Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Website categorization via design attribute learning

Doron Cohen, Or Naim, Eran Toch, Irad Ben-Gal*

Tel Aviv University, Israel

ARTICLE INFO

Article history:
Received 3 October 2020
Revised 26 March 2021
Accepted 23 April 2021
Available online 13 May 2021

Keywords:
Online learning
Crack websites
Malware
Website categorization
Website design elements
Cyber security
Human computer interaction

ABSTRACT

Malicious software (malware) is a challenging cybersecurity threat, as it is often bundled with legitimate software and downloaded by naïve users. A significant source of malware downloads is via crack websites that are used to circumvent copyright protection mechanisms. Crack websites often change URLs and IPs to avoid automatic detection; however, in many cases, they preserve specific visual designs that signal the website's function to potential users (such as particular colors, text fonts, shapes, and sizes.). Website design features are numerous, have high dimensionality and complicated interactions, making categorization challenging. This study shows that straightforward machine learning models for categorizing Crack and Malicious websites can considerably benefit from using design features. We report on two experiments based on unbalanced datasets and show that classification by using design features can reach a categorization accuracy of over 90% with an F1-score over 77% in some instances. Finally, we discuss the results in the context of developing intelligent security mechanisms.

1. Introduction and background

Humans are notorious for being the weakest link in cybersecurity (Pfleeger et al., 2014). Human factors influence how individuals interact with cybersecurity systems and are the cause of many successful cyberattacks. For example, 73% of technology professionals perceive user errors to be one of the top three information security threats (Deloitte, 2013). Human vulnerability to cyberattacks can expose large, medium, and small organizations to risks by installing malware and letting unauthorized software inside the organization's network. Various advanced approaches, such as automating dynamic malware analysis tools (Shahegh et al., 2017), detecting attempted attacks (Carlini and Wagner, 2017), and identifying phishing websites based on suspicious webpage features (Moghimi and Varjani, 2016), have been developed to protect users from potentially harmful websites. Additionally, efficient privacy-preserving tools, such as Sharemind (Bogdanov et al., 2012), intrusion detection systems (Hadžiosmanović et al., 2012), and website vulnerability detection systems (Yue and Wang 2013), have been developed to secure organizational data and processes.

Despite cybersecurity efforts, one of the most popular ways to spread malware is through crack websites, which allow users to download free software that circumvents antipiracy mechanisms (Krebs, 2011]. A crack website is defined as a website that disrupts a client's computer operations, gathers sensitive information, or is used to gain access to private computer systems. This broad definition covers websites that aim to attack browsers through JavaScript vulnerabilities (Heiderich et al., 2011), phishing users (Zhang et al., 2014), and so forth.

Crack websites attract visitors by offering free software, games, movies, or music and then redirecting naïve users to malicious sites (Zhuge et al., 2009). If successful, malware

* Corresponding author.
E-mail addresses: doroncohen1@mail.tau.ac.il (D. Cohen), ornaim@mail.tau.ac.il (O. Naim), erant@tauex.tau.ac.il (E. Toch), bengal@tauex.tau.ac.il (I. Ben-Gal).
https://doi.org/10.1016/j.cose.2021.102312

is often installed on the users' machines. Software cracking tools are heavily used by cyber criminals to spread malware and that existing virus scanners cannot fully protect users from these threats, as users often circumvent those scanners (Kammerstetter et al., 2012). The authors also found that more than 50% of crack websites included a certain type of malicious software (malware). Crack websites thus pose a serious challenge to cybersecurity. The malicious software from crack websites is actively and directly downloaded by users, making it more difficult for automatic network and computer security tracking mechanisms to provide efficient protection against malware damage.

Various methods have been suggested to identify malicious websites: sdetecting mobile malicious webpages (Amrutkar et al., 2017, Cimino et al., 2020), analyzing malicious URLs (Ma et al., 2011, Kim et al., 2018, Chiba et al., 2018), analyzing the properties of software linked to the website (Egele et al., 2008, Liu et al., 2020, Fang et al., 2020), capturing user navigation paths (Shahabi et al., 1997; Spiliopoulou et al. 1998), propagating trust of the website (Zhang et al., 2014) or detecting attacking attempts (Heiderich et al., 2011). However, crack websites change their URLs and hosting services frequently, relying on both general and dedicated search engines to attract visitors. Therefore, URL detection methods and malware directories are in a constant "arms race" with those websites, never catching up with these threats. Also, because crack websites are continuously changing (Motoyama et al., 2011, Samtani et al., 2015), it is difficult to identify them and to protect organizations and users against these threats. Indeed, no single method was found to identify all crack websites; therefore, a battery of tools are used to improve these websites' detection. This study aims to improve the identification of cracks and malicious websites in particular and websites' categorization in general by using website design features that are often retained across multiple iterations of a given crack website, as explained below.

Protecting against malware and malicious software is a critical task in supporting business service processes (Knight et al., 2007). However, in many cases, the protection of the software itself is vulnerable and prone to attacks, for example, by overloading antivirus tools (Al-Saleh et al., 2015). The threat of phishing attacks has drawn much attention, and several studies have tried to identify URLs, suspicious links, HTML structures, or visual design characteristics that are typically associated with phishing websites (Lakshmi and Vijaya 2012; He et al., 2011; Aburrous et al., 2010; Chen et al., 2014). However, the relevance of these studies to identify crack websites is questionable. Phishing website detection relies primarily on the fact that these websites try to imitate legitimate websites (e.g., e-banking or commerce sites) using similar URL and HTML structures. This is not necessarily the case for crack websites, which are seemingly "normal" websites that often contain malicious software.

An important observation that can be used to improve crack website detection is that these websites must balance two opposing requirements to successfully function: escaping malware detection tools and attracting visitors (Kammerstetter et al., 2012). When analyzing this question from the perspective of crack website operators, one can ask how a website can signal potential visitors that it is distributing cracks while avoiding exposing similar signals to cybersecurity tracking tools. Signaling theory has been used to identify and understand the cues (i.e., signals) people use to assess a website's functions when they are provided with limited information about the website (Pavlou et al., 2007). Specifically, visual design was a strong signal of the way visitors perceive a website's quality (Wells et al., 2011) and security functionalities (Pavlou et al., 2007).

Visual design factors are important in establishing a relationship between websites and users. The design strongly affects visitors' trust in websites (Pelet and Papadopoulou, 2011), frames the expectations of the website's function (Cebi 2013) and triggers visitors' selections and surfing patterns (Bonnardel et al., 2011). Concurrently, these design factors might also represent unseen visual elements, which are elements that cannot be seen by users but can be found in the website's code. One example of an unseen visual element is text words surrounded by a background that is the same color, which hides links that lead to malicious content. Cyr et al. (2010)) showed that websites' coloring could affect the factors of trust and satisfaction across different users from different cultures. The authors also found significant cultural differences in the way that users interact with different websites (Cyr 2008; Cyr and Trevor-Smith 2004). A website's visual design and aesthetics primarily use a relatively small number of styles, which differ between domains and follow different trends (Golander et al. 2012). Accordingly, we claim that the visual design of a website can serve as a dual-purpose signal. People can easily identify and detect the visual properties of these websites based on intuition (e.g., when examining four different crack website examples in Fig. 1). However, identifying and analyzing websites' visual design is a relatively difficult task for computer algorithms (Cai et al., 2003). Because it is critical for crack websites to be recognizable by users (Motoyama et al., 2011), this study hypothesizes that the visual design of crack websites have their own visual characteristics and signatures and are similar across different websites. Although there is no guarantee that crack websites will not use the visual designs of legitimate sites that are easily implementable, these design features bring traffic to the websites, and that traffic might be reduced if the design features are modified.

Signaling the functionality of crack websites through their visual design is an assumption we rely on when using machine learning procedures, even simple ones such as classification and regression trees (CART) (Breiman et al., 1984), for the modeling and analysis of visual features that enable the automatic identification of cracks and malicious websites. CART modeling is known to be a straightforward machine learning model that is ideally suited to generate various logical decision rules. CART modeling is often effective at uncovering hidden interactions among predictors (website design features in this case), which may be difficult to identify when using traditional multivariate techniques (Lewis 2000, Steinberg and Colla 2009).

In this study, we propose a method that is based on the abovementioned assumptions and evaluates whether the classification of websites' visual elements can be used to predict crack websites or even to classify website categories. Exploring this question requires formal definition and extraction

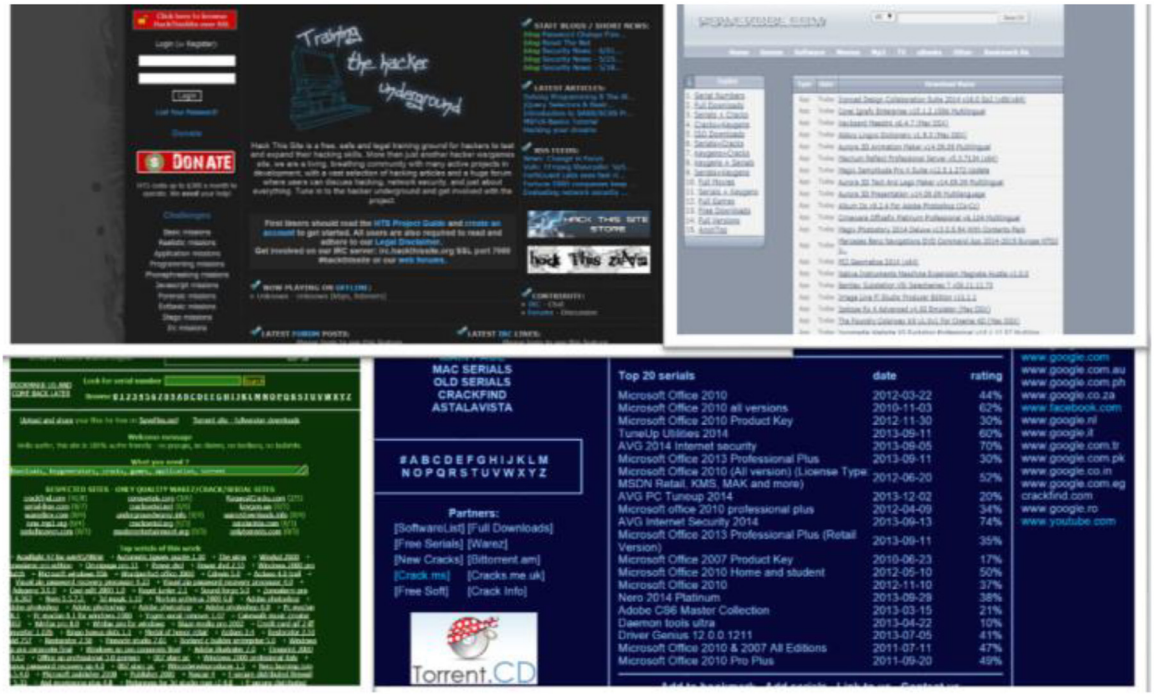

Fig. 1 – Screenshots of four examples of crack websites that have been detected by the proposed algorithm.

of website design elements as potentially relevant features, and the design of a web-mining method that can use these features for identification purposes. For this task, we propose an efficient algorithm that scrape the websites' design and style elements, processes the elements and stores relevant features in an accessible dataset that can be used for learning and identification. The stored information is then processed while using standard classification techniques to classify the website category, including crack websites.

This study is focused on the extraction and use of new visual design features, which are often overlooked in the literature, although they are readily available and apparently highly informative, to classify and identify websites. We purposely relied on standard and widespread machine learning methods that can be easily analyzed and translated to a straightforward set of rules. In particular, these algorithms were selected as representatives of simple and popular models in ML, namely, Logistics Regression as a simple parametric model, KNN as the most popular non-parametric ML model, Neural network as a demonstrative model from a family of biologically-inspired networks, AdaBoost as a pioneering ensemble-based model of weak classifiers, and CART as one of the most popular classification and regression tree models. Finally, as part of this study's contributions, we publish a new dataset that contains hundreds of thousands of website features.

Several web-mining methods have been developed in recent years to improve website personalization, provide security, maximize sales, and analyze visitor use patterns. Some studies provide a solid basis for addressing the technical aspects of analyzing website design, going beyond the document object model (DOM) that is commonly used to represent HTML-based web pages. Studies have demonstrated how to combine DOM scraping with SimHash fingerprinting (hashing technique) and agglomerative clustering (Bernardini 2018) to identify illicit websites. Sarhan et al. suggested a method for the automatic classification of a website into a phishing or legitimate website based on the aggregation of a set of predetermined features related to the site's content (Sarhan et al., 2017).

Wu et al. suggested learning methods based on summarizing various visual features to assess the visual complexity of a website (Wu et al., 2013; 2016). Mesbah et al. demonstrated that it is possible to analyze user interaction on websites that use asynchronous JavaScript and XML (Ajax) for dynamic background HTTP calls (Mesbah et al., 2012). All these techniques have the potential to enhance the proposed identification method in this study.

In this paper, we conduct two validation experiments. The first experiment is based on a manually selected dataset of 450 websites and identifies known website categories, such as "Crack", "Shopping", "Games", "News" and "Search". This experiment aims to provide a proof of concept that verifies that website design features can be used to identify general website categories. Then, in the second experiment, which focuses on identifying malicious websites, the algorithm is fine-tuned to analyze a larger feature set.

The remainder of this paper is organized as follows. Section 2 describes the proposed method and explains how it can be used to access, analyze and categorize websites based on their design features. Section 3 describes the experimental settings and evaluation method of the study, Section 4 presents the experimental results. Section 5 discusses the results (specifically within the framework of cybersecurity mechanisms,) and Section 6 concludes the paper.

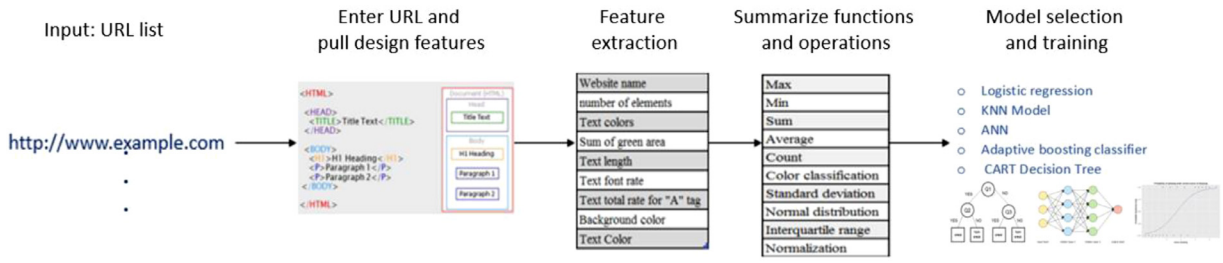

Fig 2.1 – Feature extraction and learning process scheme used in Experiment 1.

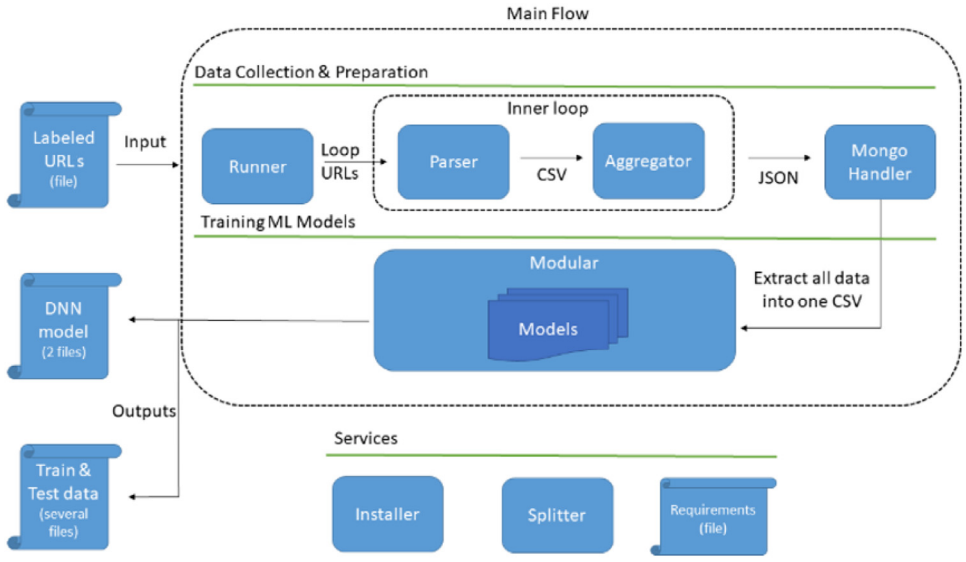

Fig 2.2 – Flow diagram of Experiment 2, which builds a robust algorithm to allow large-scale website collection.

2. Proposed website assessment scheme

This section describes the proposed scheme and the developed procedure to access and analyze a list of webpage design features. The implementation is based on design elements of the webpage, where the word "design" refers to all visual and nonvisual elements (e.g., hidden links or text) on a webpage and their related features. We assume (and evaluate) that these design features can serve as useful identifiers for improving the identification accuracy of website types or categories, including crack websites. In the first experiment, the procedure relies on the feature extraction of websites' landing pages and assumes that crack websites must quickly signal their functionality to potential visitors. As shown in Figs. 2.1 and 2.2, the proposed application receives a list of URLs as an flat input file. It thenaccesses the list, extracts websites' primary design features, processes the information, and then generates a simple machine learning classifier that models the combinations of features that define website categories as "Crack" and "Not Crack" websites.

Fig. 2.2 shows a schematic view of the proposed system that was used in Experiment 2. The algorithm obtains a list of URLs as its input, downloads the website at each URL, examines the HTML elements and then extracts their content and design features. A Document Object Model/JavaScript (DOM/JS) engine is used to access all DOM elements on a webpage. One of the primary difficulties in the construction of the DOM/JS sub-algorithm is the ability to run it from the client-side on external websites. Therefore, we have implemented a PHP-based server-side engine that downloads an HTML file and related style libraries, and saves the files on the server side. These operations allow access to websites, download related files that affect the website design and save those on the server. In the second step, the algorithm extracts elements from the webpage, ranks them based on their size, and then extracts their primary design features, as shown in Fig. 3. When using cascading style sheets (CSS), a parent element's style definition is applied to all descendant elements. For example, if a certain element background color is set as "blue", and this element is defined as a table, its descendant cells will have a blue background as well.

However, if one of those cells is set separately with a different background color, this background will be shown as defined. Accordingly, the proposed algorithm infers the style elements with respect to the CSS inheritance tree. In the third step of the procedure, the metadata, including keywords, titles and tags that are used to enrich the identification engine, are extracted from the website, as explained below.

Table 1 describes the information that is extracted from the website following the implementation of the PHP and

Table 1 – Website features example.

Feature name	Description	Units
Website address	Website domain name: "www.example.com"	Nominal value
Element tag name	Each element in HTML has a tag name. The tag name represents the type of element (for example, table, heading, image, etc.).	Nominal value
Element area	Calculation of the area for each HTML element (height X width)	Pixel
Metakeyword	The presence of a relevant keyword on the website	Binary
Text length	The text length ignoring spaces. For example, "This example!" has a text length of 12.	Number
Text font rate	Acquires the font size of the element that contains text. The font size is measured in pixel units.	Pixel
Text total rate	Text Total rate = Text length x Text font rate x Bold factor*. *If the text is in bold, we multiplied the function by 1.5.	Pixel
Background color	The final background color associated with the element, as shown in the browser to the internet surfer.	R,G,B
Text Color	The final text color associated with the element, as shown in the browser to the internet surfer.	R,G,B

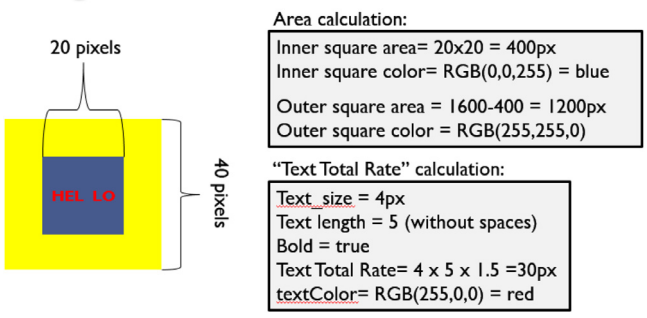

Fig. 3 – Area and text calculation example.

DOM engines. Finally, following all visual elements' extraction, the proposed algorithm also calculated new summarized attributes for each website feature using common descriptive statistic functions. We have used functions such as max, min, sum, average, count, color classification, standard deviation, normal distribution, interquartile range and normalization by the deviation of the features. Using these standard functions, this preprocessing stage results in a total of 1198 design-related features for each website page.

In the second experiment, which is described in Section 4.2, the proposed procedure is refined to support the robust analysis of large-scale datasets. Additionally, the algorithm's capabilities are extended to support various website structures and to address more design features, such as the XY position of an element, word count, color allocation and color classification. Overall, the proposed scheme shown in Fig. 2.2 extracts and scans 2522 features.

3. Experimental settings and evaluation

The proposed algorithmic engine was designed to perform a full scan of one website within a few seconds. In the first experiment, we identify "Crack" vs. "Not Crack" websites. As a baseline, and to obtain a tagged dataset for "Not Crack"

websites, we used Google's top 1000 websites in the highest-rated website categories, "Shopping", "Games", "News" and "Search" websites. A total of 579 websites were excluded from the list (e.g., websites that do not use HTML or CSS while using RGBA and server timeout). Following this extraction, 421 carefully selected websites remained in the negative training dataset (i.e., "Not Crack" websites), following a manual inspection. For the "Crack" websites, we used the top 60 websites which appeared in Google's "crack websites" search. Thirty-one crack websites were excluded from the set using the same procedure described above. The remaining "Crack" websites were checked manually to ensure that they actually did distribute crack content. Then, we created a training dataset with 450 carefully selected websites, which resulted in a relatively small and unbalanced dataset that was fairly accurate due to rigorous manual inspection to verify the studied websites' categorization. We found that a small yet accurate dataset that focused on evaluating the information embedded in websites' design features was sufficient for this study.

In the next stage, we tagged each website in the dataset with its category type. This categorization enabled us to examine whether website design features can be used for automated identification of a website's category. Then, we built a decision/classification tree for each category by considering either the design feature alone or a larger set of features that contains both the design features and the metakeyword features. The primary goal of the first experiment was to examine whether design features can be used to improve the identification of website categories, specifically crack websites, while using simple classification models such as decision trees. We used the classification based on design features and meta-keywords as a baseline benchmark because many crack websites apply specific keywords to attract traffic from mainstream search engines. The contribution of this work is based on the extraction and use of design features that are otherwise overlooked and not on the proposal of new learning models.

In the second experiment, we used a free dataset published for researchers on the *UK Web Archive* and scanned 14,922 websites that are categorized by type. We randomly checked 50 websites to validate the accuracy of the categorization, and no discrepancies were found. We added 510 URLs that contain malware content to this list and used the "Google Safe Browsing" (GSB) API to classify and tag each website. GSB classifies a malicious URL into one of the following five classes: "Malware", "Social engineering", "Unwanted software", "Potentially harmful application", and "Threat type unspecified". To obtain a high-quality dataset, we only used URLs that were tagged as "Malware" or "Unwanted software". Then, we executed an additional retrospective inspection of these URLs to verify that they continued to appear on the website list as malicious. Thus, we reduced the chance of having a misclassified URL in the malicious website learning dataset. Finally, we obtained a dataset of 15,432 URLs. Then, after tagging the dataset, we used five well-known machine learning models to classify website categories: logistic regression, k nearest-neighbors (KNN), an artificial neural network (ANN), an adaptive boosting classifier, and a CART decision tree with 5-fold cross-validation.

The proposed algorithm was designed to run on several cores simultaneously to reduce the overall scanning time. Each of the 8 cores obtained a subset of sites from the URL list, scanned the websites and wrote the results into the same data schema. The execution of this algorithm for over 15,000 websites took approximately 1.5 hours per core (with a mean time of 2.8 seconds per website per core). Two factors (the website destination and the scheme size) were found to impact the required scanning time of the website. These factors affected both the required loading time and rendering time of the site. The average scanning time was also affected by the category of the website, depending on the number of objects. For example, a typical "News" website often consists of a relatively high number of HTML objects and required a scan time of up to 5 seconds, while a typical "Search" website often consists of fewer elements and required a scan time of below 1 second. These running times were obtained with an average download rate of less than 20 Mbps, which are much slower than the download rate of commercial packages that are available on the market. Thus, the algorithm runtime can be decreased using faster download rates, faster ports and more computing power.

4. Results

4.1. *Results of Experiment 1*

We used the classical J48 decision tree (Salzberg, 1994) which is based on information gain, wherein each level starting from the root, the attribute with the highest normalized information gain is selected to obtain a decision node. The proposed procedure enables to generate a simple and interpretable set of conditional statements on the design features of different website categories.

To analyze the performance of the proposed approach, we calculated the classification accuracy, including the true positive (TP) rate and the true negative (TN) rate, for each of the website categories by each classifier based on the design features alone or on both the design features and the meta-keywords together. Table 2 shows the confusion matrix, classification accuracy, TN rate, precision, TP rate (recall) and F1-score for all instances that were included in Experiment 1. These metrics were selected because they are often used to evaluate classifications in unbalanced datasets, where certain predicted categories are minority classes (Pouyanfar et al., 2018; Al-Azani and El-Alfy 2017). The upper-left entry in each matrix indicates the number of correctly tagged websites with their category type (TP), and the lower right entry in each matrix indicates the number of websites correctly tagged as not belonging to the category type (TN). The classification accuracy was calculated as (TP+TN)/N. Results show that design features alone could successfully classify crack websites with an average accuracy of 90.7%.

The TP rate (Recall) of "Crack" websites was the highest among all the categories (62.1%), while the other categories obtained the following TP rates: "Search" (60%); "News" (48.8%); "Shopping" (35.7%); and "Games" had the lowest rate (23.1%). The true negative rate of "Crack" website identification was 92.6%, with the following values for the other cat-

Table 2 – Confusion matrix, classification accuracy and tp rates based on site categories*.

Category	Minority class proportion	Classification Type	Design Alone (Actual)		Design & KW (Actual)	
Crack	11%	Confusion Matrix	18	31	22	3
		(predicted)	11	390	7	418
		Accuracy	90.7%		97.8%	
		TN Rate	92.6%		99.3%	
		Precision	36.7%		88.0%	
		TP Rate (Recall)	62.1%		75.9%	
		F1-score	46.2%		81.5%	
Shopping	8%	Confusion Matrix	10	24	22	14
			18	398	6	408
		Accuracy	90.7%		95.6%	
		TN Rate	94.3%		96.7%	
		Precision	29.4%		61.1%	
		TP Rate (Recall)	35.7%		78.6%	
		F1-score	32.3%		68.8%	
Games	4%	Confusion Matrix	6	14	20	4
			20	410	6	420
		Accuracy	92.4%		97.8%	
		TN Rate	96.7%		99.1%	
		Precision	30.0%		83.3%	
		TP Rate (Recall)	23.1%		76.9%	
		F1-score	26.1%		80.0%	
News	10%	Confusion Matrix	21	23	31	16
			22	384	12	391
		Accuracy	90.0%		93.8%	
		TN Rate	94.3%		96.1%	
		Precision	47.7%		66.0%	
		TP Rate (Recall)	48.8%		72.1%	
		F1-score	48.3%		68.9%	
Search	2%	Confusion Matrix	6	2	6	6
			4	438	4	434
		Accuracy	98.7%		97.8%	
		TN Rate	99.5%		98.6%	
		Precision	75.0%		50.0%	
		TP Rate (Recall)	60.0%		60.0%	
		F1-score	66.7%		54.5%	

* Results are based on websites' design features and meta-keywords. The design matrix presents the following counts, starting with the upper left entry clockwise direction: true positive (TP), false positive (FP), true negative (TN) and false-negative (FN).

egories: "Shopping" (94.3%); "Games" (96.7%); "News" (94.3%); and "Search" (99.5%).

The "search" class was the smallest, representing 2% of the websites, and yet it reached a precision of 75% based solely on design features. When keywords were added, the algorithm was defocused, and the precision decreased to 50% because search engines contain many keywords that may represent many other categories. The highest improvement in the F1-score occurred for the "Games" category (26.1% to 80%), although this is a small minority class that contains only 4% of the websites in the dataset.

These results indicate that relying only on websites' design features can provide satisfactory results in many cases, primarily with respect to true negative rates. Using keywords can thus improve results, as shown in the last column of Table 2. In particular, the average TN rate of "Crack" websites was improved by 6.7%, while the average TN rate was improved by only 2%–3% for all other categories. The average TP rates, conversely, were improved significantly when keywords were considered, while the average total accuracy was improved by 1%–7% in most cases. Respectively, the recall, precision and harmonic mean, which is known as the F1-score and is typically applied to unbalanced datasets, describe similar phenomena: when relying on design features and keywords, the obtained measures were relatively high, with the exception of the "News" category, where the variety of keywords actually decreases the measure performance.

The F1-score of "Crack" websites, based on both design features and keywords, was the highest (81.5%) among all classes. Similarly, the other categories achieved the following F1-scores: "Games" (80%); "News" (68.9%); "Shopping" (68.8%); and "Search" (54.5%). Thus, it is typically true that classification performance often increases with sample size; however, these observations were also relevant for minority classes, such as "Shopping" (8%), "Games" (4%) and "News" (10%).

When adding the meta-keywords to the design features, the greatest improvement in the average classification accuracy occurred with "Crack" websites (7.1%). A possible reason for this result is that "Crack" websites discuss different top-

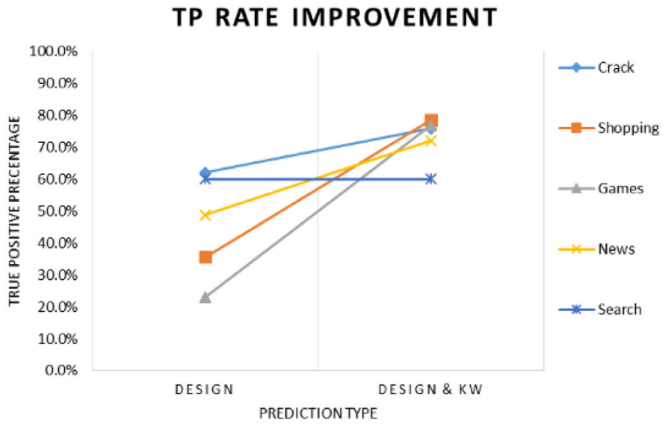

Fig. 4 – True Positive by Classification Type.

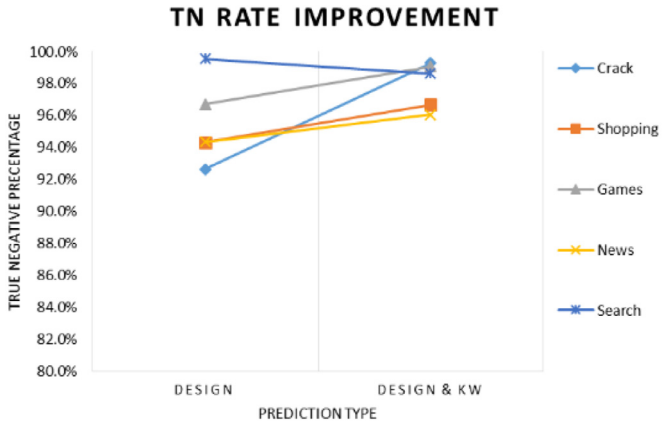

Fig. 5 – True Negative by Classification Type.

ics but use similar keywords to indicate their functionality to users, a fact the proposed learning algorithm exploits that.

Following the manual verification, we found that using keywords had significantly improved classification. However, the improvement achieved by learning keywords cannot be guaranteed in the general case. For example, 'black-hat' search engine optimization (SEO) procedures manipulate websites' keywords to attract incoming links from legitimate sites and even steal their content (Motoyama et al., 2011). Figs. 4 and 5 describe the TP and TN improvement in percentages. Fig. 6 describes the overall model accuracy. The only exception occurs with search websites in which design features do not improve classification accuracy. One possible reason for this result is that these websites often do not contain many design objects. The "Search" category had the lowest number of instances (websites) in the dataset.

4.1.1. . *Problem formulation*

To further analyze the proposed classifiers' performance, we calculated the receiver operating characteristic (ROC) curves of "Crack" websites. Fig. 7 shows the ROC curve for the classification by design features only. The area under this ROC curve for "Crack" and "Not crack" was 0.78. Fig. 8 shows the ROC curve for the classification by design features alone as well as by both design features and meta-keywords. The area under this ROC curve for "Crack" and "Not crack" was 0.78. This relatively marginal difference again shows the value of the design features for the considered classification task. The TP rate increased when comparing classifications based on design features and categories when both design features and meta-keywords were considered.

To evaluate the increase, we have performed a paired samples *t*-test (one-tailed). Using Weka software, the confusion matrix elements (TP and TN) were obtained using various folds over 100 repetitions, as shown in Table 3. We used the same J48 algorithm parameters that were used in earlier numerical studies for consistency. The following results were obtained for the paired tests. The p-value for the paired *t*-test between the TP rates was 1.11E-49; thus, the average TP rates for the classifications based on both the design features and the meta-keywords were higher than the TP rates for the classifications based on the meta-keywords alone. The p-value of the paired *t*-test between the TN rate was 6.41E-41. The average TN rates for the classifications based on both the design features and the meta-keywords were higher than the TN rates for the classifications based on the meta-keywords alone. We also used a *t*-test to evaluate the significance of the improvement in the overall accuracy when using the two sets of features. This test resulted in a significance level of 7.51E-

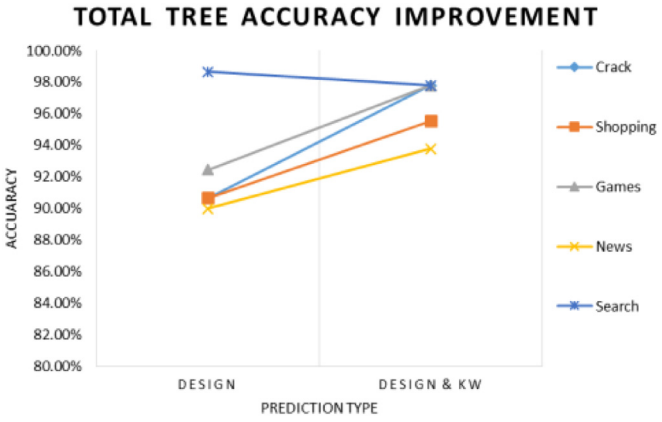

Fig. 6 – Total Tree Accuracy.

Table 3 – Illustration of the paired samples of various confusion matrix elements.

Rep	Fold	Training	Testing	TP	FP	TN	FN
1	1	405	45	43	0	2	0
1	2	405	45	39	2	1	3
1	3	405	45	42	0	3	0
1	4	405	45	42	1	2	0
1	5	405	45	42	2	1	0

*Results are based on over 100 repetitions that were used to obtain the paired t-tests' *p*-values.

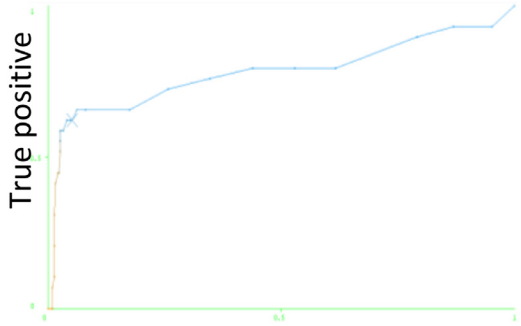

Fig. 7 – ROC curve for prediction by design features alone.

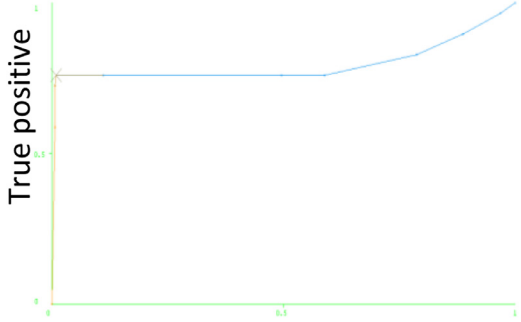

Fig. 8 – ROC curve for prediction by both design features and metakeywords.

04; thus, the average accuracy for the classifications based on both the design features and the meta-keywords was higher than the average accuracy for the classifications based on the meta-keywords alone. Thus, again, using the design features leads to a statistically significant improvement. To describe the obtained classification models in more detail, Fig. 9 shows a J48 decision tree for the classification of crack websites that rely only on design features. Fig. 10 shows a J48 decision tree for the classification of crack websites that rely on both design features and meta-keywords. As shown in these decision trees, there are different routes for classifications based on both design features and meta-keywords compared to those based only on design features.

However, there were also similar routes in both decision trees. These routes are often based on certain types of normalization (e.g., using the standard deviation of the "number of elements in a website"). Using a J48 tree to generate a list of routes, one can simplify and summarize the major decision questions in the following route: "If the 'number of white blank areas' (i.e., no text or image) decreases, and the 'text total rate for elements from type "A" (link)' increases, the probability for the website to be "Crack" increases."

Accordingly, we checked if the *color*-specific features are placed in the first two levels of depth in the tree (i.e., having the highest information gain in the first two iterations of building the tree). In 7 out of 10 tree models overall categories, we found that color features were indeed selected on the tree's first two levels on the first two levels of the tree. Two out of the three classification trees that did not contain color features in the first two levels were trees that were generated to classify the "Search" category, which had the lowest number of instances (websites) in the dataset. Thus, these results show

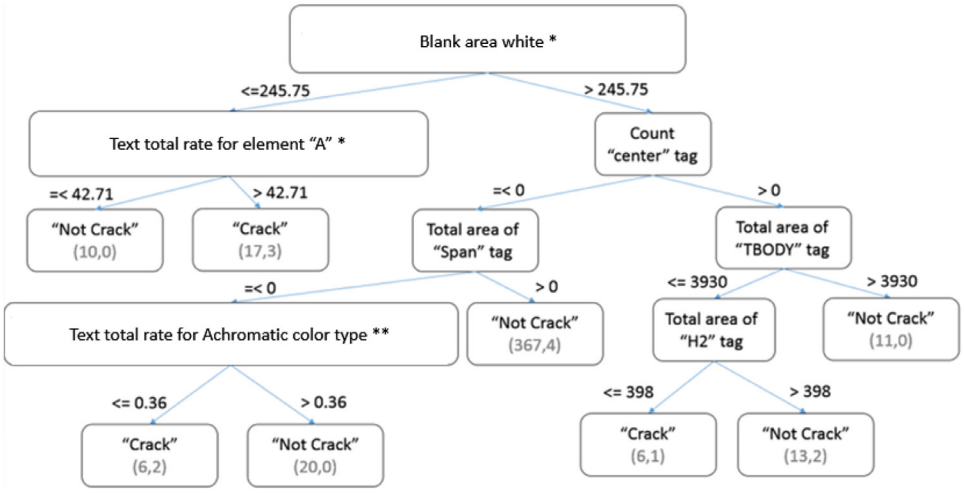

Fig. 9 – J48 decision tree for the classification of crack websites.
*** Normalized to the sum of the elements in the webpage**
**** Normalized to the text total rate for all elements in the webpage.**

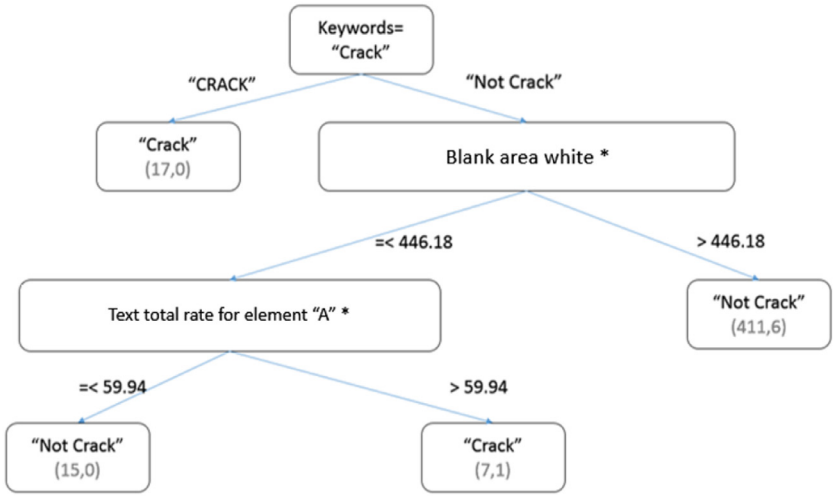

Fig. 10 – J48 decision tree for the classification of crack websites relying on both design features and metakeywords.

that color features play important roles when using a J48 decision tree to classify website categories.

4.2. *Results of Experiment 2*

The first experiment, which used manually tagged data, provided a proof of concept for the idea of website identification using website design features. Based on that experiment's promising ve results, we developed an automated "next gen algorithm" to collect and obtain a larger-scale dataset with a wider feature set.

Using an unbalanced data set of 15,432 URLs containing 2522 features for each URL, we developed and trained five popular machine learning models: logistic regression, KNN, ANN, AdaBoost, and CART decision tree. More details about these models are discussed below. We intentionally chose classic models as described, because we aim to demonstrate the importance of design features without providing major importance to the model by itself. Principal component analysis (PCA) was used, and results showed a linear combination of 700 features (out of 2522) was required to explain nearly all variance in the dataset. The feature importance level was analyzed using an ANOVA F-value score between the labels and features. The first decile of important features was found to consist of features from different feature groups, as shown in Fig. 11. For all considered models, 5-fold cross-validation was performed.

The logistic regression model attained a significantly low FP rate; however, it managed to detect only 10.5% of the malicious websites on average with an average recall of 10.5%, an average precision of 88.5%, and an average F1 score of 18.9%.

Surprisingly, the ANN model yielded unsatisfactory results. We trained and tested various networks with different sizes using a rectified linear unit function activation function and a stochastic gradient descent solver. A four-layer network containing 64 neurons in each layer obtained an average recall of

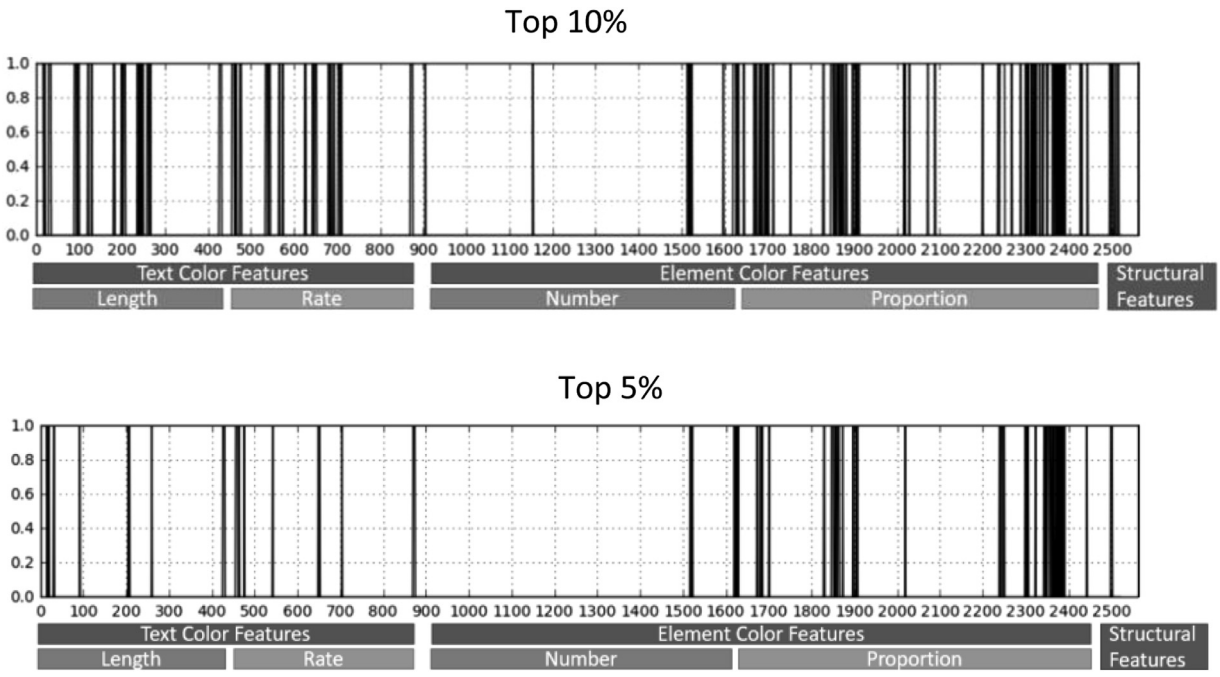

Fig. 11 – ANOVA F-value score between labels and features, top 10% and top 5%.

36.4%, an average precision of 70.1%, and an average F1 score of 47.9%. For larger networks, the classifier results worsened.

An adaptive boosting model based on decision tree classifiers with a maximum depth of three detected the majority of the malicious sites while maintaining a relatively low false-positive rate. Results showed an average recall of 53.5%, an average precision of 93.5%, and an average F1 score of 68.1%.

The KNN model provided relatively good results. Implemented with uniform weights, a Euclidean distance a brute-force start algorithm, and five neighbors, the model detected the majority of the malicious websites with an average recall of 67.9%, an average precision of 88.1%, and an average F1 score of 76.7%.

A simple CART decision tree classifier was found to outperform all other models in terms of the F1 score. When trained with a maximum depth of ten levels, the CART decision tree obtained an F1 score of 78.4%, identified 69.2% of the malicious sites on average, and maintained a precision of 90.2%.

When the tree's maximum depth was reduced, the ability to detect malicious sites did not decrease significantly, while model precision improved.

For a decision tree with a maximum depth of eight levels, the classifier achieved an F1 score of 77%, less than 0.155% false positives, a recall of 65.6%, and a precision of 93.3%. A comparison of the various learning models with their parameters is shown in Table 4. A possible explanation for the higher F1 score of the CART decision tree is its ability to manage various types of variables and identify the most significant features while eliminating insignificant features (Singh et al. 2014). This ability was found to be a significant advantage when considering the wide and sparse design feature set analyzed in this study that contained different categorical, nominal and numerical features, as shown in Table 1.

We have also verified the results on a balanced dataset. For this task we selected the 510 malicious websites and, correspondingly, we randomly selected 510 legit websites. 40 PCA components were used (out of 2500 features) to avoid overfitting. As expected, the accuracy of the model has decreased (84.6%), however the recall and F1-score achieved better results: Recall 76%, F1-score 83.2% and Precision 91.7%.

We compared the results of the proposed algorithm to those of the kAYO mechanism, as shown in Fig. 12. The kAYO mechanism is an analysis technique based on static features of mobile webpages derived from their URL, HTML and JavaScript content that is used to detect malicious mobile webpages in any language (Amrutkar et al., 2016). As shown in Fig. 12, for TPR values lower than 70%, the proposed algorithm results in significantly lower FPR values than kAYO

5. Discussion

In this study, we classified websites by learning their design features, which are often ignored in the literature. We were surprised to see that design features alone can provide a lot of information about website categories. Thus, the same types of websites have similar design features, regardless of their geography, language, and keywords. These results support an earlier hypothesis about the distinct visual style of crack and malicious websites and extend this observation to other website categories.

In Experiment 1, which was based on the identification of websites after manual tagging, we showed that websites' design features can be used to enhance the identification of the website category in general and of crack websites in particular. Using only design features for website category classification, we achieved a 90% accuracy. Adding design features to meta-

Table 4 – Comparison of various machine learning models in Experiment 2. "M" and "L" represent 'Malware' and 'Legit' websites, respectively.

	Logistic regression		KNN		ANN		Adaptive boosting		Decision Tree	
	Predicted M	Predicted L	Predicted M	Predicted L	Predicted M	Predicted L	Predicted M	Predicted L	Predicted M	Predicted L
Actual M	9.6	92.4	71.4	30.6	37.2	64.8	54.6	47.4	66.8	35.2
Actual L	1.2	2983.2	10.4	2974	32.8	2951.6	3.8	2980.6	4.8	2979.6
Recall	11%		69%		36%		54%		66%	
Precision	88%		87%		70%		94%		93%	
F1 Score	19%		77%		48%		68%		77%	
Parameters	Default		Distance Metric: Euclidean, K=5, Weighting Method: Uniform		Activation Funcion: ReLU, Solver: SGD, Hidden Layers: 4, Neurons per Layer: 64		Weak Classifier: CART DT, Maximal Depth: 3		Maximal Depth: 10, Splitting Criteria: Gini	

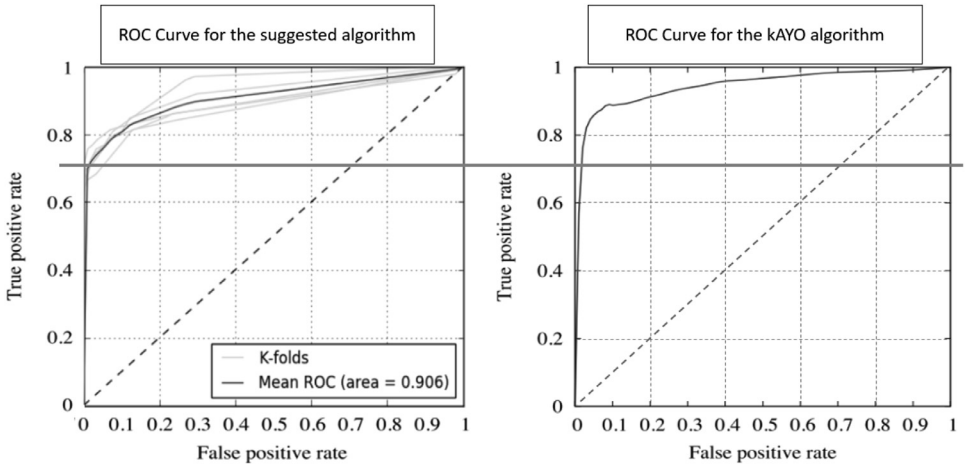

Fig. 12 – Comparison of the proposed algorithm and the kAYO ROC curves. For a TP rate range of 0%–70%, the proposed scheme outperforms kAYO in terms of the false positive (FP) rate (Amrutkar et al., 2016).

keywords increased the accuracy significantly to 97% or higher for most categories, resulting in a true positive rate increase across all tested website categories. In particular, all decision trees reached an average accuracy of 96.5%. As discussed earlier, the true positive rate plays an important role when crack websites must be identified. Therefore, we assume that the user will prefer to have marginally lower accuracy instead of avoiding crack websites that may harm his system.

When combining design and metadata for the "Crack" website category, both the TP rate and overall accuracy increased, which may be due to the compromised metadata that are used in crack websites. Thus, using design features to identify crack websites might be more effective because design features do not rely on text words or keywords that might be used to deceive the user. This result indicates that using design features along with a set of meta-keywords for classification yields good website classification.

Because using a sample size of hundreds of websites is not sufficient to conclude whether design features can be used efficiently to classify website categories, a robust automated algorithm was developed to address larger dataset sizes with a larger set of features, as shown in Experiment 2. In this experiment, we expanded the dataset to more than 15,000 websites and increased the number of design features to 2522. We used the ANOVA F-value to identify the most impactful features to avoid overfitting. We trained five popular machine learning classifiers (logistic regression, KNN, ANN, AdaBoost, and a CART decision tree) and found that the CART decision tree classifier outperforms all other models in terms of F1 score. In particular, the CART classifier achieved an F1 score of 77%, resulting in less than 0.155% false positives, a recall of 65.6% and a precision of 93.3%. This result implies that the "noise" in design features is low for website-classification tasks.

One of the proposed approach's possible applications may be relevant for large organizations, such as financial or government organizations. For example, the proposed approach can be integrated into the web browsers of employees' computers or in the organization's firewall as a distinctive layer to block suspicious websites. Based on the results of Experiment 2, only a small fraction of legitimate websites will be erroneously marked as suspicious by the proposed algorithm.

One can challenge the proposed approach by arguing that Crack websites can use visual designs similar to those used by legitimate websites. Note, however, that in this study, the word "design" represents a wide spectrum of objects, standards and specifications that malware developers cannot easily access, see or imitate. A "design" consists of all HTML code and hierarchies, JavaScript, CSS, color tables, styles, font types, objects,

etc. (; Duckett, 2011; Nixon, 2012; Chen and Ryu, 2011). The average developer cannot easily imitate this depth of a structure exactly, unlike a list of keywords that can be easily modified. Therefore, as noted above, we do not suggest using the proposed approach to develop a new search engine; rather, we suggest combining it with existing methods as an additional protective layer to manage more potential threats.

The proposed method has several advantages over other methods that can be easily deceived by the frequent changes in Crack Websites, such as those developed by Ma et al. (2011) and Zhang et al. (2014). First, the proposed approach is designed to be used independently of language, URL, geographical location, or website type. These characteristics are strong advantages because the Internet is rapidly changing, and this approach provides users with the ability to analyze information on a wide range of websites in near-real time. Second, the proposed method does not consider outgoing or incoming links and can therefore be used without requiring any information on the structure of the category to which the website belongs.

Several modifications and extensions could be implemented in future research to improve the performance and accuracy of the proposed application. One such direction is to increase the dataset to a scale of 500K-1M websites. Such an increase will require a longer learning process and more computing power but would increase the sample size and most likely improve classification accuracy. A larger dataset could support more learning features to target higher identification accuracy, including URL prefixes, countries of origin, user metrics, and display features for different devices (PC, laptop, tablet, smartphone, etc.). These features that are not necessarily associated with website design per se can result in better accuracy. Second, the proposed application could be expanded to scan and analyze images, possibly using deep learning classifiers. The experiments in this study are primarily based on scanning and analyzing sites' HTML code and extensions, such as JS and DOM, which do not include the websites' images. Analyzing images could provide an additional layer of significant features for better website segmentation, classification and prediction. For example, website images could be used to easily identify commercial products, celebrities and, similarly, illegal and crack content and porn-related objects. Third, the proposed approach could be modified to better manage timeout errors, which were found to occur often during scanning. Many scanned websites were interrupted due to timeout errors, which were set to 60 seconds per scan. Timeout errors occur due to a variety of reasons, such as bandwidth saturation; website security and blocking mechanisms; and incorrect or non-standard webpage code. To reduce the number of timeout errors, the scanning algorithm could explore and identify various errors in real time, differentiate them and then associate a specific subprocedure to manage each type of error. Fourth, the suggested algorithm was executed in an environment with a relatively low download rate with only 8 available cores, while the proposed algorithm can support a simultaneous execution by many more cores. As a result, by using a faster download rate and higher computing power the total run time can be reduced significantly to support a large-scale data collection as suggested. More specifically, by using a faster download rate, we expect the mean scanning time to be less than 2 seconds per website per core. As an example, by using a faster download rate and 24 cores, the overall expected time for scanning 100,000 websites is expected to be approximately 2.3 hours. Finally, it is important to indicate that the learning phase that can be carried offline and in parallel to the online website scanning and classification stage. These modifications and extensions could increase the prediction accuracy and performance of the proposed approach, and thus should be investigated in future research.

Most website classification tools that are available today are based on website content and not their design features. We believe that using the proposed method to detect website (sub)categories based on their design features can have various applications, such as blocking specific websites and focusing a search on a specific website category. This option is suitable for large organizations that aim to reduce the access of their employees to potentially harmful websites, such as malware, cracks, gambling and porn-related sites. Another potential use case of the proposed method is an automated comparison between different websites from the same category. This comparison may identify which design features represent more 'successful' websites, such as websites with large numbers of users that are best sellers, fast-growing and well-branded. Design features might also play an important role in marketing applications. Applying data-mining methods to these features may show patterns that enhance marketing exposure and sales activities.

6. Conclusions

In this paper, we presented a new method for website categorization based on thousands of visual and nonvisual design features, and used it to identify website categories, specifically malware- and Crack websites. We showed that design features differ significantly between website categories and can be used specifically to identify malware and crack.

In this study, the word "design" represents a wide range of features and objects, and is not necessarily limited to visual elements on a webpage. In certain cases, these design features can even show developers' behavioral patterns and trends. Design features are not always identifiable by the human eye due to the large volume of information across a large combination of features that is exposed at once. There are also hidden elements in websites that cannot be seen by users but that can be found in the webpage's code. Visual design elements, such as differences in tones among colors (dark, light, saturated, muted or achromatic), are highly variable and are difficult to categorize based solely on the human eye. However, design features are critical in conveying (underlying) messages, such as the quality or the price of the products on ecommerce websites, to users. This fact is the reason why categorizing design features can be important for website designers in various applications, such as identifying specific website topics, analyzing trends, finding anomalies, designing ad-removal algorithms, and automating and optimizing website design.

In this study, we were able to identify malicious websites at relatively high true positive rates and negligible false positive rates. These results imply that the false alarm rates for users and organizations that may apply this method are reasonably

low. Results show that a classification layer that relies on design features can serve as an effective alternative to text mining that is often time consuming and computationally intensive. We do not claim that the proposed approach is always superior to other identification methods; rather, we claim that it can serve as an efficient extension of existing methods, specifically when the application's running time and accuracy are critical. This study provides a proof of concept to identify useful information embedded in website design features. The full automation and extension of the proposed approach, in which design, images and text features are mapped into feature vectors with tens of thousands of dimensions and are then categorized by algorithms such as deep learning, latent Dirichlet allocation (LDA) or other topic modeling algorithms, should be a subject in future research.

Research Data

The python code and the database we generated and used, can be found in this link https://data.mendeley.com/datasets/xvmcmngkjw/1

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.

CRediT authorship contribution statement

Doron Cohen: Conceptualization, Methodology, Software, Formal analysis, Validation, Visualization, Writing - original draft. **Or Naim:** Investigation, Software, Resources, Data curation. **Eran Toch:** Formal analysis, Methodology, Writing - review & editing. **Irad Ben-Gal:** Conceptualization, Methodology, Validation, Writing - review & editing, Project administration, Supervision.

Acknowlegements

This paper was partially supported by the Koret Foundation Grant for Digital Living 2030, and by the ICRC Grant for cyber security.

REFERENCES

Aburrous M, Hossain MA, Dahal K, Thabtah F. Predicting phishing websites using classification mining techniques with experimental case studies. In: Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations (ITNG '10). Washington, DC, USA: IEEE Computer Society; 2010. p. 176–81.

Al-Saleh MI, Abuhjeela FM, Al-Sharif ZA. Investigating the detection capabilities of antiviruses under concurrent attacks. Int. J. Inf. Secur. 2015;14(4):387–96.

Bogdanov D, Niitsoo M, Toft T, Willemson J. High-performance secure multi-party computation for data mining applications. Int. J. Inf. Secur. 2012;11(6):403–18.

Bonnardel N, Piolat A, Bigot Lle. The impact of colour on Website appeal and users' cognitive processes. Displays 2011;32(2):69–80.

Krebs B. Software Cracks: a Great Way to Infect Your PC; 2011 http://krebsonsecurity.com/2011/06/software-cracks-a-great-way-to-infect-your-pc/.

Carlini, et al. Adversarial examples are not easily detected: Bypassing ten detection methods. Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (pp. 3-14). 2017.

Cebi S. Determining importance degrees of website design parameters based on interactions and types of websites. Decis. Support Syst. 2013;54(2):1030–43.

Cyr D, Trevor-Smith H. Localization of Web design: an empirical comparison of German, Japanese, and United States Web site characteristics. J. Am. Soc. Inf. Sci. Technol. 2004;55(13):1199–208.

Cyr D, Head M, Larios H. Colour appeal in website design within and across cultures: a multi-method evaluation. Int. J. Hum. 2010;68:1–2 1-21.

Cyr D. Modeling Web Site Design Across Cultures: relationships to Trust, Satisfaction, and E-Loyalty. J. Manage. Inf. Syst. 2008;24(4):47–72.

Cai D, Yu S, Wen Ji-R, Ma W-Y. Extracting content structure for web pages based on visual representation. In: Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications (APWeb'03). Berlin, Heidelberg: Springer-Verlag; 2003. p. 406–17.

Deloitte. TMT Global Security Study; 2013.

Chen T-C, Stepan T, Dick S, Miller J. An anti-phishing system employing diffused information. ACM Trans. Inf. Syst. Secur. 2014;16(4).

Egele M, Scholte T, Kirda E, Kruegel C. A survey on automated dynamic malware-analysis techniques and tools. ACM Comput. Surv. 2008;44(2).

Golander GK, et al. Trends in website design.. AIS Transactions on Human-Computer Interaction, 4(3), 169-189. 2012.

Hadžiosmanović D, Bolzoni D, Hartel PH. A log mining approach for process monitoring in SCADA. Int. J. Inf. Secur. 2012;11(4):231–51.

He M, Horng S-J, Fan P, Khan MK, Run R-S, Lai J-L, Chen R-J, Sutanto A. An efficient phishing webpage detector. Expert Syst. Appl. 2011;38(10):12018–27.

Heiderich M, Frosch T, Holz T. IceShield: detection and mitigation of malicious websites with a frozen DOM. In: Proceedings of the 14th international conference on Recent Advances in Intrusion Detection (RAID'11). Berlin, Heidelberg: Springer-Verlag; 2011. p. 281–300.

Kammerstetter M, Platzer C, Wondracek G. Vanity, cracks and malware: insights into the anti-copy protection ecosystem. In: *Proceedings of the 2012 ACM conference on Computer and communications security* (CCS '12). New York, NY, USA: ACM; 2012. p. 809–20.

Knight S, Buffett S, Hung PCK. The International Journal of Information Security Special Issue on privacy, security and trust technologies and E-business services: guest Editors' Introduction. Int. J. Inf. Secur. 2007;6(5):285–6.

Lakshmi VS, Vijaya MS. Efficient prediction of phishing websites using supervised learning algorithms. Procedia Eng. 2012;30:798–805.

Ma J, Saul LK, Savage S, Voelker GM. Learning to detect malicious URLs. ACM Trans. Intell. Syst. Technol. 2011;2(3).

Mesbah A, van Deursen A, Lenselink S. Crawling ajax-based web applications through dynamic analysis of user interface state changes. ACM Trans. Web 2012;6(1).

Motoyama M, McCoy D, Levchenko K, Savage S, Voelker GM. An analysis of underground forums. In: *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference* (IMC '11). New York, NY, USA: ACM; 2011. p. 71–80.
Pavlou PA, Liang H, Xue Y. In: Understanding and Mitigating Uncertainty in Online Exchange Relationships: A Principal-Agent Perspective, 31. MIS Q; 2007. p. 105–36.
Pelet JÉ, Papadopoulou P. The effect of e-commerce websites' colors on customer trust. Int. J. E-Bus. Res. 2011;7(3):1–18 July 2011.
Pfleeger SL, Sasse AM, Furnham A. From weakest link to security hero: transforming staff security behavior. J. Homel. Secur. Emerg. Manag. 2014;11(4):489–510 2014.
Shahabi C, Zarkesh AM, Adibi J, Shah V. Knowledge discovery from users Web-page navigation. Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications (RIDE '97), 1997.
Spiliopoulou M, Faulstich LC. In: International Workshop on the Web and Databases. Wum: a web utilization miner; 1998. Valencia, Spain.
Wells JD, Valacich JS, Hess TJ. What signal are you sending? how website quality influences perceptions of product quality and purchase intentions. MIS Q 2011;35(2):373–96.
Wu Ou, Hu W, Shi L. Measuring the Visual Complexities of Web Pages. ACM Trans 2013;Web 7(1).
Wu Ou, Zuo H, Hu W, Li B. Multimodal web aesthetics assessment based on structural SVM and multitask fusion learning. IEEE Trans. Multimedia 2016;18(6):1062–76.
Yue C, Wang H. A measurement study of insecure javascript practices on the web. ACM Trans. Web 2013;7(2).
Zhang X, Wang Y, Mou N, Liang W. Propagating Both Trust and Distrust with Target Differentiation for Combating Link-Based Web Spam. ACM Trans. Web 2014;8(3).
Zhuge J, Holz T, Song C, Guo J, Han X, Zou W. Studying malicious websites and the underground economy on the Chinese web. In: Managing Information Risk and the Economics of Security. Springer US; 2009. p. 225–44.
Amrutkar C, Kim YS, Traynor P. Detecting mobile malicious webpages in real time. IEEE Trans. Mob. Comput. 2017;16(8):2184–97.
Shahegh P, Dietz T, Cukier M, Algaith A, Brozik A, Gashi I. AntiVirus and Malware Analysis Tool; 2017.
Moghimi M, Varjani AY. New rule-based phishing detection method. Expert Syst. Appl. 2016;53:231–42.
Lewis RJ. An introduction to classification and regression (CART) analysis. Presented at the 2000 Annual Meeting of the Society of Academic Emergency Medicine San Francisco, California; 2000.
Bernardini A. Extending Domain Name Monitoring. Identifying Potentially Malicious Domains Using Hash Signatures of DOM Elements. ITASEC; 2018.
Sarhan Al, A J,R, Sharieh A. Website Phishing Detection Using Dom-Tree Structure and Cant-MinerPB Algorithm. American Journal of Computer Science and Information Engineering 2017;4(4):38–42.
Samtani S, Chinn R, Chen H. Exploring hacker assets in underground forums. In: 2015 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE; 2015. p. 31–6.
Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and Regression Trees. CRC press; 1984.
Steinberg D, Colla P. CART: classification and regression trees. In: The Top Ten Algorithms in Data Mining, 9; 2009. p. 179.
Amrutkar C, Kim YS, Traynor P. Detecting mobile malicious webpages in real time. IEEE Trans. Mob. Comput. 2016;16(8):2184–97.
Liu X, Lin Y, Li H, Zhang J. A novel method for malware detection on ML-based visualization technique. Comput. Secur. 2020;89.
Kim S, Kim J, Kang BB. Malicious URL protection based on attackers' habitual behavioral analysis. Comput. Secur. 2018;77:790–806.
Fang Y, Huang C, Su Y, Qiu Y. Detecting malicious JavaScript code based on semantic analysis. Comput. Secur. 2020;93.
Chiba D, Akiyama M, Yagi T, Hato K, Mori T, Goto S. DomainChroma: building actionable threat intelligence from malicious domain names. Comput. Secur. 2018;77:138–61.
Cimino MG, De Francesco N, Mercaldo F, Santone A, Vaglini G. Model checking for malicious family detection and phylogenetic analysis in mobile environment. Comput. Secur. 2020;90.
Singh S, Gupta P. Comparative study ID3, cart and C4. 5 decision tree algorithm: a survey. Int. J. Adv. Inf. Sci. Technol.) 2014;27(27):97–103.
Duckett J. HTML & CSS: Design and Build Websites (Vol. 15). Indianapolis, IN: Wiley; 2011.
Nixon R. Learning PHP, MySQL, JavaScript, and CSS: A step-By-Step Guide to Creating Dynamic Websites. O'Reilly Media, Inc.; 2012.
Chen M, Ryu YU. Facilitating effective user navigation through website structure improvement. IEEE Trans. Knowl. Data Eng. 2011;25(3):571–88.
Salzberg SL. C4. 5: Programs for Machine Learning By J. Ross Quinlan. Morgan Kaufmann Publishers, Inc.; 1994. 1993.
Pouyanfar S, Tao Y, Mohan A, Tian H, Kaseb AS, Gauen K, Shyu ML. Dynamic sampling in convolutional neural networks for imbalanced data classification. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). IEEE; 2018. p. 112–17.
Al-Azani S, El-Alfy ESM. Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text. Procedia Comput. Sci. 2017;109:359–66.

Irad Ben-Gal, Professor & Head, **L**aboratory of **AI** **B**usiness and **D**ata **A**nalytics (LAMBDA), Tel Aviv University.

Prof. Ben-Gal is a world-renowned expert in machine learning, data science and predictive analytics with more than 25 years of experience in the field, including close R&D collaborations with companies such as Oracle, Intel, GM, AT&T, Applied Materials and Nokia.

Prof. Ben-Gal wrote four books, published more than 100 scientific papers and patents, supervised dozens of graduate students and received numerous awards for his work. He held a visiting professor position at Stanford University, teaching graduate courses in analytics and co-heading the TAU/Stanford "Digital Living 2030" research initiative.