Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Content Matters: Clustering Web Pages for QoE Analysis with WebCLUST

LUIS ROBERTO JIMÉNEZ[1], (Graduate Student Member, IEEE), MARTA SOLERA[1], MATÍA TORIL[1], CAROLINA GIJÓN[1] and PEDRO CASAS[2]

[1]Instituto de Telecomunicación (TELMA), Universidad de Málaga, CEI Andalucía TECH E.T.S. Ingeniería de Telecomunicación, Bulevar Louis Pasteur 35, 29010 Málaga, Spain

[2]AIT Austrian Institute of Technology GmbH, Giefinggasse 4 Vienna, s/n E-1210, Austria

Corresponding author: Luis Roberto Jiménez (e-mail: lrjp@ic.uma.es).

This work has been funded by the Spanish Ministry of Science, Innovation and Universities (RTI2018-099148-B-I00), the Presidency of the State Research Agency (FPI/BES-2016-07631) and the Spanish Ministry of Education, Culture and Sports (FPU17/04286).

ABSTRACT The properties of a web page have a strong impact on its overall loading process, including the download of its contents and their progressive rendering at the browser. As a consequence, web page content has a strong impact on the experience of web users. In this paper, we present WebCLUST, a clustering-based classification approach for web pages, which groups pages into quality-meaningful content classes impacting the Quality of Experience (QoE) of the users. Groups are defined based on standard Multipurpose Internet Mail Extensions (MIME) content breakdown and external subdomain connections, obtained through in-browser, application level measurements. Using a large corpus of multi-device, heterogeneous web content and QoE-relevant measurements for the top-500 most popular websites in the Internet, we show how WebCLUST can automatically identify relevant web content classes showing significantly different performance in terms of Web QoE relevant metrics, such as Speed Index. We additionally evaluate the impact of content caching and device type on the identification performance of WebCLUST, showing how content classes might look significantly different, depending on the access device type (desktop vs mobile), as well as when considering browser caching. Our findings suggest that Web QoE assessment should explicitly consider page content and subdomain embedding within the analysis, especially when it comes to recent work on Web QoE inference through machine learning models. To the best of our knowledge, this is the first study showing the impact of web content on Web QoE metrics, opening the door to new Web QoE assessment strategies.

INDEX TERMS Web, Quality of Experience, clustering, WebPageTest, service performance.

I. INTRODUCTION

Recent advances in information and communication technologies have brought new opportunities for accessing digital content. As a result, user expectations for products and services involved in this interaction are increasing. This trend has forced network operators to change the way they manage their systems from a network-centric to a user-centric approach. Thus, customer experience management is a key process in the daily routine of network operators [1].

User satisfaction depends on multiple factors (e.g., human, system, context), which makes it difficult to measure and analyze [2]. Yet, network operators and content providers attempt to estimate service performance as perceived by the customer, defined as Quality of Experience (QoE) [3]. The simplest approach is to conduct subjective tests with real users. However, subjective tests are time-consuming and not valid for large-scale real-time monitoring. Alternatively, objective methods can use measurements collected by applications or network elements to infer QoE [4]. Objective methods are classified into signal-based, bitstream-based and parametric models. The former two techniques rely on decoding the content, which requires access to the application layer and are therefore only suitable for service providers. In contrast, parametric packet-layer methods analyze protocol messages to identify the different stages of a session, from which Service Key Performance Indicators (S-KPIs) can be obtained on a per-session basis. Then, S-KPIs are translated into Mean Opinion Score (MOS) values [5] by formulas derived in subjective tests. In the past, S-KPIs could be directly measured or at least inferred using deep packet inspection monitoring technology, deployed at key network interfaces of the core network [6]. Unfortunately, traffic encryption used by most content providers nowadays prevents the analysis of session events [7]. In the absence of other methods, simpler parametric models are developed per service to blindly relate basic network-level quality-of-service measurements (e.g., average session throughput) to MOS scores [8]. This approach is followed by most frameworks for large-scale, in-network passive monitoring [9] [10]. Recently, these platforms have been extended with data analytics capabilities to isolate the indicators that better reflect user experience and predict their trends [11]. These advanced QoE models still have to be calibrated in field trials by comparing their estimates with real measurements obtained with automated terminal agents [12].

One of the most challenging services to deal with in QoE management is web browsing. Originally designed for accessing static content, current web browsing consists of many intertwined processes difficult to characterize (e.g., user interactions, object downloading from multiple domains, content conversion, scripts, visual rendering, etc.). As a consequence, web experience has to be measured from multiple indicators showing service availability (e.g., web access failure ratio, first access time), integrity (e.g., average download data rate, download time) and retainability (e.g., download success ratio). Some of these indicators depend on subjective factors, as it is difficult to define when the user thinks that a page download has completed (e.g., full load, above-the-fold content or non-interactive time). The web page loading time is affected by several delay components, some of which cannot be directly measured by traffic monitoring tools [12]. For instance, delays associated to Domain Name Service (DNS) resolution and TCP handshake are often not included since these tools are only able to identify web browsing service once the first HyperText Transfer Protocol (HTTP) message is sent. Likewise, monitoring tools cannot measure the delay associated to processes executed in the terminal (e.g., scripts, rendering). In addition, the rich and heterogeneous nature of web page contents increases the complexity of the analysis, as the page download and rendering process and the associated user experience depend on the type of media, be it text, images, video, dynamic contents such as JavaScript, and more. Indeed, previous work on Web QoE modeling and assessment has already hypothesized on the need for content-specific Web QoE models [13]. All these issues justify the need for Web QoE models specifically designed for different types of web pages.

We take a first step in the automatic characterization and classification of web pages by content type, aiming at a more atomic and content-tailored analysis of Web QoE. We introduce WebCLUST, a clustering-based classification approach for web pages, which groups pages into quality-meaningful content classes impacting the QoE of the users. Using application-level, in-browser measurements, WebCLUST uses the Multipurpose Internet Mail Extensions (MIME) content breakdown of a web page and its TCP connections to external subdomains to build input features, which are then used to automatically identify groups of web pages sharing similar content characteristics. Groupings are built in an unsupervised manner through k-means, one of the simplest and most well-known clustering algorithms. Unsupervised learning avoids the need for labeled training datasets, which are difficult to obtain and maintain, as web technologies constantly evolve. The resulting groups are validated through the analysis of relevant performance indicators associated to web service, including QoE-related metrics such as Speed Index, first interactive time, fully loaded time, and more. In-browser measurements are collected through a custom web measurement platform, built on top of WebPageTest (WPT) [14], the default, open-source web-performance-analysis tool used both in industry and academia.

We apply WebCLUST to multiple web datasets corresponding to the top 500 most popular websites in the Internet, spanning different end-device configurations; these include the usage of both desktop and smartphone devices, as well as the usage of browser caching. Our analysis demonstrates how WebCLUST can identify relevant web content classes with different Web QoE; in addition, it also shows how web content varies when changing the end-device type and the browser's caching configuration.

The main contributions of this work are as follows: (a) the definition of a set of web page descriptors to characterize web pages from a QoE-relevant perspective, (b) a method for QoE-relevant web page classification in the absence of ground truth, and (c) an analysis of the impact of device type and browser caching on web page characteristics, their associated QoE, and the performance of the proposed method. The remainder of the paper is structured as follows. Section II reviews related work and presents relevant contextual concepts. Section III presents the WebCLUST system, including the measurement platform conceived for the study. Section IV elaborates on the evaluation of WebCLUST functioning and performance using desktop measurements. The impact of the device type and the usage of browser caching on the characterization of web pages and the overall functioning of WebCLUST is studied in Section V, relying on mobile/smartphone measurements. Finally, Section VI concludes this work.

II. RELATED WORK

Several objective metrics have been proposed in the literature to measure Web QoE. These metrics can be classified into network-related and visual-related metrics. The former are computed from network-layer measurements collected by traffic monitoring tools, whereas the latter requires access to the application layer. In the first group, Page Load Time (PLT) is often used to measure the performance of web browsing in the industry and academia, since it is well defined and can be quantified precisely [15].

Complementary, several tools build a timeline with all events related to the download of individual objects [16]. From this information, other relevant indicators can be obtained by detecting specific time events (e.g., time to first byte, document object model load time) or aggregating statistics (e.g., byte/object index [17]). Network indicators can be enriched by adding processing (i.e., on-load time) and rendering times (i.e., time to render the full page and time to first picture painted [18]). However, none of these indicators take into account that not all the objects in a web page are equally important. A good example are dynamic web pages (e.g., advertising), in which page load time might be arbitrarily large even if the important content is downloaded almost instantly. Likewise, the visual part of the page depends, among others, on the screen size, which is not reflected by the page load time [19]. To circumvent these issues, alternative metrics have been proposed based on the progress of the visual content, namely Above the Fold Time (AFT) [20] and Speed Index (SI) [21], [22]. Unfortunately, these indicators are computed by analyzing screen shots of the visible part of the page, which is a complex and time consuming operation that can only be done at the client side. In [23], an unsupervised learning system to measure web performance by analyzing the timeline of events at the transport layer is proposed. The output of the system is strongly correlated with the on-load time and the SI. More recent work relying on machine learning models to infer the SI from (encrypted) network traffic level features is presented in [24], [25], including the analysis of Web QoE in mobile devices [25]. A detailed comparison of expert QoE models (naïve (linear), ITU-T (logarithmic) [26], and IQX (exponential) [27]) against machine-learning based Web QoE models (support vector regression, regression trees, and random forest) is presented in [13]. Authors conclude that expert models can accommodate new metrics beyond PLT, achieving accuracy comparable to that of data-driven, machine-learning based models. However, it is stressed that a single model cannot cope with the wide heterogeneity of web pages. At the same time, per-page models show higher accuracy, but the modeling approach is not scalable, given the millions of web pages available in the Internet. To solve the latter problem, web pages can be classified into groups, so that an appropriate QoE model can be derived per group.

Large-scale characterization and analysis of web pages has been the subject of previous work [28], [29], early showing the high complexity of modern web page contents and the underlying hosting/server infrastructure. Content-based web page categorization has been done in the past mainly for retrieval and information management purposes. Semantic classification based on content is key for maintaining web directories [30], web search/ranking [31] or contextual advertising [32]. A comprehensive survey of criteria, features, and methods for content-based web categorization is presented in [33] and [34]. Web page content can be classified based on subject (e.g., arts, business, sports), role (e.g., personal, institutional) and opinion (attitude). Content features used to define groups of web pages include word chains [35], HyperText Markup Language (HTML) tags [36], images [37], and spatial relationship between objects [38]. Features for the analysis might also include characteristics from "neighboring" pages [39]. In addition, web page categorization can be done based on creation strategy (static or dynamic) or design technology (Flash, HTML, etc.). To the authors' knowledge, no method has been proposed to classify web pages from a QoE perspective, considering how different page elements affect download times.

Given a relevant set of features (generally manually selected, based on domain-knowledge) groups can be obtained by supervised learning algorithms trained with labeled datasets (e.g., k-nearest neighbors, support vector machine, neural network [40]), semi-supervised algorithms combining both labeled and unlabeled data (e.g., co-training [41]), or unsupervised algorithms (e.g., relaxation labeling [42]). The first two approaches require manually labeling web pages, which is a complex and error-prone task. Therefore, WebCLUST relies exclusively on unsupervised learning techniques for the web page characterization task; in particular, WebCLUST uses clustering approaches, applied to features describing the share of bytes for specific contents (e.g., share of image bytes, share of video bytes, share of JavaScript (JS) bytes, etc.), as well as the number of external contents and third-party resources embedded in the page, reflected by the number of connections to external subdomains.

A. WEB PAGE LOADING 101

To assess the impact of connection performance on web experience, it is important to understand first how the browser renders a web page [43]. The browser engine starts parsing the HTML text as soon as a few characters of the document are received. The result is a Document Object Model (DOM), which is a tree structure with nodes representing HTML elements. The DOM is built incrementally, but can be interrupted by the execution of embedded or external JS scripts. Concurrently, the browser provides styles to HTML elements as they become available by reading Cascading Style Sheets (CSS) from embedded or external sources. The result is a CSS Object Model (CSSOM), including only elements that can be printed on the screen. Then, the browser builds a Render Tree with objects that will be visible, which is used to compute the layout of visible objects and print individual elements on the screen. The process continues until all page objects are downloaded and displayed.

Web performance can be measured based on session events [13]. The foremost of them are the reception of the first byte (first byte), the display of the first pixel, image/text or large pieces of them (first paint/first contentful paint/large contentful paint), the start and consolidation of interactivity (first interactive/consistently interactive), the rendering of the visible part of the page (above-the-fold time), the processing of all page elements (onload) and the end of the network activity (full load). All these events are affected by page

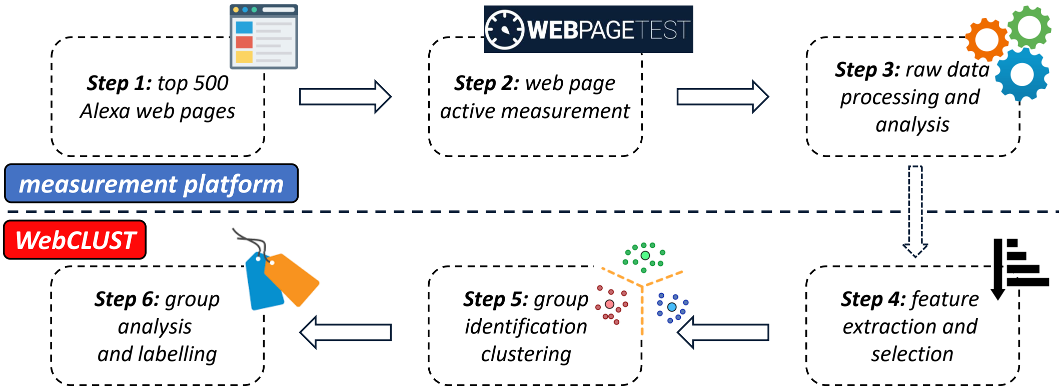

Step 1: top 500 Alexa web pages
WEBPAGETEST
Step 2: web page active measurement
Step 3: raw data processing and analysis
measurement platform
WebCLUST
Step 6: group analysis and labelling
Step 5: group identification clustering
Step 4: feature extraction and selection

FIGURE 1. Web measurement platform for WebCLUST analysis. Web measurements are instrumented and collected through a WPT private instance. WebCLUST identifies relevant groupings on top of the pre-processed web measurements, auto-calibrating the underlying k-means algorithm.

content and design, browser engine, and network conditions.

III. THE WEBCLUST SYSTEM

In this section, we describe the WebCLUST system in detail. We firstly present the web measurement platform conceived for the study, which allows for automated browsing of web pages and collection of relevant data using the WPT suite of tools. We then describe the set of features describing a single web page for the subsequent clustering analysis and present the Web QoE-relevant features that serve for web service performance assessment and analysis. Last, we describe the unsupervised grouping step based on k-means clustering, providing details on the calibration of the algorithm and the labeling of the resulting clusters.

A. WEB MEASUREMENT PLATFORM

Fig. 1 shows a diagram of the platform conceived to automate the collection of web measurements and the subsequent clustering. The platform is made up of two components: the **measurement platform** itself, responsible for automating the web browsing activity, the collection of measurements, and the extraction of specific S-KPIs or features, and the **WebCLUST** system, which classifies webs into groups based on their characteristics. The measurement platform consists of a desktop PC running a local instance of WPT [14], which automates the browsing of individual web pages, collecting all data involved in the loading process, from the initial web page request until the page is fully loaded. For each visited web page, a set of pre-defined S-KPIs is computed from the collected raw data. WebCLUST then selects a sub-set of these features for the clustering step, which is executed on top of the full dataset of web measurements. The last step of the system consists of the analysis and interpretation of the generated groups or clusters, for the subsequent labeling in relevant web page content classes. In particular, WebCLUST defines six web page content classes, which are described in Section IV.

To select a significant sample of websites capturing the diversity in the most widely used Internet applications (search engines, news, e-commerce, social media, file download sites, etc.), we use Alexa's top global 500 sites list [44]. Alexa's ranking is based on a combination of average daily visitors and pageviews over the last month. Web pages can be located anywhere in the world. As the access to these pages is done from a local server located in our laboratory (University of Málaga – EU), those sites not served from EU-located servers or Content Delivery Networks could perform worse. In this work, web page server location has not been taken into account, but it is certainly a relevant aspect to consider for future work.

Using this list of 500 websites, we rely on WPT to execute the measurement process. WPT is designed to measure the performance of a web page for different browsers and terminals and is used mainly in the industry for website optimization purposes [14]. A private instance of the WPT framework is installed on a windows desktop PC containing a standard HTTP web server. The installation of a private instance enables WPT advanced features, such as control of queues and test agents, removal of daily testing limitations, creation of custom metrics, and extensive tests (bulk testing). To implement a private instance, a WPT server and an agent emulating client requests are launched on the PC. Specifically, WPT version 20.05.01 is used. This version supports measurements on Internet Explorer, Chrome, Firefox, and Safari browsers. In our study, Chrome was the selected web browser. A total of six tests are performed per web page to build a more robust dataset and avoid the presence of outliers. Thus, $500 \times 6 = 3,000$ individual page loading tests are conducted.

WPT generates a comprehensive list of features describing both the contents of a web page and its loading performance/timing events. For each web page loading session, we extract about 90 different S-KPIs, including both content-related metrics (in particular metrics reflecting the share of bytes for specific contents (image, video, text, etc.) as well as the number of connections to external resources) and Web-QoE related metrics such as PLT, Speed Index, Byte/Image/Object Index, rendering and interactive

TABLE 1. WebCLUST web page features/S-KPIs for clustering.

Description	# Features	Feature IDs
content breakdown (%)	8	Image, JS, HTML, Font, CSS, Flash, Video, Other
# connections	1	CNX

timing events, and more. The reader is referred to WPT documentation for a detailed description of the S-KPIs collected by the system [45] [46]. We complement the set of WPT S-KPIs with an identification of all third-party contents related to advertisement sites. Advertisement is pervasive in the web, and as we show in our results, it has a significant impact on the web page loading process, due to its dynamic content nature and its distributed location. For this purpose, the hostnames of all TCP connections registered by WPT are extracted and compared with EasyList, one of the largest online databases registering the hostnames belonging to online advertisement companies [47]. Originally designed for Adblock applications, the EasyList filter lists are sets of rules that automatically remove unwanted content from web browsing sessions, including adverts, banners and trackers. The obtained dataset consists of about 3,000 individual tests, each one described by a test id, the corresponding web page URL, the broad set of WPT S-KPIs, and the number of external TCP connections belonging to advertisement domains.

B. FEATURES AND CLUSTERING APPROACH

As the goal of WebCLUST is to identify web page classes based on content properties, we base the characterization of each web page and the clustering of pages on a small and tractable set of S-KPIs describing its core components. In particular, we consider eight MIME-type content-breakdown descriptors, as well as a the number of external contents and third-party resources embedded in the page, the latter reflecting how complex and distributed the page content is. Table 1 summarizes the set of features or web page descriptors used in the clustering step, which are described next.

Content breakdown shows the total data volume (bytes) downloaded in a fully loaded page, broken down by MIME type, expressed in relative terms (percentage of total downloaded bytes). It consists of eight S-KPIs indicating the specific content type: Image, JS (JavaScript), HTML, Font, CSS, Flash, Video, and Other. Image denotes image objects, JS denotes code to create dynamic web interactivity, HTML corresponds to HTML resources, Font denotes resources to modify the size, color, or font of the text, CSS denotes cascading style sheets resources to control the appearance of documents, Flash denotes Flash resources used for animation, Video denotes video sequences, and Other denotes all other content types.

Connections (CNX) corresponds to the number of TCP sockets opened with support subdomains to download embedded objects until the page is fully downloaded.

TABLE 2. WebCLUST web performance/QoE characterization features.

Description	Units	Feature IDs
start render	s	SR
first interactive	s	FI
speed index	s	SI
full load time	s	FLT
fully loaded bytes	KB	FLB
# advertisement domains	–	ADs
advertisement domain bytes	KB	ADB

To characterize the resulting clusters and to assess their associated Web QoE, we additionally take seven S-KPIs linked to time-based loading performance and characterization of embedded contents, shown in Table 2 and described as follows:

Start Render (SR) is the time elapsed from the page request until the first non-white content is painted in the browser display. In other words, how long the user waits before seeing any part of the page.

First Interactive (FI) is the time elapsed from the page request until the page is responsive to user interaction [45].

Speed Index (SI) is an aggregate metric representing the average time at which visible parts of the page are displayed in the viewport [21]. Different from instant-like metrics (SR, FI, PLT, etc.), the SI considers the whole visual progress of the page loading, measuring how quickly the page contents are visually populated.

Fully Loaded Time (FLT) is the time elapsed from the page request until 2 seconds of no network activity after the document complete event, which generally corresponds to the time when all of the static page content has loaded.

Fully Loaded Bytes (FLB) is the total data volume downloaded until the page is fully loaded.

Advertisement Domains (ADs) is the number of TCP connections to advertisement domains, according to EasyList [47].

Advertisement Domain Bytes (ADB) is the total data volume downloaded for all advertisement domains ADs.

The group discovery is done through the well-known k-means algorithm [48] [49]. k-means is a partitioning-based algorithm that assigns samples to a fixed number of disjoint clusters k, based on their similarities. Each cluster is represented by a centroid. The algorithm starts by randomly selecting k samples within the dataset as initial centroids. Then, k clusters are formed by associating every sample of the dataset to the nearest centroid, according to some distance criterion (e.g., Euclidean distance, Hamming distance, etc.). Centroids are recomputed after all samples have been assigned to one of the k clusters. This process is repeated until a convergence criterion is met; in particular, as the goal of k-means is to find centroids that minimize their intra-cluster variance, the standard stopping criterion corresponds to the total sum of squared distances from each sample to its corresponding cluster centroid (referred to as WSS_T reaching a value below a certain threshold). If we define $WSS(j)$ as the intra-cluster sum of squared distances to cluster centroid c_j, then WSS_T can be expressed as:

$$WSS_T = \sum_{j=1}^{k} WSS(j) = \sum_{j=1}^{k} \sum_{x_i \in \ j} |c_j - x_i|^2 \ , \qquad (1)$$

where j is the cluster index (i.e., web pages group), k is the number of clusters, c_j is the centroid of cluster j, i is the sample index (i.e., a web page), and x_i is the feature representation of sample i (i.e., web page descriptors). The k-means algorithm assumes normalized data, so that the different dimensions are comparable and have a similar scale. To this end, the eight web page features/S-KPIs reflecting the content breakdown are expressed as a ratio to the total page size, ranging from 0 to 1. In addition, a min-max feature scaling method is used to normalize the number of connections CNX [50]. The normalized value of CNX, denoted as CNX_{norm}, is computed as:

$$CNX_{\text{norm}} = \frac{CNX - CNX_{\text{min}}}{CNX_{\text{max}} - CNX_{\text{min}}} \ , \qquad (2)$$

where CNX_{max} and CNX_{min} are the maximum and minimum values of CNX in the dataset, respectively. Min-max normalization is one of the most common and simplest ways to normalize data. However, min-max scaling might not be robust with outliers. If this was the case, a standard Z-score normalization for CNX could be used, but the output might not be in the range [0,1].

In our specific web page clustering problem, the number of clusters k is unknown a priori. Therefore, as suggested in previous work [51], the optimal number of clusters is selected by comparing the evolution of WSS_T with the average silhouette score, obtained for a growing number of clusters k. Naturally, WSS_T decreases with growing values of k (i.e., when considering a more atomic partitioning of the data). The silhouette is a structural measure of the similarity of a clustered sample to other samples in its own cluster compared to those in other clusters. This measure is commonly used to assess the goodness of a clustering technique by checking that the identified clusters are compact and separated from each other. In this work, the silhouette score provides complementary information that is used to select the optimal number of clusters by visual inspection. The score ranges from -1 to +1, where a high value indicates that the sample is well matched to its own cluster and poorly matched to neighboring clusters. Thus, the higher the average silhouette score, the better. The best value of k is that providing a "reasonably low" WSS_T value (which depends on the specific application), while maximizing the average silhouette score. We would explain the notion of "reasonably low" for our particular problem in the evaluations, but in a nutshell, when increasing the number of clusters does not result in further noticeable reductions in WSS_T, then we should select the smallest value of k that results in the most cohesive partitioning, reflected by the silhouette score.

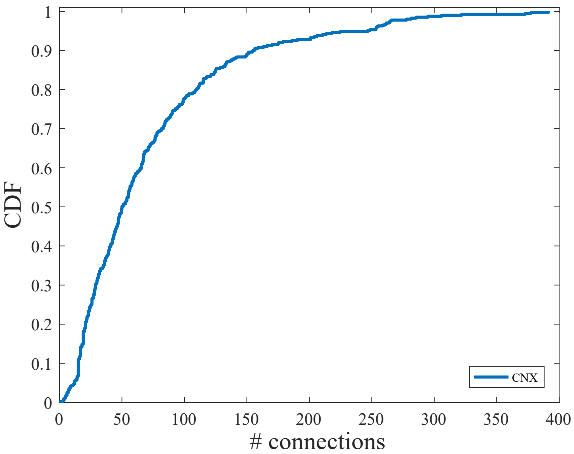


FIGURE 2. Distribution of number of connections to external subdomains (i.e., CNX), for desktop PC browsing without browser caching.

The last step consists of the interpretation of the clustering results, understanding the commonalities in each group. For ease of analysis, each group or **web class** is characterized by the median value of its features (share of MIME types and connections) for the web pages in the corresponding group. Note that we take median values instead of mean values, to filter out outliers and provide more robust characterizations. To enrich the characterization and analysis, we additionally consider the median values of the loading performance and content features previously defined in Table 2. Our study exclusively considers features that are intrinsic to web pages, and independent of the characteristics of the end-to-end web-browsing system, consisting of the web browser, the end-device, the network connection and the content distribution server. Our thesis, and as we show next, is that the contents of a web page (e.g., images, videos, JavaScript, etc.) and the number of connections to external subdomains (CNX) have a direct and strong influence on the user experience when browsing a web page, in terms of perceived loading times (e.g., Speed Index).

IV. WEBCLUST PERFORMANCE ASSESSMENT

In this section, we study the performance of WebCLUST through its application to a dataset of desktop web measurements, collected through the above-described web measurement platform (cf. Section III-A). We firstly describe the collected measurements, and then dive into the obtained results.

A. DATA DESCRIPTION

We collect a first batch of web measurements, using a desktop PC as terminal, emulating a user accessing the aforementioned top-500 Alexa websites (no browser caching is considered in this first evaluation). We refer to this dataset as the **PC-First** dataset. As explained before, six tests are performed per web page. Among the 500 web pages initially selected, 53 presented failed S-KPIs values in at least one of the tests. For the sake of robustness of the analysis, these web pages were discarded. Thus, the PC-First dataset includes the value of the selected sixteen S-KPIs (cf. Table 1 and Table 2) for a total of 447 web pages, where the value of these S-KPIs for each web page are computed as the median value over the six test runs. We take the median values to avoid skewed conclusions (i.e., outliers) due to variable network conditions, that are out of the control of the measurement platform, as the websites are hosted at the open Internet.

To get a first glimpse on the data, Figs. 2 and 3 depict the empirical distributions of the nine clustering features, for the 447 web pages. As shown in Fig. 2, about 50% of the web pages have less than 50 connections to external domains; the number of connections is above 150 for about 10% of the pages, and may reach up to almost 400 connections, showing the diversity in page complexity within the dataset. In terms of downloaded contents, Fig. 3 shows boxplots for the absolute values of the MIME data-volume contents (some upper whiskers are not shown for better visualization). The maximum size of Image, Video, and Others MIME contents are 21 MB, 186 MB, and 39 MB, respectively. Image and JavaScript represent the highest-volume contents, with a median size of 580 KB and 495 KB, respectively.

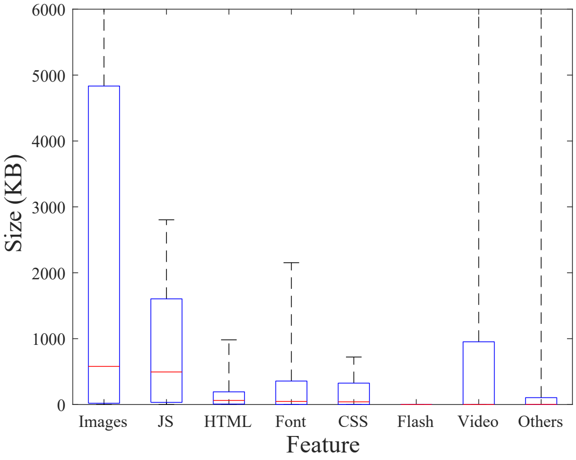


FIGURE 3. MIME content metrics boxplots. Images and JavaScript are the heaviest components of the top Alexa web pages, showing the richness of current websites.

B. CLUSTERING RESULTS

The 447 web pages are divided into groups by k-means, based on the number of support subdomains (CNX) and the MIME content breakdown ratios. To identify the optimal number of web classes, we follow the approach described in Section III: we perform ten different clustering tests by changing the number of clusters k from 1 to 10, evaluating the quality of each resulting partitioning. For each clustering test, 500 runs are executed with random initial centroid positions, selected by the k-means++ seeding scheme [52]. In a nutshell, k-means++ defines an improved initialization algorithm for the cluster-centroids, which enhances convergence of the partitioning. The maximum number of iterations per run is set to 2000 and the distance metric is squared Euclidean distance.

The number of groups to identify in a clustering analysis is rarely known in advance, and the literature provides multiple approaches to identify the optimal number of clusters in a data-driven manner. In general terms, the more clusters identified by an algorithm, the more homogeneous each cluster is (the extreme case is a number of clusters equal to the number of data points). However, if the number of clusters is high, then the usefulness of clustering analysis diminishes, as the interpretation of the obtained clusters becomes more difficult. There is therefore a trade-off between homogeneity of the resulting clusters and usefulness of the clustering analysis, as the number of clusters increases. Fig. 4 shows the evolution of the total intra-cluster sum of squared distances to clusters' centroids, WSS_T, and the average silhouette score for increasing number of clusters k. For comparison, values for each indicator are normalized to their maximum values. As expected, WSS_T decreases as k increases, but the amount of the decrease stagnates for $k = 6$ onwards. Likewise, the silhouette score shows two local maxima at $k = 2$ and $k = 6$. Taking both indicators, a value of $k = 6$ results in more compact and cohesive clusters with a bigger inter-cluster separation. Therefore, the solution splitting the web pages into six classes or web groups is the one adopted by WebCLUST. Additionally, we also tested other structural, unsupervised clustering quality metrics, such as the well-known Rank Index [53], and even considered density-based evaluation metrics, such as DBCV (Density-based Clustering Validation) [54] and CDbw (Composed Density validity index) [55]. However, results were comparable to the usage of silhouette scores, so we kept the latter, which is by far simpler and easier to understand and interpret.

To characterize the resulting web classes, Table 3 presents the values of the sixteen S-KPIs for each of the clusters. For each S-KPI, the table shows the median value for the web pages within the corresponding cluster. Again, using median values provides better results than mean values in our analysis, given the heterogeneity of web pages (reflected in Figs. 2 and 3). Probably here where the 95%-percentile could make more sense, at least when considering the QoE-relevant metrics. We still decided to go for the median because we wanted to reflect the characterization of general or expected behavior for each web page group. To improve readability, unless stated otherwise, it should be clear to the reader that all results presented in the paper correspond to median values. For ease of analysis, the minimum and maximum values per column are highlighted in red and green, respectively. Each of the groups is manually labeled, based on the most dominant feature (e.g., most downloaded type of content)

TABLE 3. Web classification results for PC-first dataset (desktop PC browsing without browser cache). Cluster attributes correspond to the median values, for each cluster and for each different dimension.

			Cluster attributes								Content metrics			Time metrics (s)			
Index	**Groups**	**# Webs**	**CNX**	**Images**	**JS**	**HTML**	**Font**	**CSS**	**Flash**	**Video**	**Others**	**FLB (KB)**	**ADs**	**ADB (KB)**	**SR**	**FI**	**SI**	**FLT**
1	HTML	**5**	**6**	**0.02**	**0**	**0.63**	**0**	**0**	0	0	**0**	**145**	**0**	**0**	**0.66**	**0.56**	**0.66**	**0.89**
2	Video	20	84	0.14	0.09	**0.02**	0.02	0.01	0	**0.58**	0	**5716**	9	128	1.50	1.79	3.10	7.76
3	Images	123	45	**0.79**	0.12	0.02	0.01	0.01	0	0	0	2551	2	41	1.74	2.16	2.97	5.10
4	JS	**143**	30	0.10	**0.65**	0.07	0.01	0.03	0	0	0	765	4	21	1.35	1.48	1.79	3.5
5	CNX	32	**265**	0.43	0.36	0.04	**0.04**	0.01	0	0	**0.01**	3253	**25**	**369**	**1.80**	**3.11**	**3.52**	**13.09**
6	sIMG	124	65	0.46	0.35	0.04	0.04	**0.05**	0	0	0	1838	6	56	1.50	2.18	2.65	6.14

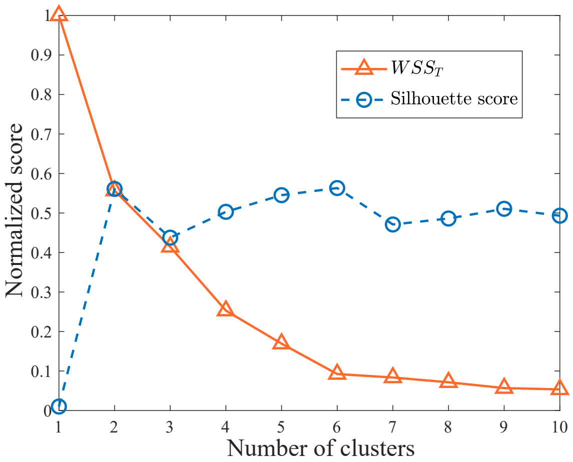


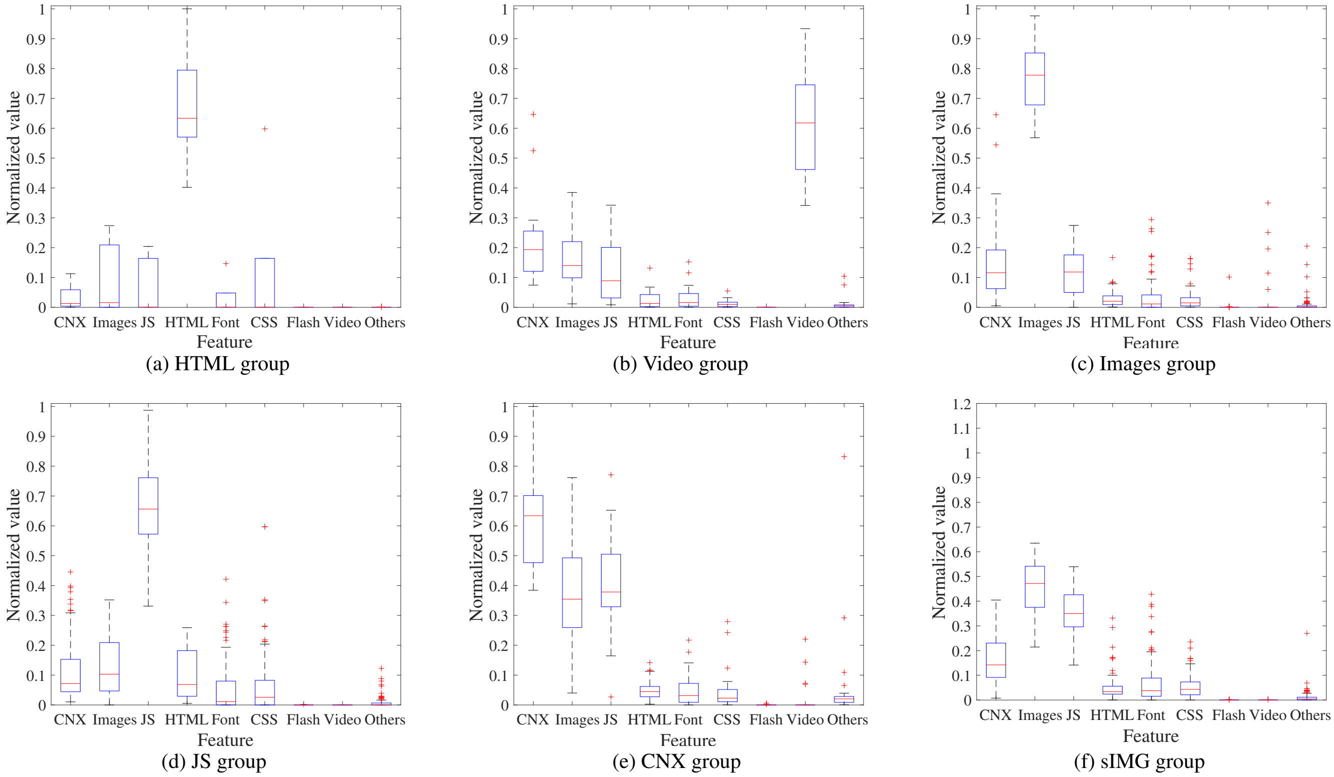
FIGURE 4. Clustering performance and identification of optimal number of classes. The silhouette score is highest for $k = 2$ and $k = 6$, with the latter showing a much more compact structure.

in the data points of each cluster. For example, if image content is the most relevant/most present type of content in the web pages of a certain cluster, it makes sense to refer to this cluster as the Images Group. Fig. 5 additionally shows the normalized features per cluster as boxplots, to ease the interpretation of the labeling. Note that, here, outliers (i.e., red crosses) correspond to values which are higher than (1.5 x 75% percentile). For the sake of comparison among groups, cluster attributes in Fig. 5 are normalized. Thus median values are not directly comparable to those reported in Table 3.

Group 1 is referred to as **HTML**, group 2 as **Video**, group 3 as **Images**, group 4 as **JavaScript (JS)**, group 5 as **CNX**, and group 6 as **Styled Images (sIMG)**, the latter based on the strong images component (about 50% of the content) and the dominance in CSS as compared to the other groups. The number of web pages clustered together is also reported, showing an uneven distribution across groups. For example, while Images, JS, and sIMG groups have around 130 pages, Video and CNX groups have a few tens of pages, and the HTML group only five pages. Again, this points out the dominance of Images and JS contents in modern web pages, with more traditional only-HTML pages being marginally used today, at least within the most popular sites.

For characterization and QoE assessment purposes, the table also indicates additional content metrics reflecting downloaded volumes and advertisement contents (FLB, ADs, and ADB), as well as loading-time metrics (SR, FI, SI, and FLT). To ease interpretation, Fig. 6 depicts the QoE-related metrics (loading times), along with the downloaded volume (FLB) and the number of connections to support domains (CNX). Next, we provide a deeper assessment of each web group. As reference for QoE assessment, the SI thresholds recommended as target for excellent web performance vary between 1 second (desktop) and 3 seconds (mobile), with an overall accepted target around 2 seconds for good Web QoE [25].

Fig. 6.(a) reports the (median) SI values per group, which serves as the central Web QoE-relevant metric [25]. **The first interesting observation is that, indeed, web content has a significant impact on Web QoE** (recall that web groups are identified exclusively by content-based descriptors). There are significant QoE differences among the different web groups, as reflected by median SI values varying between around half a second (excellent Web QoE) and up to 3.5 seconds (average to poor Web QoE) [25]. To better understand the influence of web contents on QoE, Fig. 7 reports the (rank) correlation (RC) between Speed Index and selected content S-KPIs, including FLB, CNX, Images, JS, and CSS, for all web pages (first column), as well as per web groups. Content has a moderate, but non-negligible, correlation to Web QoE, with page size (RC = 0.62), number of connections (RC = 0.61), and relative image volume (RC4 = 0.53) as the most relevant content characteristics for loading performance. While there are particular exceptions per web page class linked to the different mix of contents, both FLB and CNX are strong indicators of Web QoE. As expected, the smaller the page and the less connections to



(a) HTML group (b) Video group (c) Images group
(d) JS group (e) CNX group (f) sIMG group

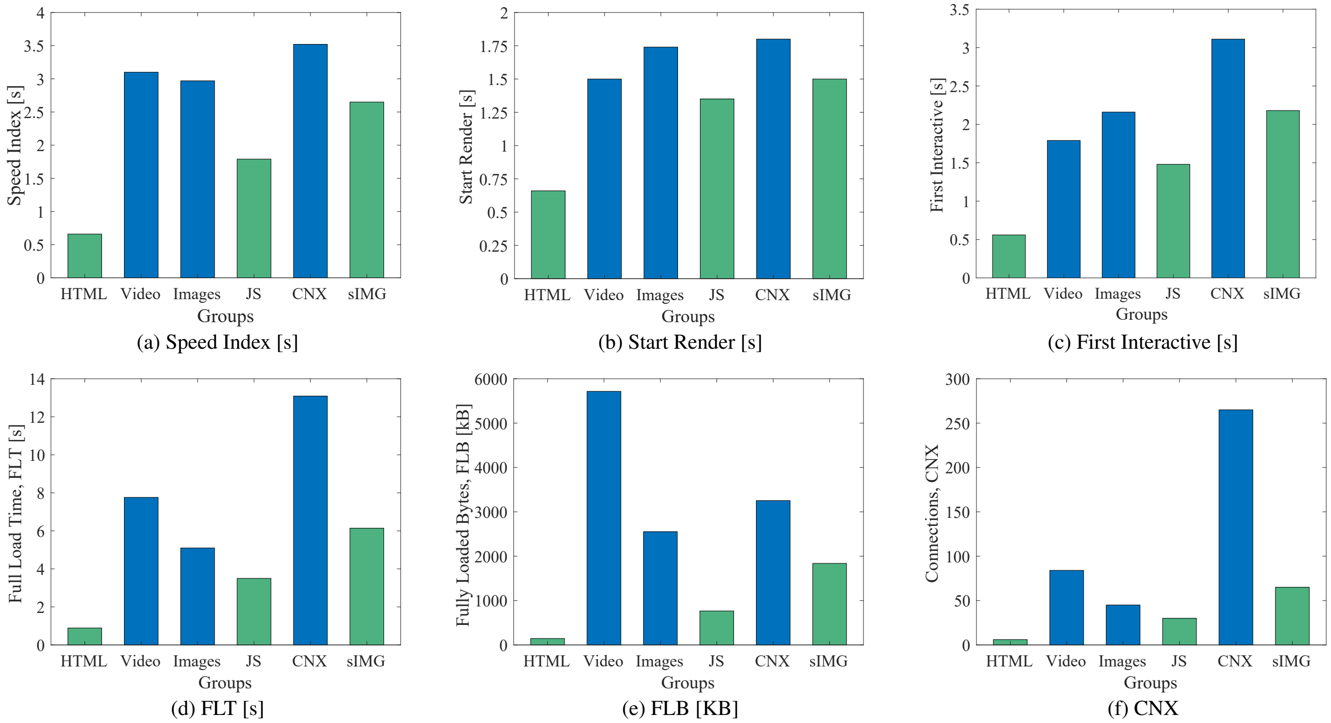
FIGURE 5. Normalized clustering metrics per group. For the sake of comparison among groups, cluster attributes are normalized, thus median values are not directly comparable to those reported in Table 3.

support domains, the faster the loading and the better the QoE. As shown in Fig. 6.(a), HTML, JS and sIMG web pages present the best Web QoE. Not surprisingly, pages in these groups are smaller (cf. Fig. 6.(e)) and have less connections (Fig. 6.(f)). Let us look into the specifics of each web group.

Group 1 (**HTML**) consists of only five web pages, and it is characterized by a MIME data type mostly concentrated in HTML, with few images, JS, and a small number of external connections. Web pages in this group are simple, lightweight (FLB $=$ 145 KB), they load very quickly (SI $=$ 0.66 s and FLT $=$ 0.89 s), and have very few connections to other domains (CNX $=$ 6), none of which corresponds to advertisement domains. Some examples of the web pages in this group include https://godaddy.com and https://friv.com. As expected, given their simplicity and low volumes, web pages in the HTML group have the best loading performance and best QoE-related indicators as compared to the other groups. Note for example that SR, FI, and SI values are basically the same, meaning that this class of pages immediately displays fully above the fold once the content has been rendered. The median value for FI is slightly smaller than for SR, which in theory should not be the case (i.e., SR $\leq$ FI $\leq$ SI $\leq$ FLT). As the measurement tool WPT has a reported reduced precision to measure FI values below one second, so we can consider them as equal.

Group 2 (**Video**), with 20 web pages, comprises sites using video content. These web pages are the heaviest, with a median size FLB $=$ 5716 KB. They have the second highest number of connections to other subdomains (CNX $=$ 84), a few of them corresponding to advertisement domains (ADs $=$ 9), and a significantly high loading time (FLT $=$ 7.76 s). They show reasonably good SR $=$ 1.50 s and FI $=$ 1.79 s starting loading times, but their median Speed Index has the second worst value SI $=$ 3.10 s in the dataset, probably due the high volumes of downloaded content, and from advertisement domains (ADB $=$ 128 KB, the second highest value). Some examples of web pages in this group include https://netflix.com and https://youtube.com.

Group 3 (**Images**) consists of 123 web pages where images prevail. Images are compressed resources (JPEG, GIF, and PNG are the standard file formats) downloaded from different servers and inserted into the HTML web page code. Some examples are https://mercadolibre.com.ar and https://amazon.com. This group includes relatively heavier web pages with less connections to other domains (CNX $=$ 45), very few corresponding to advertisement domains (ADs $=$ 2). Regarding QoE-related indicators, the SR time is usually higher than in the other groups, probably linked not only to the larger content volumes, but also due to the image decompression process which has to be done at the browser side during rendering. The SI is high (about 3 seconds) and comparable to the video web pages. While images take between 2 to 3 seconds to appear on the screen, the fully loading time is rather high, taking about 5 seconds



(a) Speed Index [s]

(b) Start Render [s]

(c) First Interactive [s]

(d) FLT [s]

(e) FLB [KB]

(f) CNX

FIGURE 6. Web performance and page characteristics for the six web-page groups identified by WebCLUST. Data correspond to desktop measurements, without using browser caching (first view results). Results report median values per group.

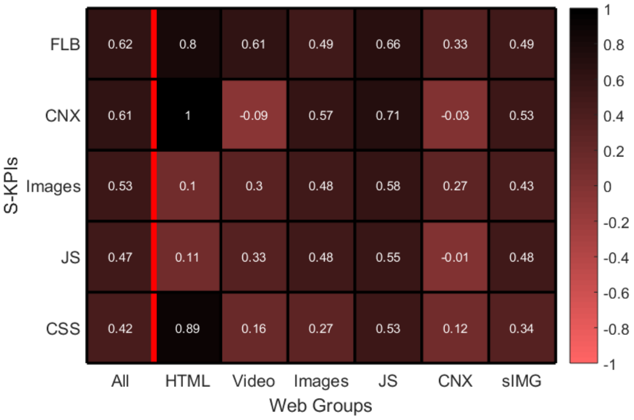


FIGURE 7. Rank correlation between Speed Index and selected content S-KPIs, including FLB, CNX, Images, JS, and CSS, for all web pages (first column), as well as per web group.

for the process to conclude.

Group 4 (**JS**) comprises 143 web pages with a highly dominant percentage of JS content. JS code is inserted into HTML to instruct the browser to execute actions for dynamic and interactive pages. Examples of pages in this group are https://twitter.com and https://pixnet.net. This group presents very good QoE indicators and is ranked second in terms of web performance (below the HTML group), probably due to the relatively lightweight content and the small number of connections to other subdomains (CNX $=$ 30). Even if FLT $=$ 3.5 s, the user can notice screen activity after only SR $=$ 1.35 s, and see the visible contents before 2 seconds.

Group 5 (**CNX**), with 32 web pages, contains highly connected sites with a large number of connections to other domains. It is worth noting that many of these external connections correspond to advertisement domains (ADs $=$ 25). Webs with embedded advertisements are often designed as dynamic pages, and their content is generated on the fly by the server, depending on the specific user and time (e.g., due to targeted advertisement). This might explain the extremely high loading times, with FLT $=$ 13.09 s, almost 15 times higher than the HTML group. CNX web pages undergo the worst QoE of all groups, with a SI value higher than 3.5 seconds. Web pages in the CNX group have also an important fraction of their contents as Images and JS, suggesting a more complex page structure, that combined with the high number of external connections, slows down the page loading process. Examples of web pages in this group are https://speedtest.net and https://accuweather.com.

Finally, group 6 (**sIMG**) consists of 124 web pages where Images and CSS files have considerable size. CSS files are used to describe the style of the website (e.g., the layout and variations for different devices or screen sizes). Pages in this group also stand out because they contain JS and Font files. All these features justify the name of the group, styled images. Examples of pages in this group include https://foxnews.com and https://wikipedia.org. The sIMG group contains rather lightweight web pages (FLB $=$ 1838 KB), and their loading performance is comparable or slightly better than for

TABLE 4. **Number of websites per group for the four different datasets, using the PC-first dataset as baseline for centroids' computation, and a minimum-distance classifier to assign web pages to groups.**

TEST	HTML	Video	Images	JS	CNX	sIMG
PC-First	5	20	123	**143**	32	124
PC-Repeat	**200**	14	79	61	32	61
Mobile-First	7	12	115	**175**	13	125
Mobile-Repeat	**220**	9	83	60	22	53

those in the Images group, which have a bigger size (FLB $= 2551$ KB), but a smaller number of connections to other subdomains (CNX $= 45$ vs. CNX $= 65$). Interestingly, while the SR time is lower than for the Images group, it takes longer to achieve interactivity status, due to the more complex content processing and rendering. Indeed, websites having HTML, CSS, and JS contents usually have a slower rendering process, due to the time spent by the browser engine to decode and arrange the contents in the so-called render tree [56].

V. WEBCLUST FOR MULTI-DEVICE AND CACHING

We devote the last part of the study to understand the impact of the browsing technology used at the end terminal on the page grouping and characterization through WebCLUST. In particular, we focus on (i) the impact of the device type (i.e.,. mobile browsing with a smartphone), and (ii) the impact of in-browser caching. In both cases, the volume of data downloaded during a web session reduces as compared to the baseline scenario (i.e., desktop PC browsing without caching). Indeed, smartphone terminals often access a simpler version of web pages, optimized to their smaller screen size and usability, while enabling in-browser caching reduces the number of downloaded static contents, such as images.

To this end, three additional datasets are generated, following the same methodology and targeting the same web pages explored in the **PC-First** dataset. These datasets include the **PC-Repeat** dataset, which corresponds to the activation of in-browser caching in the desktop PC web browsing measurements (repeat view results in WPT), and two datasets using a smartphone as end-device, considering either the results from the first visit to the pages (**Mobile-First** dataset), or the usage of in-browser caching (**Mobile-Repeat** dataset). The particular question we try to answer is how the web page groups identified by WebCLUST in the baseline scenario vary when relying on caching and/or mobile browsing. For doing so, we now pose the WebCLUST analysis as a classification problem, where we take the six web groups identified by WebCLUST in the PC-First dataset as the classes, and the distance to their centroids as classification rule, to construct a minimum-distance classifier. In a nutshell, given a web page i to classify, we assign it to group ID $j = 1..6$ such that

$$\mathrm{ID}(w_i) = \underset{j=1..6}{\arg\min}\, d\left(w_i, c_j^{\text{baseline}}\right), \qquad (3)$$

where w_i is a vector with the median values of the nine clustering S-KPIs (cf. Table 1) across the six tests performed per web page i (cf. Section IV-A), and c_j^{baseline} corresponds to the centroid of group j, as identified by WebCLUST in the baseline (i.e., using the PC-First dataset).

Table 4 shows the classification results obtained in the different datasets with this minimum-distance classifier, in terms of number of web pages assigned to each group. As a first general observation (and as expected), note how in-browser caching (i.e., PC-Repeat and Mobile-Repeat) makes a major share of the web pages to be shifted from Images, JS and sIMG groups to HTML group. The change of device type has also and impact as compared to the baseline, but shifts are less relevant, with a decrease in the number of pages in the Video and CNX groups, and an increase in the JS group size. To better understand these changes, we break down cluster shifts as compared to the baseline through Confusion Matrices (CMs). CMs are used in general classification problems to visualize the performance of a classifier as compared to the ground truth. Here we use them slightly differently, as the target is to understand how web pages are re-assigned to the different web classes when using caching and/or mobile browsing, using the baseline partitioning as reference or ground truth. Discussion is presented next.

1) Impact of in-browser caching for desktop PC

In-browser caching happens by default in every browsing session, to accelerate the loading of a page in future visits. Static contents are prone to caching, including images, files, and even scripts that are stored in the browser's cache the first time a website is visited (or updated in subsequent visits, upon content expiration). In-browser caching makes subsequent page loading sessions faster, as the overall volumes to download are smaller. Fig. 8 shows the empirical distribution of the Fully Loaded Bytes (FLB) for the first visit (PC-First) and the repeated visit (PC-Repeat) of the targeted web pages. As expected, there is a significant reduction in the FLB values when relying on cached contents. For example, while more than 70% of the pages have a FLB value above 1 MB for non-cached contents, this fraction drops to only 15% of the pages when using caching.

TABLE 5. Confusion matrix showing re-assignation of web pages to web groups for the PC-Repeat dataset (caching).

PC-First / PC-Repeat	HTML	Video	Images	JS	CNX	sIMG	Total
HTML	**4**	0	0	0	0	1	5
Video	4	**9**	2	3		2	20
Images	**53**	2	44	11	2	11	123
JS	**92**	0	8	20	6	17	143
CNX	4	2	3	0	**20**	3	32
sIMG	**43**	1	22	27	4	27	124
Total	200	14	79	61	32	61	**447**

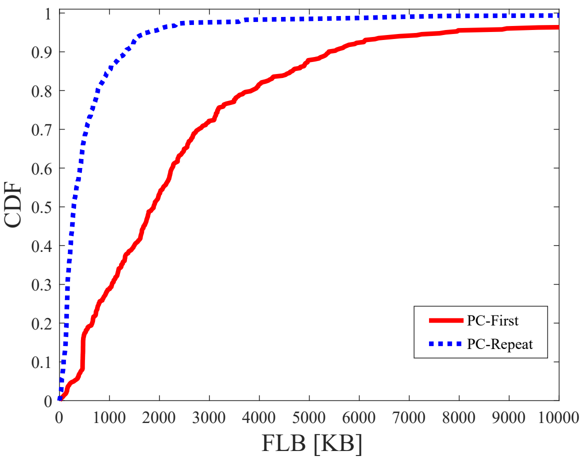


FIGURE 8. Distribution of FLB for a desktop PC browser with and without caching (first page visit vs. second page visit in WPT).

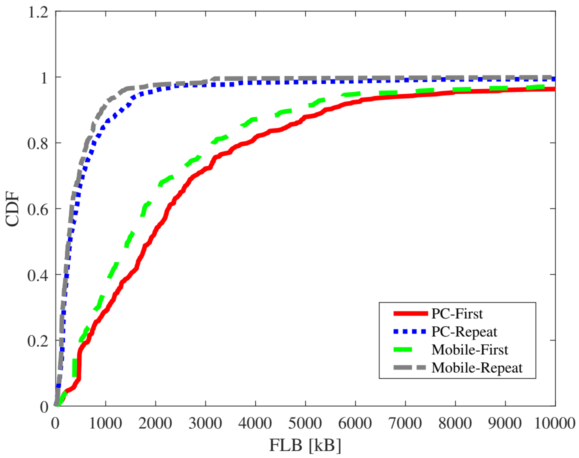


FIGURE 9. Distribution of FLB for desktop PC and mobile browser with and without caching (first page visit vs. second page visit in WPT).

To understand the changes in web page groups compared to the baseline, Table 5 presents the CM obtained when applying the minimum-distance classifier (3) to the PC-Repeat dataset. An important fraction of web pages under caching-based browsing are now classified as HTML, as static contents (mostly images and JS contents) are no longer part of the loading process. In fact, now nearly half of the pages (200/447) are classified as HTML according to the baseline, whereas only six pages were in this category in the PC-First results.

Overall, four out of the six groups mostly move to HTML due to the changes in contents' distribution realized by caching. Video and CNX groups remain more stable, as either contents in both groups are less prone to local caching (e.g., video and other dynamic contents, such as ADs), or their structure is more independent from caching (e.g., number of support domains).

2) Impact of mobile browsing

To evaluate the changes with respect to the baseline when browsing the web pages in a mobile device, we consider both the non-caching, first visit (Mobile-First) scenario and the caching, repeated visit (Mobile-Repeat) scenario. Fig. 9 shows the empirical distribution of the FLB values for the desktop PC and mobile scenarios, with and without caching. Even if FLB values are higher for web pages browsed in desktop PC, the trend is very similar on both device types when moving from non-caching to caching, and downloaded bytes become very similar when using caching for both device types.

Table 6 shows the CM obtained for the Mobile-First dataset. The CM is mostly a diagonal matrix, as the number and distribution of web pages along web groups remains pretty much the same. Indeed, web pages mostly belong to the same groups for both device types. The noticeable exception is the CNX group, where almost two thirds of the pages are now assigned to other groups, with half of those assigned to the sIMG group. A closer analysis reveals that the 11 web pages re-assigned to sIMG actually reduce the number of connections to external subdomains by 34% as compared to the PC-First dataset.

Similarly, Table 7 shows the CM obtained for the Mobile-Repeat dataset, where in-browser caching is activated in the mobile device. Results are close to those obtained for the in-browser caching on desktop PC, with a strong re-assignment of web pages from the Images, JS, and sIMG groups to the HTML group. The variation in the CNX group is again rather different in mobile, with a stronger shift from CNX to HTML as compared to the desktop PC browsing scenario.

TABLE 6. Confusion matrix showing re-assignation of web pages to web groups for the Mobile-First dataset (mobile browsing).

PC-First / Mobile-First	HTML	Video	Images	JS	CNX	sIMG	Total
HTML	**4**	0	0	1	0	0	5
Video	0	**8**	1	4	1	6	20
Images	2	1	**83**	12	0	25	123
JS	1	0	1	**126**	0	15	143
CNX	0	2	3	5	**11**	11	32
sIMG	0	1	27	27	1	**68**	124
Total	7	12	115	175	13	125	**447**

TABLE 7. Confusion matrix showing re-assignation of web pages to web groups for the Mobile-Repeat dataset (mobile browsing + caching).

PC-First / Mobile-Repeat	HTML	Video	Images	JS	CNX	sIMG	Total
HTML	**3**	0	0	1	0	1	5
Video	**6**	3	5	2	1	3	20
Images	**61**	3	33	10	1	15	123
JS	**97**	1	9	21	5	10	143
CNX	7	1	6	3	**10**	5	32
sIMG	**46**	1	30	23	5	19	124
Total	220	9	83	60	22	53	**447**

3) Impact of in-browser caching and mobile browsing on web performance

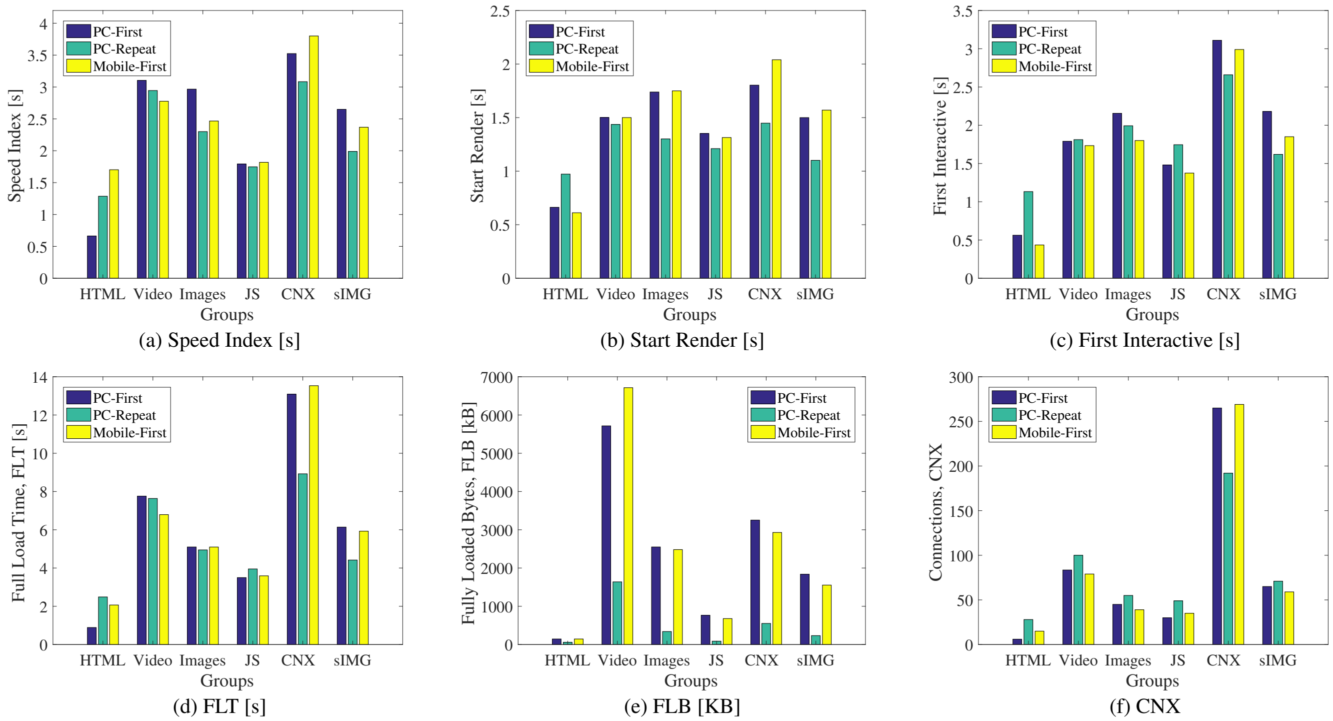
How does caching and mobile browsing impact web performance and QoE? This is the final aspect we study. To this end, similarly to Fig. 6, we compare the QoE-related metrics (loading times), along with the downloaded volume (FLB) and the number of connections to support domains (CNX), among the PC-First (baseline), the PC-Repeat (caching), and the Mobile-First (mobile browsing) datasets, for the web groups previously identified, using the minimum-distance classifier (3). Fig. 10 depicts the results, corresponding to median values per group. It is worth noting that the set of web pages in each group is not the same for the different datasets and classes (cf. Table 4), which should be considered in the analysis.

In-browser caching has a significant impact on Web QoE, as reflected by the reduction on all the page loading time metrics. Fig. 10.(b) shows the SR, which is the time for something to appear in the screen. As expected, in-browser caching reduces this time considerably, especially for Images, sIMG, and CNX, with a SR about 20% faster. This is due to the reduction of downloaded data, as depicted in Fig. 10.(e). These three groups reduce the size of their web pages by more than 80%. This is also the main reason for the strong shift of web pages to the HTML group (cf. Table 4). As a consequence of this shift, all loading time metrics for the HTML group actually increase for the caching scenario (PC-Repeat); in fact, the HTML group in PC-Repeat contains 200 web pages, as compared to the 5 pages in the HTML group for PC-First, which significantly increases the heterogeneity of the group, with an impact in the median performance values. Note also how the median number of connections significantly increases for this group, changing from 6 to 28 (cf. Fig. 10.(f)), which has a direct connection to the worse performance, as noted by the correlations shown in Fig. 7. When it comes to the mobile browsing scenario, there are no significant observable differences in terms of SR times, with the only exception of the CNX group, where the SR increases by about 12%.

Fig. 10.(c) depicts the FI, which is the time for the website to be interactive for the first time. In this case, the mobile browsing consistently shows better performance than desktop PC, most probably due to the smaller page sizes, as reflected in Fig. 9. Interestingly, in-browser caching does not always result in an advantage for the FI compared to the non-caching browsing, as observed for example in the JS group.

Fig. 10.(a) shows the SI, which reflects the average time for the visible part of the page to complete. In-browser caching improves the SI due to smaller data sizes to download, with reductions ranging from 2.5% in JS to about 25% in Images/sIMG. Mobile browsing also performs slightly better than desktop PC, with reductions of more than 10% in the SI for Images, sIMG, and Video groups. However, for those groups where the number of connections actually increases (JS and CNX, cf. Fig. 10.(f)), desktop PC outperforms mobile, but the differences are limited.

Finally, Fig. 10.(d) shows the FLT, representing the end of network activity when downloading the web page. Caching shows significant improvements for the CNX and sIMG groups, but there are no noticeable differences for the rest of the groups. When considering mobile browsing, the only noticeable difference in FLT happens for the Video group, which is surprising given the increase in the FLB values for the Video group in mobile (cf. Fig. 10.(f)). Again here, the



(a) Speed Index [s]

(b) Start Render [s]

(c) First Interactive [s]

(d) FLT [s]

(e) FLB [KB]

(f) CNX

FIGURE 10. Impact of in-browser caching (PC-Repeat) and mobile browsing (Mobile-First) on web performance indicators. Results correspond to median values per group. In-browser caching has a strong impact on the Web QoE, as reflected by the reduction on the Speed Index metric.

number of connections seems to play an important role.

VI. CONCLUDING REMARKS

The type and distribution of contents of a web page play a key role on the QoE perceived by the user. We have introduced WebCLUST, an unsupervised web characterization approach to group web pages based on content features affecting the QoE as perceived by the end-user. The classification method relies on clustering techniques, using as input the MIME content breakdown of a web page and its TCP connections to external subdomains. Through WebCLUST, we have identified six different families of web pages in the top-500, most popular web pages of the Internet.

Our analysis has shown that the different web groups or families of web pages identified by WebCLUST realize significant different end-user experience in terms of loading times. Indeed, we found significant QoE differences among the different web groups, as reflected by median Speed Index values varying between around half a second (excellent Web QoE) and up to 3.5 seconds (average to poor Web QoE) [25].

We have also studied the influence of in-browser caching and browsing device type, both in the identification of web groups through WebCLUST, as well as in the associated web group QoE. In-browser caching improves Web QoE, mainly driven by the reduced volume of data to be downloaded; caching speeds up the page rendering process and reduces the time to page interactivity, especially for those web pages with strong CSS and images contents, and for those with a larger number of connections to external resources. Mobile browsing has a less relevant impact, mostly improving loading times and QoE for web pages with video contents.

The identification of families of web pages realizing markedly different Web QoE opens the door to better Web QoE modeling and monitoring approaches, as more targeted (per class) yet generic enough (not single-page oriented, but class oriented) Web QoE models could be constructed independently for each of the web page families. Future work will consider the development of parametric QoE models to estimate web performance from high-level network-layer metrics, specifically designed for the different types of web pages.

REFERENCES

[1] A. Banerjee, "Revolutionizing CEM With Subscriber-Centric Network Operations and QoE Strategy," Heavy Reading, Tech. Rep., Jul. 2014, Accessed: Mar. 2019. [Online]. Available: http://www.accantosystems.com/wp-content/uploads/2018/05/Heavy-Reading-Accanto.pdf

[2] P. Le Callet, S. Möller, A. Perkis et al., "Qualinet White Paper on Definitions of Quality of Experience," European Network on Quality of Experience in Multimedia Systems and Services, Tech. Rep., Mar. 2016, Accessed: Mar. 2019. [Online]. Available: http://www.qualinet.eu/index.php?option=com_content&view=article&id=45&Itemid=52

[3] E. Liotou, D. Tsolkas, N. Passas, and L. Merakos, "Quality of Experience Management in Mobile Cellular Networks: Key Issues and Design Challenges," *IEEE Communications Magazine*, vol. 53, no. 7, 2015.

[4] A. Raake, J. Gustafsson, S. Argyropoulos, M.-N. Garcia, D. Lindegren, G. Heikkilä, M. Pettersson, P. List, and B. Feiten, "IP-Based Mobile and Fixed Network Audiovisual Media Services," *IEEE Signal Processing Magazine*, vol. 28, no. 6, pp. 68–79, 2011.

[5] ITU-T, "P.800.1 Mean Opinion Score (MOS) terminology," Jul. 2016.

[6] F. Ricciato, "Traffic Monitoring and Analysis for the Optimization of a 3G Network," *IEEE Wireless Communications*, vol. 13, no. 6, pp. 42–49, 2006.

[7] Google. *HTTPS Encryption on the Web*. Accessed: Nov. 2020. [Online]. Available: https://transparencyreport.google.com/https/overview

[8] T. Hori and T. Ohtsuki, "QoE and Throughput Aware Radio Resource Allocation Algorithm in LTE Network with Users using Different Applications," in *Proceedings of 27th Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2016.

[9] P. Casas, M. Seufert, and R. Schatz, "YOUQMON: A System for On-line Monitoring of YouTube QoE in Operational 3G Networks," *ACM SIGMETRICS Perf. Eval. Rev. (PER)*, vol. 41, no. 2, 2013.

[10] A. Baer, P. Casas, A. D'Alconzo, P. Fiadino, L. Golab, M. Mellia, and E. Schikuta, "DBStream: A holistic approach to large-scale network traffic monitoring and analysis," *Computer Networks*, vol. 107, pp. 5–19, 2016.

[11] L. Skorin-Kapov, M. Varela, T. Hoßfeld, and K.-T. Chen, "A Survey of Emerging Concepts and Challenges for QoE Management of Multimedia Services," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 2, pp. 1–29, 2018.

[12] A. J. Garcia, M. Toril, P. Oliver, S. Luna-Ramirez, and R. Garcia, "Big Data Analytics for Automated QoE Management in Mobile Networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 91–97, 2019.

[13] D. N. da Hora, A. S. Asrese, V. Christophides, R. Teixeira, and D. Rossi, "Narrowing the Gap Between QoS Metrics and Web QoE Using Above-the-fold Metrics," in *International Conference on Passive and Active Network Measurement*. Springer, 2018, pp. 31–43.

[14] P. Meenan. *WebPageTest*. Accessed: Apr. 2021. [Online]. Available: https://www.webpagetest.org

[15] Z. Wang and A. Jain, "Navigation Timing Level 2," W3C Recommendation, Tech. Rep., Jan. 2020, Accessed: Jul. 2020. [Online]. Available: https://www.w3.org/TR/navigation-timing-2/

[16] I. Grigorik, J. Mann, and Z. Wang, "Performance Timeline Level 2," W3C Recommendation, Tech. Rep., Jan. 2020, Accessed: Jul. 2020. [Online]. Available: https://www.w3.org/TR/performance-timeline-2/

[17] E. Bocchi, L. De Cicco, and D. Rossi, "Measuring the Quality of Experience of Web users," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 4, pp. 8–13, 2016.

[18] S. Panicker, "Paint Timing Level 1," W3C Recommendation, Tech. Rep., Sep. 2017, Accessed: Jul. 2020. [Online]. Available: https://www.w3.org/TR/paint-timing/

[19] A. Saverimoutou, B. Mathieu, and S. Vaton, "Web Browsing Measurements: An Above-the-Fold Browser-Based Technique," in *Proceedings of 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1630–1635.

[20] J. Brutlag, Z. Abrams, and P. Meenan, "Above the fold time: Measuring web page performance visually," in *Proceedings of Velocity: Web Performance and Operations Conference*. O'Reilly, 2011.

[21] P. Meenan. *Speed Index*. Accessed: Apr. 2021. [Online]. Available: https://docs.webpagetest.org/metrics/speedindex/

[22] Q. Gao, P. Dey, and P. Ahammad, "Perceived Performance of Top Retail Webpages In the Wild: Insights from Large-scale Crowdsourcing of Above-the-Fold QoE," in *Proceedings of the Workshop on QoE-based Analysis and Management of Data Communication Networks*. ACM, 2017, pp. 13–18.

[23] M. Trevisan, I. Drago, and M. Mellia, "PAIN: A Passive Web performance indicator for ISPs," *Computer Networks*, vol. 149, pp. 115–126, 2019.

[24] A. Huet, A. Saverimoutou, Z. B. Houidi, H. Shi, S. Cai, J. Xu, B. Mathieu, and D. Rossi, "Revealing qoe of web users from encrypted network traffic," in *Proceedings of IFIP Networking Conference*, 2020, pp. 28–36.

[25] S. Wassermann, P. Casas, M. Seufert, N. Wehner, J. Schuler, and T. Hossfeld, "How Good is your Mobile (Web) Surfing? Speed Index Inference from Encrypted Traffic," in *Proceedings of Posters, Demos, and Student Research Competition (SIGCOMM)*. ACM, 2020, p. 616Ű639.

[26] ITU-T, "G.1030: Estimating End-to-End Performance in IP Networks for Data Application," Apr. 2014.

[27] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A Generic Quantitative Relationship between Quality of Experience and Quality of Service," *IEEE Network Magazine*, vol. 24, no. 2, pp. 36–41, 2010.

[28] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, "Understanding Website Complexity: Measurements, Metrics, and Implications," in *Proceedings of Conference on Internet Measurement*. ACM, 2011, pp. 313–328.

[29] S. Ihm and V. S. Pai, "Towards Understanding Modern Web Traffic," in *Proceedings of ACM Conference on Internet Measurement*. ACM, 2011, pp. 295–312.

[30] C.-C. Huang, S.-L. Chuang, and L.-F. Chien, "Using a Web-Based Categorization Approach to Generate Thematic Metadata from Texts," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 3, no. 3, pp. 190–212, 2004.

[31] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web." Stanford InfoLab, Tech. Rep., Nov. 1999, Accessed: Jul. 2020. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[32] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel, "A Semantic Approach to Contextual Advertising," in *Proceedings of the 30th annual international SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 559–566.

[33] X. Qi and B. D. Davison, "Web Page Classification: Features and Algorithms," *ACM Computing Surveys (CSUR)*, vol. 41, no. 2, pp. 12:1–12:31, 2009.

[34] M. Hashemi, "Web Page Classification: A Survey of Perspectives, Gaps, and Future Directions," *Multimedia Tools and Applications*, pp. 1–25, 2020.

[35] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[36] K. Golub and A. Ardö, "Importance of HTML Structural Elements and Metadata in Automated Subject Classification," in *Proceedings of the International Conference on Theory and Practice of Digital Libraries*. Springer, 2005, pp. 368–378.

[37] A. Ahmadi, M. Fotouhi, and M. Khaleghi, "Intelligent Classification of Web Pages using Contextual and Visual Features," Applied Soft Computing, vol. 11, no. 2, pp. 1638–1647, 2011.

[38] M. Kovacevic, M. Diligenti, M. Gori, and V. Milutinovic, "Visual Adjacency Multigraphs: Novel Approach for a Web Page Classification," in *Proceedings of the Workshop on Statistical Approaches to Web Mining (SAWM)*. ECML, 2004, pp. 38—49.

[39] X. Qi and B. D. Davison, "Knowing a Web Page by the Company It Keeps," in *Proceedings of the 15th International Conference on Information and knowledge management*. ACM, 2006, pp. 228–237.

[40] N. Gövert, M. Lalmas, and N. Fuhr, "A Probabilistic Description-Oriented Approach for Categorizing Web Documents," in *Proceedings of the 8th International Conference on Information and knowledge management*. ACM, 1999, pp. 475–482.

[41] R. Ghani, "Combining Labeled and Unlabeled data for Multiclass Text Categorization," in *Proceedings of the 19th International Conference on Machine Learning (ICML)*, vol. 2, 2002, pp. 187–194.

[42] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Yraining," in *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*. ACM, 1998, pp. 92–100.

[43] U. Hiwarale. *How the browser renders a web page? DOM, CSSOM, and Rendering*. [Online]. Available: https://medium.com/jspoint/how-the-browser-renders-a-web-page-dom-cssom-and-rendering

[44] Alexa. *500 Global Sites*. Accessed: Nov. 2020. [Online]. Available: https://www.alexa.com/topsites

[45] P. Meenan. *WebPageTest Documentation*. Accessed: Apr. 2021. [Online]. Available: https://docs.webpagetest.org/

[46] R. Viscomi, A. Davies, and M. Duran, Using WebPageTest: web performance testing for novices and power users. O'Reilly, 2015.

[47] R. Petnel. *The official easylist web site*. Accessed: Nov. 2020. [Online]. Available: https://easylist.to/easylist/easylist.txt

[48] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[49] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Prentice-Hall, 1988.

[50] J. Han, J. Pei, and M. Kamber, Data mining: concepts and techniques. Elsevier, 2011.

[51] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[52] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding," in *Proceedings of the 18th annual Symposium on Discrete Algorithms(SIAM)*, vol. 2. ACM, 2007, pp. 1027–1035.

[53] F. Iglesias, T. Zseby, and A. Zimek, "Absolute Cluster Validity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2096–2112, 2019.

[54] M. Zaki, Z. Obradovic, P. N. Tan, A. Banerjee, C. Kamath, and S. Parthasarathy, Proceedings of the 2014 SIAM International Conference on Data Mining. SIAM, 2014.

[55] M. Halkidi, M. Vazirgiannis, and C. Hennig, "Method-Independent Indices for Cluster Validation and Estimating the Number of Clusters," in Handbook of Cluster Analysis. Chapman, 2015, pp. 616–639.

[56] E. Ohans. *How browser rendering works - behind the scenes*. Accessed: Nov. 2020. [Online]. Available: https://blog.logrocket.com/how-browser-rendering-works-behind-the-scenes-6782b0e8fb10/