Extending domain name monitoring. Identifying potentially malicious domains using hash signatures of DOM elements

Andrea Bernardini[1]

Fondazione Ugo Bordoni, Rome, Italy
abernardini@fub.it

Abstract

The usage of disposable domains for malicious activities is an increasing trend and a fertile ground for fraudsters actions from phishing to fake goods selling. Those domains are registered daily, go online within a few hours and last for a very short time. Due to the existence of automated developing tools for content creation and efficient site management, a single monicker entity may be in charge of large networks of thousands of domains. Such a grade of automation requires the usage of easy transmutable websites templates and repetitive code snippets. In order to deploy a more resilient anti-fraudster strategy, we present a framework which combines web pages scraping procedures, simhash fingerprint based near duplicate document detection and agglomerative clustering. The objective is twofold: firstly to identify common and repetitive structural patterns in potential illicit websites; secondly to monitor new emerging technical trends in short period time frames. The framework has been tested on a corpus of newly registered .com domains for a period of three weeks. The results consistently confirm the existence of recurring technical schemes. We showed that, by using document fingerprinting, it considerably increases the overall comprehension of strategies used in complex suspicious domains networks and it may be of support for a new concept of domain protection.

Contents

1 Context

Thousands of domains are registered and dropped every day across all top-level domains. The second quarter of 2017 closed with approximately 332 million of registration and it is an increasing number quarter by quarter [5]. The registered domains are completely new or could be

expired/deleted recently. Malicious entities use to register domains daily for creating complex networks with the objective of generating spam, promoting and selling fake goods or induce users in phishing [12] [13]. Often a single entity (person or organization) is in charge of a complex network of interconnected domains, hosted in various countries. The life cycle of such domains may range from few days to years depending on the responsiveness of potential victims, of the targeted brands and controlling authorities. It may occur that those domains are finally inserted in a DNS Blacklists [6], be mentioned in sites contrasting scam and phishing [7] or be removed by search engine indexing system. The loss of a domain due DNS block or removal by search engine has negligible effects. A new domain can be quickly registered, and a new site can be deployed. Indeed, all the procedures for managing contents can be executed by toolkits [3] able to clone templates and randomize contents producing web pages similar or identical. Recent studies [10] [23] focus on asserting the trustworthiness of websites by the analysis of search engine results on the base of a range of features extracted from pages and integrated eventually with external resources as domain registration data and other metrics. In this work, we focus on the detection of potential illicit websites by identification of anomalous density of structural similarities on newly registered websites. The usage of management toolkits and templates leave often traces and technical marks. Based on this observation we tackle the problem of identification of potentially malicious sites to a well-known research topic as the nearly duplicate document detection applied to web pages document structure.
The rest of the paper is structured as follows:

2. Introduction reviews the related works and summarize some main concepts;
3. Proposed approach describes the proposed framework and its three main components;
4. Experimental setup describes the settings and the data corpus;
5. Results and observations
6. Conclusions

2 Introduction

Duplicated and mirrored web pages are seen in plenty in the World Wide Web, and they have been object of studies for as long as the web exists as its overlap with many topics as plagiarism detection, spam detection and even crawlers optimization. The field of near-duplicate document (NDD) detection focuses on the individuation of almost identical documents which differ in small portions. In the context of web pages the differences could be in the text, in the images, as well as in the structure. More in detail a web page may be decomposed in different layers of features: semantic, structural, and visual.

- The semantic layer of a page is the information and the topics it expresses;
- The structural layer of a page is the underlying skeleton of a page;
- The visual layer of a page represents the visual information conveyed by the page.

In general, all the approaches of NDD vary on the strategy for the document features selection, for the compression of the features in a signature, for the comparison the documents with a similarity measures and for the corpus under consideration. The growing size of datasets and the high-dimensional space of documents leaded to the necessity of investigating dimensionally reduction techniques. Broder [9] introduced the algorithm MinHash and a technique called shingling used for the estimation of document resemblance based on the overlapping of a subset of adjacent words sequences (called shingles). The MinHash is the first of a more general framework of algorithms, the Locality Sensitive Hashing [25] devised for solving near duplicate and similarity problems for web pages and images [17] [24]. Broder studies have been reviewed to investigate the evolution of individual pages and subsequently related clusters of near-duplicates pages. It emerged that two documents that are near-duplicates of one another are very likely to still be near-duplicates after months [14]. MinHash was at first applied to a set of 30 million web pages, but the dimension of dataset kept growing, so optimization techniques were investigated. Charikar [11] proposed a fingerprinting techniques on documents for mapping high dimensional vectors to small-sized fingerprints. The process is organized in two steps. Firstly, for each document it is calculated a representative hash and then near duplicates are detected by identifying documents that have similar hashes. Manku [20] demonstrated the goodness of Charikar's technique and proposed an algorithm for identification of fingerprints which differ from a chosen one for a maximum of k positions. Other studies focused on the performance respect to the overall complexity of the analysis [22]. Narayana proposed an approach working on keyword extraction and the correlation between two documents it is given if the similarity score is greater than a threshold [21]. Other mixed approaches have been proposed. [19] proposed a mixed approach of keywords extraction and fingerprinting. Most of those studies concentrate on a vector based on semantic features, the text contained in the page. All other page components as the HTML markup tags are substituted by whitespaces. In the study [15] the structural similarity of documents based on the Tree Edit Distance between Document object model (DOM) [1] trees is investigated. Recent studies explore the combination of LSH techniques with clustering process [18] to reduce computational costs working on only semantic layer or a mixed semantic and structural layer, which joins content and other data as title, description, keywords, and tags [24]

3 Proposed approach

In our proposed approach we focus on the analysis of structural layer features of web pages, extracting from an HTML page the corresponding DOM tree and transforming it in a stripped version, containing only HTML tags, by the removal of all textual contents. The sequence of tags is then converted, using the simhash algorithm, in hashed fingerprints. Due to the properties of simhash algorithm similar documents have similar fingerprints so it is possible to group fingerprints in hash buckets and then store them in hash tables. The similarity is calculated by using the Hamming [16] distance since we converted the document features space to a hash f-bit space. Subsequently, we randomly select from each bucket candidates for nearly duplicate selection, and we construct, by the resulting aggregations, clusters of similar documents. The system architecture is composed of three main components as described in fig. 1 .

3.1 Scraper module

The scraper module is in charge of the preprocessing activity consisting in the page selection, crawling and parsing. The scraper module makes usage of various scraping tools as cURL [4], a library as Beautiful Soup [2] and a headless browser as Selenium [8] to access web pages. The selection consists of accessing the daily newly registered domains lists and choosing the domains containing terms belonging to the bag of word in the url. Crawling consist of establish

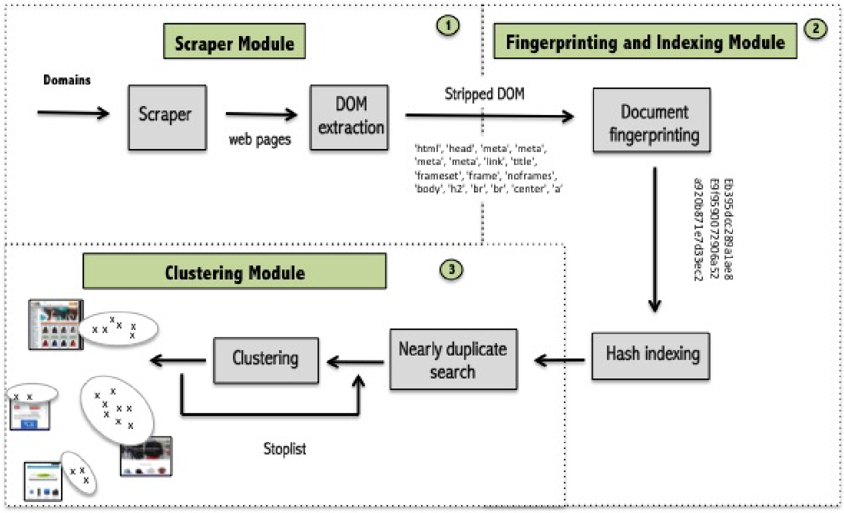

Figure 1: System Architecture

communication with the correspondent domain through the HTTP protocol and, in case of reachability, digest (download) the domain homepage. Web pages are based on an underlying object structure, namely, the DOM. The DOM is a cross-platform and language-independent application programming interface that treats an HTML, XHTML, or XML document as a tree structure (fig. 2) where each node is an object representing a part of the document. The DOM defines the logical structure of documents and the way a document is accessed and manipulated.

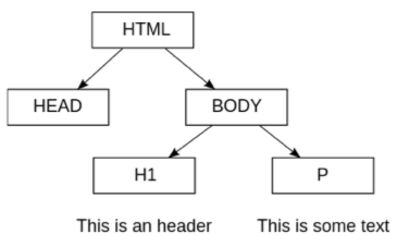

Figure 2: DOM Tree

Stripping procedure is a transformation which manipulates the crawled document, removes contents and terms not of interest to obtain a linear stripped DOM tree. Consequently, all pages contents, all the tag attributes are ignored. as for example in
[html', 'head', 'meta', 'meta', 'meta', 'meta', 'link', 'title', 'frameset', 'frame', 'noframes', 'body', 'h2', 'br', 'br', 'center', 'a',..]

3.2 Fingerprinting and Indexing module

A fingerprinting hashing algorithm is a transformation that maps an arbitrarily large data item (in this paper, we refer as items to web pages) to a much shorter bit string that is likely to identify the original data. More formally, lets introduce H as a family of hashing functions h mapping an input x to a compact code y as y = h(x).

In our case we opted for a family H of transformations, the locality sensitive hashing which let the generated fingerprints to maintain an internal similarity even in case of an item small change. It means that a similar set of item in the domain of these functions have a higher probability of colliding in the range space than dissimilar ones. More formally an LSH [25] family F is defined for a metric space M=(M,d), and an approximation factor c > 1. This family of function F is a family of functions h: M -> S which map elements from the metric space to a bucket s ∈ S . The LSH family satisfies the following conditions for any two points p, q ∈ M , using a function h ∈ F which is chosen uniformly at random:

- if d(p,q) ≤ R then h(p) = h(q) (i.e. p and q collide) with probability at least P_1
- if d(p,q) ≥ cR , then h(p)=h(q) with probability at most P_2

A family of function is of interest when $P_1 > P_2$. Such a family F is called (R, cR, P_1, P_2) -sensitive. For the objectives of this research, we identified Charickar's simhash algorithm [11] as an appropriate algorithm for fingerprint generation. The advantage of this algorithm is the compression of high dimensional documents to small sized fingerprints of a chosen size, traditionally 64 bits. So a standard web pages of hundreds of KB size is firstly linearized to a stripped DOM version and then the correspondent fingerprint is generated with the simhash algorithm (Table 1).

Domain	Fingerprint
Domain 1	eb395dcc289a1ae8
Domain 2	e9f9590072906a52
Domain 3	a920b871e7d33ec2

Table 1: Registering time frame for a multi-term cluster of domains

Lastly the fingerprints are indexed using hash tables, formed by storing the items with similar codes in hash buckets.

3.3 Clustering module

After creating a compact signature for documents and hashing them in hash tables, a way to efficiently compute a similarity measure for constructing clusters of homogeneous documents, has now to be found. More formally, given a set of hash items X = x_1 .. x_n and given a query item q in q_1 .. q_m the goal is to find the closest points to q in X as candidates for building up clusters. As similarity measure between hash items, we use the Hamming distance. Given a set of items hash X, we define the Hamming distance between two items x_1, x_2 to be the number of components in which they differ. As an example in a cluster C containing (x_1 = 00011001, x_2 = 00010011), the Hamming distance H(x_1,x_2) = 2 because the bit-hash differs in the 5th and 7th positions. While fingerprinting allows to computer quickly and efficiently the resemblance of two documents, it does not solve the computational issue of considering all possible pairs. However, the locality sensitive property of LSH implies that similar items have a larger probability to be mapped to the same bucket than dissimilar items, so for a query instance x, it can be used for a first approximation the instances stored in buckets containing x. From each of the hash buckets we randomly choose an item as query item qi for a similarity search. This search can be executed efficiently as search of all fingerprints that differ from a given fingerprint in at most k bit positions, where k is a small integer (Manku, 2007). The results are then aggregated using an agglomerative clustering technique where elements are merged into the cluster until the similarity condition is respected.

4 Experimental setups

The computer used for the experiments is an IMac, with 3.4 GHz intel Core I7 with 16 GB of ram. The Data Corpus has been generated from the daily list of registered .com domains in the month of September 2017. The daily list contains an average of 100K domains a day with peaks during the working days and cusps during weekend. For the purposes of this research, we oriented on potential counterfeiting domains by using a bag of words of interest containing both fashion brands and transactional terms as it follows:

Bag of words = louboutin, iceberg, armani, gucci, hogan, iceberg, vuitton, prada, tods, moncler, nike, adidas, sales, outlet, ferragamo.

Then the newly registered domains were filtered on the base of the bag of words and a daily monitoring of resulting domains were executed for a period of three weeks. Globally 4000 domains have been monitored.

5 Results and observations

A simhash fingerprints dimension of 64 bits was used for the experiment. For the corresponding value of k, the tolerated bit positions range between similar fingerprint, we evaluated a range from 1 to 5. It is clear that boosting the value of k implies an increasing tolerance of differences between fingerprints and the gathering of items structurally more different. The value of k is directly proportional to the number of multi-item clusters (2 or more items) and inversely proportional to the number of clusters containing one unique item (fig. 3). Subsequently a reduced number of clusters are generated and it is altered the distribution of items within a cluster.

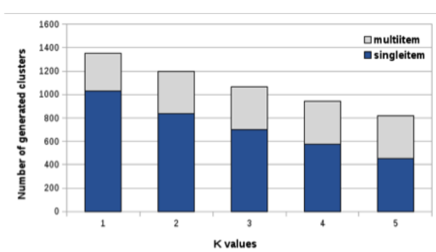

Figure 3: Value of k and cluster single-item, multi-item distribution

For choosing the correct value of k, we computed two internal measures: cohesion (how closely related are the items in a cluster) and separation (the distinctiveness of a cluster from the others) using the fingerprint with a higher similarity with all the other fingerprints within the cluster, as the centroid of a cluster. Results indicate a reasonable value of k=3 as already evidenced in other studies [20].

To compute the appropriateness of data partitions we manually inspected the clusters by visual analysis and source code inspection. Two experts labeled the true positives(tp),the decision assigns two similar documents to the same cluster, and false positives(fp), the decision to assigns two dissimilar documents to the same cluster. Two misleading results were noticed. Firstly, some javascript injected pages were not processed correctly, as in the case of specific web pages building platforms as vix.com or leadpages.net, and it led to the creation of few high-density clusters. For maintaining a reasonable computational time we did not further investigate domains sharing those fingerprints as it would require a much higher scraping time. We either noticed that in case of sites not yet published, due the freshness of newly registered domains, the framework analyzed and clustered the corresponding parking pages (the placeholder for a newly registered domain before a website is ready for launching) for services as *godaddy.com*, *a2hosting* and so on.

We then calculated the measure of precision, the percent of positive predictions, as it follows:

$$Precision = \frac{tp}{tp + fp}$$

From this measure(Table 2), it appears that the proposed framework performs well, in the case of traditional web pages.

Typology	Precision
Parking pages	0,940066593
Javascript injected pages	0,916666667
Traditional web pages	0,847250509

Table 2: Cluster precision by page typology

The fingerprints of parking pages and javascript injected pages, corresponding to nearly the 55% of the total clusters, were inserted in a stoplist and removed at the successive clustering phase. In fig. 4 the clusters occurrence by cluster size.

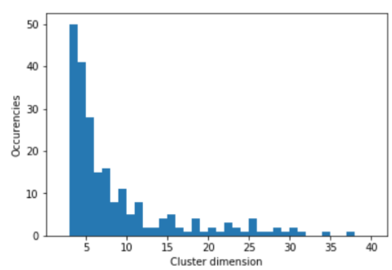

Figure 4: Cluster size and occurrence

We then switched to corresponding domains url to investigate the distribution of terms within the clusters. By the comparison of terms from our bag of words and the domains urls, we then proceeded to separate clusters related to one term i.e. single-term cluster from clusters related to more than one term i.e. multi-term clusters. A single-term cluster contains fingerprints of domains related to only one term of the bag of words as *loveinnike.com*, *hotnike.com*, *uknike.com* with *nike* as term. A multi-term cluster contain fingerprints of domains related to two or more term from the bag of words as *pradashoes.com*, *nikempire.com*, *moncletcoat2019.com* with *nike, prada, moncler* as terms. It results, with such segmentation, a partition of 109 multi-item single-term clusters and 73 multi-item multi-term clusters.

The results of single-term cluster analysis reinforce the traditional brand domain monitoring analysis, which works mainly on lexical similarity, offering both a measure of a lexical and structural similarity (fig. 5).

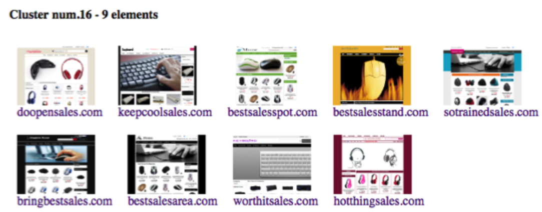

Figure 5: Items from single-term cluster sales

The results of multi-terms cluster analysis leads to the discovery of not obvious connections between domains based on the usage of similar structured templates. In fig. 6 an example of two domains registered on the same day without any lexical url similarity, promoting different products using different keywords, were clustered together due the usage of a nearly similar structural layer.

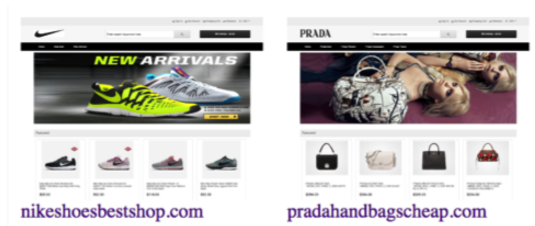

Figure 6: Items from a multi-terms cluster

Domain name	Created on	Updated on
hotthingsales.com	2017-09-18T06:42:58Z	2017-09-18T07:11:46Z
worthitsales.com	2017-09-18T06:43:02Z	2017-09-18T07:11:51Z
bringbestsales.com	2017-09-18T06:43:03Z	2017-09-18T07:11:45Z
doopensales.com	2017-09-18T06:43:04Z	2017-09-18T07:11:38Z
keepcoolsales.com	2017-09-18T06:43:15Z	2017-09-18T07:11:45Z
sotrainedsales.com	2017-08-29T09:14:12Z	2017-08-30T00:57:47Z
bestsalesspot.com	2017-08-29T09:14:21Z	2017-08-30T00:57:42Z
bestsalesstand.com	2017-08-29T09:14:43Z	2017-08-30T00:57:42Z

Table 3: Registering time frame for a single-term cluster of domains

We then reassessed clusters results taking in account the time variable. For each domain the time of registration and update was extracted using a whois service. A correlation both on structural similarities and domain registration timing could be a direct indication of the usage of automated bulk publishing tools and an indirect confirmation of the affiliation to a moniker domains network. Preliminary results confirm this correlation and indicate the usage of bulk registering tools. Table 3 shows as domains of the single-term cluster (fig. 5) have been generated in very specific time span.

A block of five domains was registered in a time frame of 17 seconds and updated within

Domain name	Created on	Updated on
pradahandbagscheap.com	2017-09-07T02:29:48Z	2017-09-07T02:33:14Z
nikeshoesbestshop.com	2017-09-07T02:29:48Z	2017-09-07T02:36:17Z

Table 4: Registering time frame for a multi-term cluster of domains

6 seconds. Another block of four domains was registered in a time frame of 21 seconds and updated within 4 seconds. Same correlation between structural layer similarities and domain registering time frame was found in multi-terms clusters (fig. 6) as shown in Table 4.

6 Conclusion

In this paper a framework for identifying potentially malicious domains on the basis of recognition of recurrent structural patterns in a restricted registering time frame is proposed. The framework has been tested on a corpus of 4000 domains, filtered from newly registered .com domains, for a period of three weeks, on the basis of a bag of words of terms of interest. The results reveal unusual aggregations of domains sharing a similar, nearly identical template structure suggesting the potential use of management systems or content generators for pursuing malicious activities. These findings suggest a possible strategy for enhancing domain name monitoring by integrating, in addition to lexical analysis, the fingerprint analysis of website structure. Furthermore, we reassessed the obtained results with OSINT data, as the domain registration and update time. A correlation both on structural similarities and OSINT data emerges validating the hypothesis of usage of bulk publishing tools and confirming the existence of huge networks of potentially malicious domains added on the Internet on each day. It clearly emerges the necessity of enhancing traditional domains name monitoring from an individual domain analysis to domains networks analysis. We leave the further investigation of an efficient methodology to outline malicious networks of domains, for future research.

References

[1] Document object model (dom) level 3 core specification. http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/, 2004.

[2] Beautiful soup. https://pypi.python.org/pypi/beautifulsoup4, 2017.

[3] Black hat tools. https://www.blackhatworld.com/forums/black-hat-seo-tools.9, 2017.

[4] Curl. https://curl.haxx.se, 2017.

[5] The domain name industry brief. https://investor.verisign.com/releasedetail.cfm?releaseid=980215, 2017.

[6] Opendns. https://www.opendns.com, 2017.

[7] Phishtank. https://www.phishtank.com, 2017.

[8] Selenium. http://www.seleniumhq.org, 2017.

[9] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. In *Selected Papers from the Sixth International Conference on World Wide Web*, pages 1157–1166, Essex, UK, 1997. Elsevier Science Publishers Ltd.

[10] Claudio Carpineto and Giovanni Romano. Learning to detect and measure fake ecommerce websites in search-engine results. In *Proceedings of the International Conference on Web Intelligence*, WI '17, pages 403–410, New York, NY, USA, 2017. ACM.

[11] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 380–388, New York, NY, USA, 2002. ACM.

[12] Marco Cova, Christopher Kruegel, and Giovanni Vigna. There is no free phish: An analysis of "free" and live phishing kits. In *Proceedings of the 2Nd Conference on USENIX Workshop on Offensive Technologies*, WOOT'08, pages 4:1–4:8, Berkeley, CA, USA, 2008. USENIX Association.

[13] Europol. Over 4500 illicit domain names seized for selling counterfeit products. https://www.europol.europa.eu/newsroom/news/over-4500-illicit-domain-names-seized-for-selling-counterfeit-products, 2016.

[14] Dennis Fetterly, Mark Manasse, and Marc Najork. On the evolution of clusters of near-duplicate web pages. In *Proceedings of the First Conference on Latin American Web Congress*, LA-WEB '03, pages 37–, Washington, DC, USA, 2003. IEEE Computer Society.

[15] T. Gowda and C. A. Mattmann. Clustering web pages based on structure and style similarity (application paper). In *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, pages 175–180, July 2016.

[16] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, April 1950.

[17] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[18] Hisashi Koga, Tetsuo Ishibashi, and Toshinori Watanabe. Fast agglomerative hierarchical clustering algorithm using locality-sensitive hashing. *Knowledge and Information Systems*, 12(1):25–53, 2007.

[19] J Prasanna Kumar and Paladugu Govindarajulu. Near-duplicate web page detection: an efficient approach using clustering, sentence feature and fingerprinting. *International Journal of Computational Intelligence Systems*, 6(1):1–13, 2013.

[20] Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. Detecting near-duplicates for web crawling. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 141–150, New York, NY, USA, 2007. ACM.

[21] V. A. Narayana, P. Premchand, and A. Govardhan. Fixing the threshold for effective detection of near duplicate web documents in web crawling. In *Proceedings of the 6th International Conference on Advanced Data Mining and Applications: Part I*, ADMA'10, pages 169–180, Berlin, Heidelberg, 2010. Springer-Verlag.

[22] V. A. Narayana, P. Premchand, and A. Govardhan. Article: Performance and comparative analysis of the two contrary approaches for detecting near duplicate web documents in web crawling. *International Journal of Computer Applications*, 59(3):22–29, December 2012. Full text available.

[23] John Wadleigh, Jake Drew, and Tyler Moore. The e-commerce market for "lemons": Identification and analysis of websites selling counterfeit goods. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1188–1197, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[24] J. H. Wang and J. Z. Lin. Improving clustering efficiency by simhash-based k-means algorithm for big data analytics. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1881–1888, Dec 2016.

[25] Wikipedia. Locality sensitive hashing. https://en.wikipedia.org/wiki/Locality-sensitive_hashing, 2017.