

As pointed out in Section 2, deep learning and end-to-end representation learning have revolutionized the way of learning in tasks, especially for computer vision. Similarly, approaches dedicated to object detection problem have been also affected since the underlying computational model that is based on convolutional

Table 7

The parameters applied for Mask R-CNN during the training and inference stages.

Parameter	Value
Steps per epoch	1000
Backbone	Resnet 50 and Resnet101
Number of classes	16 (15 + 1 background class)
RPN NMS Threshold	0.7
Mini Mask Shape	(56,56)
Image Resize Mode	Square
Image Min Dim.	Varying (Tested with 384 and 640)
Image Max Dim.	Varying (Tested with 384 and 960)
Detection Min Threshold	Varying (Tested with 0.7, 0.8, 0.9, 0.95)
Batch Size	1 (i.e. One image per batch)
Learning rate	0.001
Weight decay	0.0001
Momentum	0.9

neural networks have significantly improved the obtained success in various metrics such as accuracy, precision, recall and mAP (mean average precision). Meanwhile, recent years have witnessed an arms-race between DL based object detection approaches in terms of obtained mAP values tested on large scale datasets such as PASCAL VOC (PascalDataset, 2019) and COCO (MSCOCO, 2020). During the recent years, various state-of-art object detection mechanisms such as R-CNN (Girshick et al., 2013), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2015), Single Shot Detection (SSD) (Liu et al., 2015), YOLO (Redmon et al., 2015) and Mask R-CNN (He et al., 2017) have been proposed. Among these methods, YOLO comes into prominence with its rapid detection property, whereas Faster R-CNN and Mask R-CNN perform better mean average precision. On the other hand, in terms of mAP, SSD falls between the YOLO and the Faster R-CNN. In this context, apart from the conventional scheme “Verilogo”, we have also compared our model with Mask R-CNN approach since it is an improved version of Faster R-CNN and introduces state-of-art detection performance in terms of precision and recall.

For this purpose, we have first annotated the dataset we collected via an open-source annotation tool named “Labellmg” (Labellmg Tool, 2020) (Asudeh and Wright, 2016) in order to label the dataset to be compatible with the PASCAL VOC format which is acceptable for well known state-of-art object detection models. Prior to this stage, we have identified and removed the exact copies of the images in the dataset (training and validation). Thus, we have obtained 870 *training* and 540 *validation* samples. In order to make a fair comparison, we did not change or modify the *test* samples involving 1979 instances. Next, we trained several models by experimenting with some parameters such as epoch number, input image size, underlying classification backbone (i.e., Resnet-50 and Resnet-101) and detection threshold. In this way, we have both compared the Mask R-CNN with our scheme and discovered the behavior of this state-of-art deep CNN based approach under different conditions.

For training, we have utilized a reliable Mask R-CNN implementation located at the URL “https://github.com/matterport/Mask_RCNN”. For the training stage, we employed the parameters listed in Table 7; hence we have not modified the rest of the parameters. The experiments carried on a computer equipped with an NVIDIA GTX 1050 TI 4GB GPU and 24 GB memory. Throughout the experiments, we have trained the models for 60 epochs. It should also be noted that our training regime fine-tunes a pre-trained Resnet-101 network by tuning the layers starting from 5th level and head of Regional Proposal Network. Our initial experiments have shown that the Mask R-CNN requires a large amount of GPU memory to work on large images. Since the size of the screenshots belonging to our dataset is larger than 1000 pixels in either height or width,

Table 8

Precision, recall and F1 scores obtained with the use of Mask R-CNN models on wild-set corpus. (on Resnet-101 backbone).

Inp. Res.	Det. Thrsh.	Prec.	Recall	F1-score
384×384	0.7	0.6701	0.8388	0.7451
384×384	0.8	0.7505	0.8195	0.7835
384×384	0.9	0.8469	0.7711	0.8072
384×384	0.95	0.9068	0.7058	0.7938
640×960	0.7	0.6714	0.8976	0.7682
640×960	0.8	0.7524	0.8815	0.8118
640×960	0.9	0.8550	0.8509	0.8529
640×960	0.95	0.9125	0.8154	0.8612
Our best	–	0.9350	0.7794	0.8502

we have first tested with the “none” setting for the parameter of *image resize mode*. However, this attempt has failed due to insufficient GPU memory. At this stage, we have configured the algorithm to resize (i.e. down-sampling) the input images along with relocating the coordinates of the ground truth bounding boxes. In this sense, we have trained our model by resizing the side length of the images as 384 pixels. Furthermore, in order to discover how the detection probabilities affect the prediction, we have set 4 different values for “detection threshold” parameter such as 0.7, 0.8, 0.9, and 0.95. Meanwhile, it took more than 8 h to train the model.

Our initial expectation was obtaining higher scores compared to the LogoSENSE. The outcome, however, fell behind the LogoSENSE across all the detection threshold values. The scores belonging to the first and second models are listed in Table 8. As can be seen, compared with the LogoSENSE, the first model (the first four rows) achieves higher recall rates. However, it has performed poorer precision values yielding worse F1-scores. Moreover, compared to Resnet-50, the use of Resnet-101 has yielded better precision, recall and mAP scores.

At this point, we have carefully investigated whole test images and detections. A careful investigation has clearly laid out that the first model detects false boxes which slightly mimic to original logos. Moreover, we explored that, in some cases, web page screenshots might be very confusing for a detection algorithm since web pages contain much more textual parts compared to natural images. Thus, for the brand logos including text, detections could fail. Likewise, we observed several false detections belonging to “Microsoft”, “Office” and “Facebook”. Due to obtaining relatively lower F1 score, we decided to build a second model by increasing the input image size such as 640×960. Therefore, we did not hurt the aspect ratio of input images and logo patches as well. The remaining parameters were kept the same. It took more than 15 h to train the second model due to the increased image size.

As can be inferred from Table 8, keeping the aspect ratio of the images closer to their original counterparts and increasing the input image size have significantly contributed to the recognition and detection quality. The best F1-score (0.8612) has been obtained by using the detection threshold of 0.95. However, it should be noted that the increased input image size (i.e., 640×960) has also increased the inference time. The first model takes 0.28 s for inference while the second one needs 0.94 s.

According to the results obtained by use of the second model, Mask R-CNN outperforms our scheme in terms of recall across all the values of detection threshold. Moreover, with the second model, it slightly outperforms LogoSENSE at the detection threshold values of 0.9 and 0.95. Nonetheless, LogoSENSE surpasses Mask R-CNN in terms of precision providing a competitive performance in terms of both F1 and running speed. Note that LogoSENSE does not require any GPU since it is compiled to run only on CPU and it

needs much less training time compared to a deep learning based solution.

7. Discussion

7.1. Effectiveness of logosense

As stated before, the main goal of LogoSENSE is to recognize brand logos of interest in phishing web pages and emails. Hence, we have created a logo dataset first and evaluated the performance of the proposed scheme in terms of both detection quality and running time. Today, the number of phishing content is rising exponentially. Therefore, it is becoming more crucial for an effective phishing detection system to run as real-time as possible. The reasons behind this necessity source from two facts: (1) in case of running at end-user computer, phishing detection must be done before user input the sensitive credential or clicks on a malicious link, (2) in case of running at a cybersecurity or web hosting company, in order to provide real-time protection, it must process tens of thousands of content (e.g., email) as soon as possible. For the case 1, LogoSENSE achieves its goal since it requires less than a second even in upsampling mode. However, time for analyzing an image by LogoSENSE has been affected by two factors: (1) image size and (2) upsampling zoom factor. Our observations point out that the cases where the height of phishing web pages become extremely high (e.g., ≥ 2000 pixels) dramatically reduce the detection speed. Besides, the upsampling process also requires a certain amount of time. Mixing up these two facts extends the running time of the scheme.

One another shortcoming about the logo detection quality of LogoSENSE comes from the sliding window approach. Though MMOD works in a multi-scale fashion, it visits lots of unnecessary portions of the visual content. This mainly affects the running time performance and degrades the precision score. In particular, logos located at corners or escaping from sliding windows pose challenges for the proposed approach. In order to overcome this problem, smart strategies such as *selective search* (Uijlings et al., 2013) could be incorporated as a startup process. In this way, possible logo positions can be forecasted that yields a reduction of the total amount of time for visiting candidate locations.

7.2. Robustness against various evasive attacks

Anti-phishing literature has witnessed numerous studies in the last two decades. Many of the works have performed a considerable amount of detection and recognition quality. As is stated before, most of the contemporary works deal with machine learning methods to predict suspicious cases. However, it should be remembered that anti-phishing is a security field and there always exists the risk for target-dependent evasion attacks. If the literature is reviewed, it can be seen that most of the studies have a lack of an analysis related to adversarial sampling attacks.

A very recent work of Shirazi et al. (Shirazi et al., 2019) has shown that the classification success of phishing detection schemes can be dropped up to 70% by manipulating only a single feature. Furthermore, the increase in the number of manipulated features (e.g., 4) enables phishing web pages by-pass the detection system. Therefore, currently, it is becoming more crucial for a security-oriented system to be tested against potential adversarial examples before its industrial employment. In their work, Corona et al. (Corona et al., 2017) have experimented on whether their suggested approach is robust against worse case scenarios (i.e., feature vector manipulations) by modifying the HTML content.

Apart from other kinds of adversarial findings, being a vision problem at heart, logo based phishing detection exposes several weaknesses. The findings reported in (Dhamija et al., 2006; Wang et al., 2011; Alsarnouby et al., 2015) and our preliminary visual investigations carried out on more than 12,000 phishing web pages shed light on them. We can simply categorize these vulnerabilities under two groups, namely (a) *inherent vulnerabilities* and (b) *adversarial attacks*. According to (Wang et al., 2011), the design of the logo and the branding strategy could cause those inherent weaknesses sourced from various reasons such as (1) visual variations in company logo, (2) logos having lack of details, and (3) logos mostly composed of text. On the other hand, adversarial attacks are primarily due to the malevolent modifications (e.g. rotation, shearing, cropping, scaling, color change) carried out on logos. Moreover, avoiding logo placement by phishers is another popular form of adversarial behavior.

Inherent vulnerabilities usually source from the logo design and branding policies followed by the owner company. Technically speaking, different versions of a brand logo could be challenging due to the higher intra-class variations yielding lower precision scores during the detection phase. For instance, in recent years, the “Paypal” company has used quite similar but also varying logotypes. This situation makes it essential to collect different visual signatures that capture unique edge, color as well as texture for each logo. At this point, it is generally believed that having unique visual cues for a logo makes it simpler to be detected. Besides, the “fingerprint” – the structure – of the logo should involve adequate cues to be identified. As reported in (Wang et al., 2011), “Orange” company employs a logo that lacks details when it is analyzed via SIFT algorithm. One another potential problem regarding company logos arises from “having designed with only texts”. This situation could make vision algorithms hard to distinguish the actual logo from unrelated text pieces unless they involve unique font faces or typography.

To date, researchers have identified different types of logo related adversarial attacks based on real-world cases. It is generally observed that the most employed evasion technique for phishers is to avoid from logo placement. In this way, attackers aim not to leave any logo related visual feature that can be captured. Besides, there exist various types of attacking methods which have been addressed in the literature. For instance, Asudeh and Wright (Asudeh and Wright, 2016) have tested the robustness of their logo detection system by applying noise, scaling and rotation effects on logo patches. According to their results, adding 10% salt and pepper noise caused more than 20% false negatives. Besides, they discovered that the HOG based detection is not invariant to rotation. Thus, they also suggested the use of intensity histograms along with HOG to alleviate the problem of rotational distortions. Wang et al. (Wang et al., 2011), similarly, have conducted a user-study based experiment with 11 participants in order to explore which type of logo specific visual deceptions would be perceived as suspicious. For this purpose, they have built an experiment corpus involving 49 unmodified web pages along with 16 phish pages having modifications in their logos. The results surprisingly indicate that 19.7% of the users reported unmodified pages as suspicious whereas 51.7% of them found modified pages as suspicious. More importantly, Wang et al. (Wang et al., 2011) reported that “poor aesthetics, or what appeared to be an unprofessional design, often contributed to the belief that a page was *phishy*”. In this context, they state that even if a user is not familiar to a brand or brand logo, certain transformations such as drastic shear or rotation may be perceived as aesthetically “problematic” Wang et al. (2011). So, for a phisher, it is a trade-off to apply any visual deformation, since it causes a decrease in credibility.

To this end, we have conducted an experiment by choosing a subset of our test corpus (i.e., 100 web pages – around 11%) by

Table 9

Performance evaluation of LogoSENSE (i.e. ours) and Mask R-CNN (i.e. DL) schemes against various distortion effects.

Scheme	Effect type	Precision	Recall	F1-score
Ours	No-effect (D)	0.873	0.820	0.846
Ours	Rotation (A)	0.988	0.664	0.794
Ours	Shearing (B)	0.977	0.656	0.785
Ours	Noise (C)	0.850	0.761	0.803
DL	No-effect (D)	0.970	0.738	0.838
DL	Rotation (A)	0.961	0.753	0.845
DL	Shearing (B)	0.971	0.768	0.858
DL	Noise (C)	0.970	0.746	0.843

Table 10

Performance evaluation for our scheme after improvement.

Scheme	Effect type	Precision	Recall	F1-score
Ours	Rotation (A)	0.948	0.694	0.801
Ours	Shearing (B)	0.977	0.656	0.785
Ours	Noise (C)	0.882	0.783	0.830

also making it include an equal number of brand logos. In line with the suggestions and practical guidelines presented at (Wang et al., 2011), we have separately applied different effects such as rotation (A), shearing (B) and noise (C) in order to modify the *small* test corpus. For the rotation, we have randomly rotated the logo patches from -20 to $+20^\circ$. Similarly, we have also distorted the logo regions by applying random shearing (i.e. $\pm 10^\circ$ for both horizontal and vertical axes). Finally, we have added 10% random RGB noise by utilizing GIMP tool shipped with Ubuntu Linux. It should be noted that, according to our observations, the existence of noise in phishing web pages is a rare case.

Apart from the manipulated web page sets (i.e. A, B, C) we have also included the unmodified *original* set (D) in order to evaluate the baseline performance of our scheme. Further, we have tested all these sets via our best Mask R-CNN (indicated as DL) model produced before. In Table 9, we presented the experimental results regarding each type of distortion and baseline for both our scheme and Mask R-CNN model. As can be seen, each of these manipulations hurt the performance of our scheme. Among the others, the shearing effect was found to be the most hurting modification. In contrast, deep convolutional neural network based Mask R-CNN model has performed much better in this experiment. We believe that, although we did not apply a data-augmentation during the training stage, generalization capacity and invariance to rotation and shearing of DL has played a key role.

In order to mitigate this issue, we have implemented a solution. First, we created two rotated (i.e. -10 and $+10^\circ$) versions of an unmodified screenshot and merged them with the original one to be processed at one shot. Moreover, rather than employing simple median filtering like in (Asudeh and Wright, 2016), we employed fast colored image denoising function “fastNlMeansDenoisingColored()” shipped with OpenCV for adaptive denoising. Nonetheless, we have not provided a solution for sheared logos. Following these improvements, we again measured the performance scores in terms of precision, recall and F1 and presented them in Table 10. As can be inferred easily, especially for rotation and noise effects, detection quality has risen in terms of recall and F1 scores. Nonetheless, the amount of time to process each image has also increased to 2.54 s from 0.86 s.

Note that, our proposal shows poorer performance when the shearing effect comes into prominence. For this reason, it can be deduced that deep learning based methods are more robust to well known logo-specific evasion techniques. According to our best knowledge, this small scale experiment is the first one which examines and distinguishes the performance differences between

deep learning and conventional methods in the context of phishing detection via logo detection.

7.3. Brand authorization

We have set the purpose of LogoSENSE as being a logo recognition based companion tool working together with existing phishing detection systems. The premise of this decision sources from the fact that not every logo containing web page and email is phishing. For instance, a web page belonging to a conference may include these brand logos to illustrate its sponsors. Therefore, predictions done by LogoSENSE do not provide solid evidence for visual content to be claimed as phishing. One another core point of this fact is that there exists no brand authorization scheme which “refers to a one brand enterprise that acknowledges a web site using its brand entity” Geng et al. (2015).

At this point, one may ask the question of “Is it possible to use only LogoSENSE as a single scheme for phishing detection?”. In a similar fashion, this issue has also been addressed in the works of Wang et al. (Wang et al., 2011) and Geng et al. (Geng et al., 2015). They proposed various possible and futuristic techniques in order to validate how the detection of a logo or any brand-related property may prove a real phish alert or decrease false positive rate (FPR).

The first approach proposed by (Wang et al., 2011) suggests brand holders embed a digital signature for their logo images. Furthermore, they have also proposed to create a specific DNS (Domain Name Server) record type. Nevertheless, these suggestions are currently not available and require revolutionary changes in today's Internet world. In contrast, in (Geng et al., 2015), three different features namely (i) name server and IP resolution, (ii) redirection information and (iii) incoming links were discussed and suggested. According to Geng et al. (Geng et al., 2015), each brand holder owns more than one domain name and its related domain name servers. Thus, they state that a comparison between the domain name of the suspicious web page and the legitimate brand yields a trustable source of evidence. Besides, they report that a suspicious web page redirecting to legitimate brand URLs is more likely to be a legitimate web page. Their last suggestion comes from the fact that a known targeted legitimate brand web site has several incoming links from other web pages. Thus, by use of query services of Alexa Internet, it could be possible to check whether a suspicious web page has incoming links from legitimate web pages. Though these assumptions and techniques do not still provide solid evidence for phishing accusation, they are useful to avoid misclassification of a legitimate web page that can be referred to reduce FPR.

In addition to techniques mentioned above, regarding the information obtained from LogoSENSE (i.e. coordinates of detected logo and name of the brands extracted from a phishing content) we suggest to use association rules mining – a data mining method to reveal frequent association patterns in data – in order to discover which associations or patterns exist in real-world examples. Our observations reveal that, apart from phishing contents containing a single brand logo, there exist numerous phishing cases involving more than one brand logo. This kind of associations along with coordinates on visual content enables us to extract useful patterns that can be used to degrade false positives. Furthermore, having an online learning method, it is possible to either add or update the rules with the help of a limited amount of human effort.

8. Conclusion and future work

In this work, we proposed a HOG based brand logo detection and recognition scheme for phishing web pages and e-mails so-

called “LogoSENSE”. In contrast to other studies, our proposed approach treats the issue of phishing detection as an object detection problem which originates from one of the core fields of computer vision. In this context, we achieved an underlying source code and language invariant detection and recognition module. In order to achieve this goal, we have incorporated King’s (King, 2015) MMOD schema into our approach. The core advantage of MMOD is that it requires much less positive samples and removes the need for hard negative sample mining. Moreover, we have collected, annotated and published a publicly available logo dataset for research purposes.

According to the experimental results, by using our novel dataset, LogoSENSE serves a feasible and promising solution by providing: (i) precision (0.9350), recall (0.7794) and F1-scores (0.8502, ii) a fast and scalable framework to capture phished brands located both in email and web pages. With the help of this scheme, we believe that existing phishing detection systems which require identification of the targeted brands will gain extra robustness. In addition, since it is capable of detecting multiple logos on an image, there exist some other potential fields for LogoSENSE to be used such as logo retrieval from scanned documents or car logo recognition in videos.

Consequently, though it provides promising results, the proposed scheme has some limitations due to the semi-rigid representation of HOG features. To overcome this problem, we are planning to design and implement a deep convolutional neural network based detection model which is robust to geometric distortions as a future work along with extending the dataset. A second future work for us is to integrate the next version of the LogoSENSE into a character-level CNN based URL classification scheme in order to obtain a complete anti-phishing solution rather than a companion scheme. Last but not least, a third contribution would be the development of a self-logo-learning scheme by utilizing manifold learning and image classification in order to self recognize updated versions of the brand logos. This kind of expansion will also be useful for the recognition of logos that were not included in the corpus.

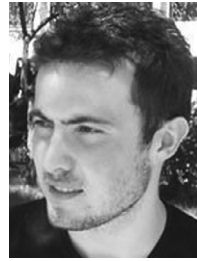
Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Alsarnouby, M., Alaca, F., Chiasson, S., 2015. Why phishing still works: user strategies for combating phishing attacks. *Int. Journal of Human-Computer Studies* 82.
- Asudeh, O., Wright, M., 2016. POSTER: phishing Website Detection with a Multi-phase Framework to Find Visual Similarity. In: *In the Conference: 2016. ACM SIGSAC*.
- APWG, Phishing activity trends paper. [Online]. Available at [http://www/antiphishing.org/resources/apwg-papers/](http://www.antiphishing.org/resources/apwg-papers/), 2018.
- Bay, H., Tuytelaars, T., Gool, L.V., 2008. “Speeded-up robust features (SURF). *Computer vision and image understanding* vol.110, 346–359.
- Bianco, S., Buzelli, M., Mazzini, D., Schettini, R., 2017. Deep learning for logo recognition. *Neurocomputing* 5, 23–30.
- Bozkir, A.S., Akcapinar Sezer, E., 2016. Use of HOG Descriptors in Phishing Detection. In: *In 4th International Symposium on Digital Forensic and Security (ISDFS)*.
- Chen, K.T., Huang, C.R., Chen, C.S., 2009. Fighting Phishing with Discriminative Keypoint Features. *IEEE Computer Society*.
- Chiew, K.L., Chang, E.H., Sze, S.N., Tiong, W.K., 2015. Utilization of Website Logo for Phishing Detection. *Computers & Security* 54, 16–26.
- Common Object in Context (COCO) Dataset, [Online]. Available at <http://cocodataset.org/#home>, 2020.
- Corona, I., Biggio, B., Contini, M., Piras, L., Corda, R., Mereu, M., Mureddu, G., Ariuand F. Roli, D., 2017. DeltaPhish: detecting Phishing Webpages in Compromised Websites. In *European Symposium on Research in Computer Security* 370–388.
- Dalal, N., Triggs, B., 2005. Histogram of Oriented Gradients for Human Detection. *Computer Vision and Pattern Recognition*.
- Dalgic, F.C., Bozkir, A.S., Aydos, M., 2018. Phish-IRIS: a New Approach for Vision Based Brand Prediction of Phishing Web Pages via Compact Visual Descriptors. In: *In 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies*. Ankara, Turkey.
- Dhamija, R., Tygar, J.D., Hearts, M.A., 2006. Why phishing works? In: *In SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, pp. 581–590.
- Dunlop, M., Groat, S., Shelly, D., 2010. GoldPhish: using images for content-based phishing analysis. In: *In Proceedings of the 5th International Conference on Internet Monitoring and Protection*.
- Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D., 2010. Object Detection with Discriminatively Trained Part Based Models. *IEEE Trans Pattern Anal Mach Intell* 32.
- Firefox Browser, [Online]. Available at <https://www.mozilla.org>, 2020.
- Fu, A.Y., Wenyn, L., 2006. Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover’s Distance (EMD). *IEEE Trans Dependable Secure Comput* 3.
- Geng, G.-G., Lee, X.-D., Zhang, Y.-M., 2015. Combatting phishing attacks via brand identity and authorization features. *Security and Communication Networks* 8, 888–898.
- Girshick, R., Donahue, J., Darrell, T., Malik, J., 2013. “Rich feature hierarchies for accurate object detection and semantic segmentation”, arXiv:1311.2524.
- Girshick, R., 2015. Fast R-CNN. In: *In the IEEE International Conference on Computer Vision (ICCV)*.
- Google Chrome Browser, [Online]. Available at <https://www.google.com/chrome/>, 2020.
- He, K., Gkioxari, G., Dollar, P., Girshick, R., 2017. “Mask R-CNN”, arXiv:1703.06870.
- Imran, M., Afzal, M.T., Qadir, M.A., 2017. A comparison of feature extraction technique for malware analysis. *Turkish Journal of Electrical Engineering & Computer Sciences* 25, 1173–1183.
- Indola, F.N., Shen, A., Gao, P., Keutzer, K., 2015. “DeepLogo: hitting Logo Recognition with the Deep Neural Network Hammer”, arXiv:1510.02131.
- Jain, A.K., Gupta, B.B., 2017. Phishing Detection: analysis of Visual Similarity Based Approaches. *Security and Communication Networks*.
- Jain, A.K., Gupta, B.B., 2018. “Detection of phishing attacks in financial and e-banking websites using link and visual similarity. *International Journal of Information and Computer Security* 10 (4), 398–417.
- Kausser, F., At-Otaibi, B., Al-Qadi, A., Al-Dossari, N., 2014. Hybrid client side phishing website detection approach. *International Journal of Advanced Computer Science and Applications* 5, 132–140.
- King, D.E., 2009. Dlib-ml: a machine learning toolkit. *International Journal of Machine Learning Research* 10, 1755–1758.
- King, D.E., 2015. “Max-Margin Object Detection”, arXiv:1502.0046v1.
- Labelling Tool, [Online]. Available at <https://github.com/tzutalin/labelling>, 2020.
- Lam, I.F., Xiao, W.C., Wang, S.C., Chen, K.T., 2009. Counteracting Phishing Page Polymorphism: an Image Layout Analysis Approach. In: *In International Conference on Information Security and Assurance*.
- Lazebnik, S., Schmid, C., Ponce, J., 2006. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition (CVPR)*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2015. “SSD: single Shot MultiBox Detector” arXiv:1512.02325.
- Liang, C.W., Huang, C.F., 2015. Moving object classification using local shape and HOG features in wavelet-transformed space with hierarchical SVM classifiers. *Appl Soft Comput* 28.
- Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P., 2017. “Focal loss for Dense Object Detection”, arXiv:1708.02002.
- Llorca, D.F., Arroyo, R., Sotelo, M.A., 2013. Vehicle logo recognition in traffic images using HOG Features and SVM. In: *In Proc. International IEEE Conference on Intelligent Transportation Systems*.
- Lowe, D.G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int J Comput Vis* vol.50, 91–110.
- Mao, J., Tian, W., Li, P., Wei, T., Liang, Z., 2017. Phishing-Alarm: robust and efficient phishing detection via page component similarity. *IEEE Access* 5, 17020–17030.
- Moghim, M., Varjani, A.Y., 2016. New rule-based phishing detection method. *Expert Systems with Applications* 53, 231–242.
- Moore, T., Clayton, R., 2007. Examining the Impact of Website Take-down on Phishing. In *eCrime’07 Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*.
- OpenCV, [Online]. Available at <http://opencv.org/>, 2020.
- Oliveira, G., Frazao, X., Pimentel, A., Ribeiro, B., 2016. Automatic Graphic Logo Detection via Fast Region-based Convolutional Networks. In *IJCNN 2016*.
- Pascal VOC Dataset, [Online]. Available at <http://host.robots.ox.ac.uk/pascal/VOC/>, 2019.
- Phishtank, [Online]. Available at <https://www.phishtank.com/>, 2020a.
- Phishbank, [Online]. Available at <https://phishbank.org/>, 2020b.
- Ramanathan, V., Wechsler, H., 2012. phishGILLNET – phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training. *EURASIP Journal on Information Security*.
- Rao, R.S., Ali, S.T., 2015. PhishShield: a Desktop Application to Detect Phishing Webpages through Heuristic Approach. *Procedia Comput Sci* 54, 147–156.
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2015. “You Only Look Once: unified, Real-Time Object Detection”, arXiv:1506.02640.
- Ren, S., He, K., Girshick, R., Sun, J., 2015. “Faster R-CNN: towards Real-Time Object Detection with Region Proposal Networks”, arXiv:1506.01497.
- Rosiello, A.P., Kirda, E., Kruege, C., Ferrandi, F., 2007. A layout-similarity-based approach for detecting phishing pages. In: *In Proceedings of the 3rd International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm ’07)*.

- Sathish, S., Thirunavukarasu, A., 2015. Phishing Webpage Detection for Secure Online Transactions. *International Journal of Computer Science and Network Security* 15.
- Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J., Zhang, C., 2009. An empirical analysis of phishing black-lists. In: *In Proceedings of the 6th Conference on E-Mail and Anti-Spam*.
- Shirazi, H., Bezawada, B., Ray, I., Anderson C., "Adversarial Sampling Attacks Against Phishing Detection", arXiv:1805.12177v2, 2019.
- Su, H., Gong, S., Zhu, X., 2018. "Scalable Deep Learning Logo Detection", arXiv:1803.11417.
- The-Chung, C., Torin, S., Scott, D., James, M., 2014. An anti-phishing system employing diffused information. *ACM Transaction on Information and System Security* 16, 1–31.
- Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M., 2013. Selective Search for Object Recognition. *Int J Comput Vis* 104, 154–171.
- Varshney, G., Misra, M., Atrey, P.K., 2016. A survey and classification of web phishing detection schemes. *Security and Communication Networks* 9, 6266–6284.
- Wang, G., Liu, H., Becerra, S., Wang, K. "Verilogo: proactive Phishing Detection via Logo Recognition," Technical Report CS2011-0669, UC San Diego, 2011.
- Xiang, G., Hong, J., Rose, C.P., Cranor, L., 2011. CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security* 14.
- Xiang, G., 2013. Ph.D. Thesis. Carnegie Mellon University.



AHMET S. BOZKIR was born in Muğla Turkey, in 1983. He received the B.S. degree in computer engineering from the Eskişehir Osmangazi University in 2002. Besides, he received M.S. and Ph.D. degrees in computer engineering from the Hacettepe University in 2009 and 2016 respectively. He is currently working as a Research Assistant with at Hacettepe University Multimedia Information Laboratory (HUMIR). He has more than 25 studies covering fields such as Information Security, Human Computer Interaction, Information Retrieval, Machine Learning and Engineering Geology.



MURAT AYDOS. Dr. Murat Aydos received the B.Sc. degree from Yildiz Technical University (Turkey) in 1991, and M.S. degree from Electrical and Computer Engineering Department, Oklahoma State University (USA), in 1996. He completed his Ph.D. study in Oregon State University, Electrical Engineering and Computer Science Department in June 2001. Dr. Aydos joined Informatics Institute at Hacettepe University in April 2013. He is the Head of Information Security Division at the Informatics Institute. Dr. Aydos is the author/co-author of more than 30 technical publications focusing on the applications of Cryptographic Primitives, Information & Data Security Mechanisms.