Perceived Performance of Top Retail Webpages In the Wild

Insights from Large-scale Crowdsourcing of Above-the-Fold QoE

Qingzhu Gao, Prasenjit Dey, Parvez Ahammad
Instart Logic Inc., 450 Lambert Ave, Palo Alto, CA.
parvez@ieee.org

This paper was originally published in the proceedings of the 2017 SIGCOMM Internet-QoE workshop.

ABSTRACT

Clearly, no one likes webpages with poor quality of experience (QoE). Being perceived as slow or fast is a key element in the overall perceived QoE of web applications. While extensive effort has been put into optimizing web applications (both in industry and academia), not a lot of work exists in characterizing what aspects of webpage loading process truly influence human end-user's perception of the *Speed* of a page. In this paper we present *SpeedPerception*, a large-scale web performance crowdsourcing framework focused on understanding the perceived loading performance of above-the-fold (ATF) webpage content. Our end goal is to create free open-source benchmarking datasets to advance the systematic analysis of how humans perceive webpage loading process.

In Phase-1 of our *SpeedPerception* study using Internet Retailer Top 500 (IR 500) websites, we found that commonly used navigation metrics such as *onLoad* and *Time To First Byte (TTFB)* fail (less than 60% match) to represent majority human perception when comparing the speed of two webpages. We present a simple 3-variable-based machine learning model that explains the majority end-user choices better (with $87 \pm 2\%$ accuracy). In addition, our results suggest that the time needed by end-users to evaluate relative perceived speed of webpage is far less than the time of its *visualComplete* event.

CCS Concepts

•**Information systems** → *Crowdsourcing; Web mining;*
•**Networks** → *Network performance modeling; Network performance analysis;*

Keywords

Quality of Experience; Web Performance; Above-the-Fold; Crowdsourcing; Perceived Speed; SpeedIndex; Perceptual SpeedIndex; onLoad; TTFB

1. INTRODUCTION

Bad quality of experience (QoE) is not just annoying to end-users, but also costly for website owners. A recent survey indicated that 49% of users will abandon a site after experiencing performance issues and that a 1-second delay meant a 7% reduction in conversions [11]. Page-level navigation metrics (e.g. onLoad, TTFB) are typically thought to not only reflect the speed of application-level delivery pipeline, but also have direct impact on the business for E-Commerce websites [5].

Improving *onLoad* (or other performance metrics) became a popular area of research in recent years [17, 12, 6, 19, 21]. Unfortunately, none of these techniques directly take real end-user experience into account. [10, 7] studied the underlying pattern of how web page complexity affects user experience. Recently, [18] took an important step towards estimating user's perception using eyeball-tracking technology. Inability to account for dynamic third-party contents and being specific to one website at a time are the main drawbacks for [18]. Google has put forth SpeedIndex [2] to replace traditional W3C metrics for measuring *above-the-fold* content performance. [8] introduced two SpeedIndex like metrics and described the correlation among them. They made a strong assumption that SpeedIndex alone is sufficient to account for end-user QoE without any end-user validation.

Varvello et al. [20] created an experiment that allowed users to look at the webpage loading frame-by-frame to determine the user perceived page load time (UPPLT). However, understanding the relationship between static measurements and user experience is non-trivial and hard to generalize across websites for a key reason: a user's perception of speed (when presented a single webpage in isolation) is subjective [14]. For example, consumers may tolerate a local small business site loading in 5 seconds, but they may not wait the same amount of time if they were browsing a top-tier popular webpage, since their expectations are different. We want to fill this critical gap by creating an A/B comparison framework and identify the most relevant metric(s) that explain user perception across a broad swath of commercial websites.

We built *SpeedPerception* [4] to crowdsource users' perceived performance of ATF webpage content. Our aim is to enable reproducible research that improves understanding of the web application QoE at scale. The advantage of our comparative paradigm is that we can resolve the scalability limitations of small grouped experiments [16]. Our belief (and hope) is that *SpeedPerception*-like benchmarking datasets can provide a quantitative basis to compare different algorithms and spur progress on helping quantify perceived webpage performance.

In the rest of this paper, we will first present Perceptual SpeedIndex (PSI) as a complementary metric to Google's SpeedIndex (SI). We follow this by explaining the design of *SpeedPerception* experiment along with data validation.

Our results suggest that although most single performance metrics fail to explain end-user visual perception of webpage speed accurately (less than 70% matching), modified SI and PSI serve as better metrics. Finally, we will present machine learning models that can achieve $87 \pm 2\%$ accuracy when predicting end-user perception of QoE in terms of relative speed.

2. METRICS FOR PERFORMANCE

In the past years, the web performance community settled on Page Load Time (*onLoad*) and a few other page-level navigation metrics for evaluating the performance (QoE) of a webpage. An often-repeated industrial dogma is that, given the same network conditions, content structure, and other controllable factors, the smaller these metrics, the better the QoE of a webpage (from an end-user perspective). Some recent studies have proposed a new set of metrics (such as byteIndex [8], and UPPLT [20]) for measuring end-user QoE.

While new metrics continue to be proposed, what is sorely missing is a systematic and reproducible way to link them to perceived QoE via real human feedback. *SpeedPerception* fills this gap for the webpage speed. Using *SpeedPerception*, we have attempted to quantify how accurate the current web performance metrics are in representing real user judgments of perceived speed.

The metrics included in the *SpeedPerception* study are mostly defined by WebPagetest[1] and W3C[2], plus one novel metric, Perceptual SpeedIndex (PSI). We also explore novel variations to SI and PSI through changing the end point of their integrals in a systematic way. We group these synthetic metrics into two categories: *non-visual* and *visual* metrics.

Non-Visual Metrics

Time to First Byte (TTFB) is the time from the initial navigation until the first byte is received by the browser. ***DOM Content Load Event End (DCLend)*** is time at which the Document Object Model (DOM) has been loaded by parsing the response. ***onLoad (Load Time or PLT)*** is measured as the time from the start of the initial navigation until the beginning of the window load event.

Visual Metrics

First Paint is a measurement reported by the browser itself about when it thinks it painted the first content. It is available from JavaScript and can be reported from the field.***Render Start (render)*** is the time from the start of the initial navigation until the first non-white content is painted. It is measured by capturing video of the page load and looking at each frame for the first time the browser displays something other than a blank page. It can only be measured in a lab and is generally the most accurate measurement for it. ***SpeedIndex (SI)*** is the average time at which visible parts of the page are displayed. It measures how quickly the page contents are visually populated (where lower numbers are better). ***Visual Complete*** is the time from the start of the initial navigation until there is no visual process within above-the-fold content.

[1]https://www.webpagetest.org/

[2]https://www.w3.org/TR/navigation-timing/

Perceptual SpeedIndex (PSI)

Google introduced the SI in 2012 as a metric to measure above-the-fold (ATF) visual QoE. The main idea was to use an aggregate function on the quickness of ATF visual completion process. The frame-to-frame visual progress in SI is computed from pixel-histogram comparisons. SI's histogram-based visual progress calculations can lead to some issues. In particular, visual jitter (caused by layout instability, weird ad behavior or carousel elements, etc.) cannot be captured by pixel-histogram based comparisons.

To address these issues, we proposed PSI as a complementary visual QoE metric to serve as a proxy for end-user perception[3]. Empirical experiments demonstrated that PSI and SI are linearly correlated with a strong correlation score of 0.91. Despite the strong correlation, SI and PSI are actually complementary to each other. While SI focuses on addressing how most of the webpage ATF content loads quickly, PSI focuses on addressing if the webpage ATF content loads quickly without visually noticeable jitter. Since October 2016, Google Chrome has officially incorporated PSI in their LightHouse project for measuring Progressive Web App performance [1].

The math behind SI and PSI is relatively straight-forward: aggregating the visual progress along a web page loading time-line. This can be expressed as:

$$Index_{end} = \int_{start}^{end} \Big(1 - \frac{\text{Visual Completeness}}{100}\Big) dt \qquad (1)$$

Visual Completeness in **Equation 1** is with respect to the last frame of a webpage loading process (video-based), and is expected to be 0 at $t = \text{start}$, and 100 at $t = \text{end}$. Instead of using mean pixel-histogram difference (MHD) to measure visual completeness, PSI uses Structural Similarity [22]. Both SI and PSI start the integration at time 0 and end at Visual Complete. In practice, truncating this integral at the right endpoint can make a big difference. For example, SI_{onLoad} would be defined as:

$$SI_{onLoad} = \int_{start}^{onLoad} \Big(1 - \frac{\text{Visual Completeness (MHD)}}{100}\Big) dt \qquad (2)$$

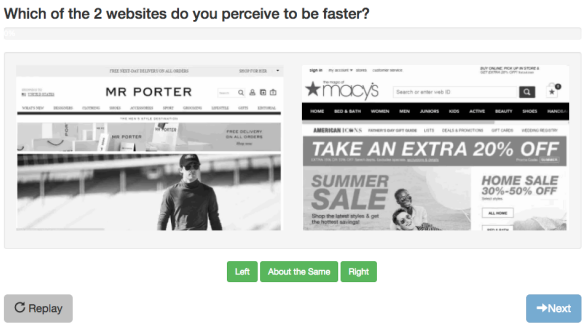
SI_{TTC} and PSI_{TTC} can similarly be defined with respect to Time to Click (TTC, Section 5.1):

$$PSI_{TTC} = \int_{start}^{TTC} \Big(1 - \frac{\text{Visual Completeness (SSIM)}}{100}\Big) dt \qquad (3)$$

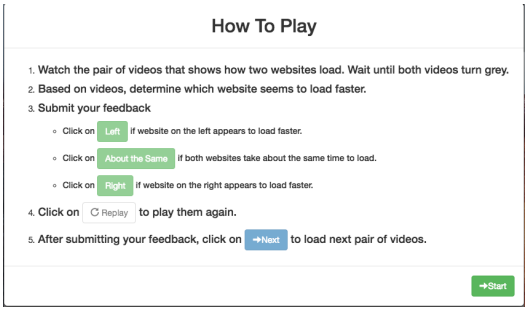
3. SPEEDPERCEPTION: PHASE-1

Consider the question: *How does one evaluate end-user perception of webpage speed?* As subjective as this sounds, perception of webpage speed is also relative (A vs B). Participants in our study were expected to answer a simple question, *which web page do you perceive to be faster?* The *SpeedPerception* framework enables the study of how human end-users perceive the "faster" page given two choices, where a pair of webpages' ATF loading process are displayed side-by-side to participants and their responses are recorded. To ensure identical visual experience, we used videos that

[3]http://www.parvez-ahammad.org/blog/perceptual-speed-index-psi-for-measuring-above-fold-visual-performance-of-webpages



(a) Side-by-side layout for each pair of videos.



(b) Instructions for participation.

Figure 1: SpeedPerception user interface

are generated from WebPagetest. These videos capture the above-the-fold content rendering process.

The web application is built with the Meteor.Js framework. We use MongoDB as the backend database. Both the application and database are hosted on (separate) cloud servers that can be scaled up or down dynamically. We have open-sourced our entire framework on Github[4] with the MIT license to facilitate reproducibility and ease of use.

3.1 Hypotheses and Workflow

Three testable experimental hypotheses were set up before the Phase-1 experiment:

HYPOTHESIS 1. ***No single existing performance metric can explain an end-user's perception of Speed with above 90% accuracy.***

HYPOTHESIS 2. ***Visual metrics perform better than non-visual metrics in predicting end-user's judgments on ATF performance.***

HYPOTHESIS 3. ***An end-user will not wait until* Visual Complete *event to make their choice.***

SpeedPerception work flow starts with collecting static measurements (HAR or *HTTP Archive* files) and videos from WebPagetest. Videos and HARs need to be processed so that we have a more structured data. Let us discuss some key characteristics of Video and HAR files.

HAR On an private instance (e.g., EC2 on Amazon) of WebPagetest, we launched a series of tests on Internet Retailer Top 500 (IR 500) URLs[5]. Test configurations are consistent across all test runs; we used "Chrome 50.0.2661.102," "Cable 10/5 Mbps," and "North California" machines. We ran 10 unique tests on each URL to have a fair sampling and reduce outliers caused by network hang-ups, traffic spikes and other factors. HAR file returned from each test contains the log of a web browser's interaction with the site when loading it.

Video A video of the ATF content is associated with each HAR file so that we can later map performance metrics to end-users' perception of "Speed." WebPagetest makes it possible to record a live webpage video while running a test. The cut-off was set at *Visual Complete* of these videos. Most videos have length (time) less than the actual "Fully Loaded' simply because there is more content to be loaded below users' view-ports.

[4]https://github.com/pdey/SpeedPerceptionApp
[5]https://www.digitalcommerce360.com/product/top-500/

3.2 Video Pair Selection

We then applied a group of 16 conditions to generate our target videos. In order to fairly compare between pairs, we need to pay attention to **Visual Complete** (VC). Controlling for the end point of a video, we only selected video pairs within 5% normalized difference. The normalized difference for VC (similar for other metrics), is calculated as:

$$VC_{diff} = \text{Diff}(VC_1, VC_2) = \frac{(VC_1 - VC_2)}{(VC_1 + VC_2) \cdot 0.5} \quad (4)$$

Within 5% *Visual Complete* difference, we subgroup them based on 4 conditions of SpeedIndex difference:

- $SI_{diff} \geq 10$;
- $1 \leq SI_{diff} < 10$;
- $-10 < SI_{diff} \leq -1$;
- $SI_{diff} \leq -10$

Within each SI difference condition, we again stratified each of them on 4 conditions of PSI difference. In total, we have $4 \cdot 4 = 16$ conditions of video pairs for our experiment. The reason we selected our video pairs based on SI and PSI for Phase-1 experiment is that we believed these are key QoE metrics to best express user perception. We also selected 5 fixed "honeypot pairs" (see section 4) in addition to 10 sets of 16 video pairs (total of 160 pairs + 5 honeypots) for the purpose of data validation. The final 160 pairs came from 115 unique webpages.

3.3 Platform Design

Figure 1a shows the UI of SpeedPerception. A typical session begins with an instruction banner (**Figure 1b**), which participants are expected to carefully read and follow these steps during the experiment. After clicking the **Start** button, a total number of 21 pairs of videos will be displayed sequentially, 16 assessment pairs + 5 honeypot pairs. For each instance, 2 videos start at the same time and play in parallel side by side. The parallel layout allows participants to better evaluate the webpage loading process. After watching a video pair, we provide 3 options for users to report their response. They can pick "Left" or "Right" if they perceived one of the webpages to load faster, otherwise they can pick "About the same" when unable to determine a "winning" candidate. The **Replay** button enables participants to replay the video pairs as many times as they want

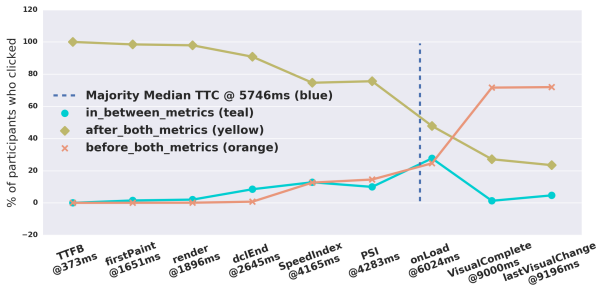


Figure 2: X-axis labels are different timing milestones (evenly spaced for ease of perusal) of a webpage loading. Y-axis is the percentage of users who pressed a button to indicate their choice.

until they feel comfortable to make a choice. After reviewing the pair, user will click on **Next** to proceed.

We randomly chose a set of video pairs out of 10 to present to our participants. We assigned a unique session ID to every single attempt from participants who clicked on the "Start" button, because one could have perceived each session differently given the randomized selection of video pairs.

4. ENGAGEMENT VALIDATION

One of the key challenges of crowdsourcing is to ensure the quality of the data. We promoted *SpeedPerception* experiment mostly through social media channels in the web performance community, as well as colleagues and friends. To mitigate contamination of the data, or malicious responses, we set up a series of validation mechanisms.

Instructions: Participants are given clear instructions as shown in **Figure 1b**. We expected people to follow these rules, except we did accept decisions before both videos reach *Visual Complete*.

Enforcement: During any stage of the experiment, participants cannot skip to the next video pair without providing a choice first. We want every video pair to be assessed — so the "Click" button was not made available until a choice had been made. Failure to complete all 21 pairs in a given session was considered as invalid session. Data points from invalid sessions were excluded from our analysis.

Honeypots: To prevent any malicious participant or bot, we used a "honeypot" mechanism. We inserted 5 video pairs at random order with known (very obvious) choices in each session of the study. One honeypot mistake is allowed per session, so that we only take responses that exceed 80% or more on these honeypots.

Majority Vote: For each of 160 (excluding 5 honeypots) video pairs, we aggregate across the participants' votes to formulate a "majority vote". For example, if a given pair has 10% votes for "Left", 30% votes for "Equal" and 60% votes for "Right", then we consider "Right" video as the majority human choice.

5. RESULTS & ANALYSIS

A total number of 5,400+ sessions were recorded in the *SpeedPerception* Phase-1 experiment, during a period of 2 months. 51% of the sessions successfully finished all 21 evaluations and passed the "honeypot' threshold. Accordingly, we have more than 40,000 valid votes that are nicely distributed over the 160 video pairs, with 250+ votes on each pair. Each video pair has votes split between 3 choices. 47% of the majority votes are on "Left" and 46% on "Right", while 7% of them fall in "Equal". Participants seem to have a strong preference to pick one of the two sides, instead of "Equal", when comparing between two webpages. It also indicates that our video pairs were fairly selected and displayed without any bias to one side.

Our primary goal is to examine the pattern of how page-level performance metrics reflect user perception. We calculated the normalized difference on each metric for every pair, using **Equation 4**. Then a "synthetic vote" was assigned for each video pair using the normalized difference of these metrics, where difference falls into ±5% will be assigned to "Equal." Difference smaller than -5% will considered as "Left," and larger than +5% as "Right." We also tried different thresholds using ±10% and ±1%, just to make sure that the choice of threshold doesn't have an undue impact on the results. We found that the overall results do not change significantly. For conciseness, we only include plots using ±5% in this paper.

5.1 Time to Click (TTC)

Time to Click (TTC) was measured from the start of the pairwise video display till the time when user clicks on a button to indicate their choice. TTC informs us when user believes they have sufficient information to make a judgment on perceived speed difference between the two webpages being shown.

Figure 2 shows the median position of TTC event, among votes that align with majority choice, as it relates to the median of synthetic metrics across the entire dataset. The position of synthetic metrics in this figure is evenly spaced to make visual inspection easier. The median TTC, for majority votes across all valid sessions, is at 5746ms. From **Phase-1** dataset [3], median TTC is close to median onLoad but not exactly the same. Phase-1 experiment didn't account for the small variability in human visuomotor response across various device types (say smartPhone to tablet). In future work, we plan fix this gap and build better estimates for TTC.

It is worth noticing that *almost* all participants voted *after* the *Render* event of both webpages. On the other hand, decision patterns start to shift between *Render* and *Visual Complete* events. Most participants waited until post-onLoad of at least one webpage video. Yet, very few waited until the *Visual Complete* event — thus confirming our hypothesis-3 (Section 3.1).

5.2 Matching Human Perception

To determine how well various synthetic metrics explain user perception, we computed both the fraction of the "synthetic votes" (associated with each metric) that match real user population's majority votes and the statistical correlation. **Figure 3** validates our first hypothesis, that there does not exist a single metric that can explain users perception above 90% accuracy. While onLoad only matches 55% of the majority vote, original SI (integrated up to Visual Complete) is even worse (53% match).

For each synthetic metric, we evaluate the Cramér's V (association measure for nominal random variables) [13] for the correlation between it's "synthetic votes" and user's majority pick. Chi-square test statistics was also used to determine

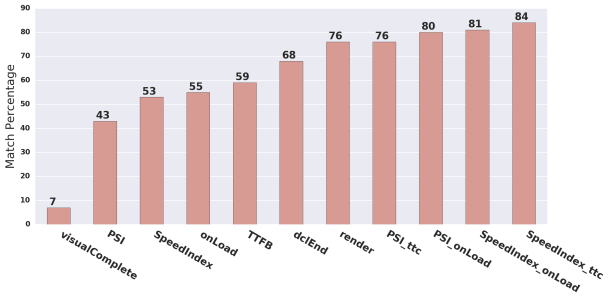


Figure 3: Rank ordered synthetic metrics matched to end-user votes on perceived speed. SI and PSI using onLoad and TTC as end points are included to demonstrate the significant improvement.

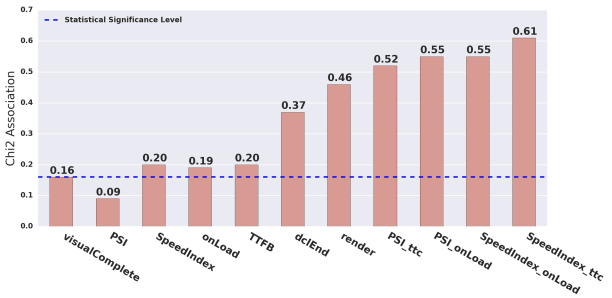


Figure 4: Chi-square correlation of each synthetic metrics' votes with majority votes (Note the dashed line denoting threshold for statistical significance).

the correlation power. The dashed blue line in **Figure 4** demonstrates a clear cut-off at the Chi-square significance level. Modified SpeedIndex and PSI yield "synthetic votes" that are highly correlated with majority pick and far beyond statistical significance.

Top 5 metrics that best match user perception on IR-500 webpage speed are SI_{TTC}, PSI_{TTC}, SI_{onLoad}, PSI_{onLoad} (i.e., *SI* and *PSI* integrated up to onLoad and TTC (majority) - **Equation 2**), and *Render*. This validates our hypothesis-2 about visual metrics (Section 3.1).

5.3 Joint ML model

Although we can explain 84% of the majority perception using SI_{TTC}, the percentage match score can only serve as an empirical observation. We also didn't want to solely rely on *SI*, knowing that it doesn't account for layout instability that significantly impacts human user QoE. To make our findings more actionable and robust, we tried simple supervised ML modeling approaches using all synthetic metrics to build a predictive model for user perception of speed. The predictive label of the model is three options that we provided for our participants. Model features are constructed from the normalized difference of each metric. All models were trained and tested using 10-fold cross validation. We show results from two state-of-art classification models: Random Forest (RFC) [9] and Gradient Boosting (GBC) [15]. Both are ensemble methods that build a classifier on a large number of smaller classifiers.

We can build a very expansive set of models using permutations of different metrics and features. Due to the limited

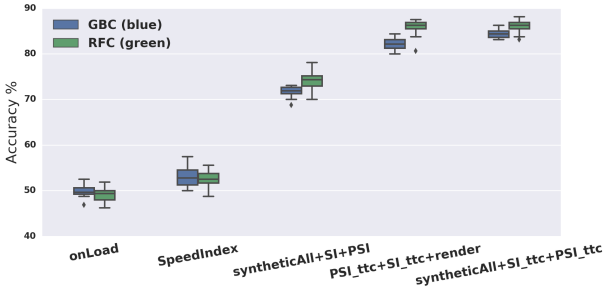


Figure 5: Box plot of ML models predicting majority vote of human users using different features.

space in this paper, we only show the illustrative results from (1) *onLoad*, (2) *SI*, (3) All synthetic metrics (noted as syntheticAll in the plot) + *SI* + *PSI*, (4) PSI_{TTC} + SI_{TTC} + *render*, (5) syntheticAll + PSI_{TTC} + SI_{TTC}. **Figure 5** provides us a clear view that a joint model of all synthetic metrics without any fine tuning can predict users' speed-based QoE choices at the accuracy level of 70% to 75%, which is an improvement compared to the onLoad or SpeedIndex based models. Despite such significant jump from 50%+ to 70%+, the *syntheticAll+SI+PSI* model uses original SI and PSI, which aren't the best (Section 5.2). We then fitted an alternative model replacing SI and PSI with our modified SI_{TTC} and PSI_{TTC}.The new model achieves an accuracy ranges from 87% to 90%. In fact, using only three visual metrics ($SpeedIndex_{TTC}$, PSI_{TTC} and *render*) can achieve almost the same level of accuracy as all metrics combined.

A lot of content relevant to visual QoE is rendered after onLoad — which explains why joint ML models using onLoad alone do poorly. On the other hand, many websites load visual jitter (such as carousals and pop-ups) after onLoad — which explains why visual-change aggregation beyond TTC is not that useful for SI or PSI calculations. We speculate that integrating visual change beyond TTC inserts noise into the computation, since a lot of visual jitter happens post-TTC.

6. SUMMARY AND DISCUSSION

We presented a novel large-scale crowdsourcing framework (*SpeedPerception* [4]) for A/B comparison of end-user QoE. Code for replicating the experimental framework as well as the crowdsourced benchmark data [3] are freely available, and serve as a useful basis for investigating better QoE metrics in future.

SpeedPerception Phase-1 study on IR 500 websites enabled us to analyze the associations between various web performance metrics and end-user judgments on perceived speed for ATF content. We introduced *Perceptual SpeedIndex* (PSI), and a few systematic variations of both SI and PSI — all of which serve as key indicators for real user perception of QoE. Phase-1 results showed that while no single performance metric reflected user judgments perfectly, SI_{TTC} and PSI_{TTC} appear to be able to match about 80% of the majority votes. Moreover, our joint machine learning models predict majority opinions above an accuracy of 85%.

It is now worth discussing some of the shortcomings of our Phase-1 study. ML models from our study, despite being

Figure 6: Word cloud of participants feedback.

much better than any individual metric, didn't fully match the real user decisions. One possibility for the observed error rates in ML models could be due to the presence of visual jitter (such as carousels, pop-ups, or advertisements) in the IR-500 ATF webpage videos. Several users left written feedback at the end of their SpeedPerception challenge sessions. One can see from **Figure 6** that pop-ups appear to have had a significant impact on the visual QoE of the real users who took our challenge. Since most of our video pairs both contained visual jitter elements, we couldn't truly evaluate the impact of visual jitter on the perceived speed choices that users made. While TTC appears to be a very good end-point for SpeedIndex and PSI calculations, it would be useful to develop a computational model to estimate TTC, or a better end point for these integral metrics. Demographical information was not collected in the Phase-1 of our study, which prevented us from analyzing how different groups of users could potentially perceive the speed of webpage loading differently. Furthermore, our experiment only studied user's QoE on single webpage, while web browsing typically involves sequences of page visits.

In future, we plan to investigate if our findings on human end-user perceived QoE vary from IR 500 ranked websites to Alexa 1000 ranked websites, as well as from desktop webpages to mobile webpages. While we showed improved performance using joint ML models, the models we used here don't lend themselves to easy interpretation.To address this, we plan to develop interpretable models to infer rules that relate webpage structure and page-level metrics to end-user QoE.

7. ACKNOWLEDGMENTS

We thank Estelle Weyl, Patrick Meenan (WebPagetest), and Ilya Grigorik for evangelizing *SpeedPerception* and generating significant user participation. We thank Paul Irish, Pierre-Marie Dartus, Shubhie Panicker, and Addy Osmani for independently evaluating and porting PSI to Google Chrome Lighthouse project.

8. REFERENCES

[1] Google Chrome Lighthouse Project https://github.com/googlechrome/lighthouse.

[2] SpeedIndex https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index.

[3] SpeedPerception Benchmark and Results https://github.com/pahammad/speedperception.

[4] SpeedPerception Experimental UI http://speedperception.com.

[5] The Very Real Performance Impact on Revenue http://blog.catchpoint.com/2017/01/06/performance-impact-revenue-real/.

[6] P. Ahammad, R. Gaunker, B. Kennedy, M. Reshadi, K. Kumar, A. Pathan, and H. Kolam. A flexible platform for QoE-driven delivery of image-rich web applications. In *IEEE ICME 2015*, pages 1–6, 2015.

[7] A. Balachandran, V. Aggarwal, E. Halepovic, J. Pang, S. Seshan, S. Venkataraman, and H. Yan. Modeling web quality-of-experience on cellular networks. In *ACM MobiCom 2014*, pages 213–224, 2014.

[8] E. Bocchi, L. De Cicco, and D. Rossi. Measuring the quality of experience of web users. *ACM SIGCOMM Computer Communication Review*, 46(4):8–13, 2016.

[9] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[10] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding website complexity: measurements, metrics, and implications. In *ACM SIGCOMM IMC*, pages 313–328, 2011.

[11] E. Carbery. Website Performance: The Need for Speed http://www.6smarketing.com/blog/website-performance-the-need-for-speed/.

[12] E. Cohen and H. Kaplan. Prefetching the means for document transfer: A new approach for reducing web latency. *Computer Networks*, 39(4):437–455, 2002.

[13] H. Cramér. *Mathematical Methods of Statistics*, volume 9. Princeton university press, 2016.

[14] S. Egger, P. Reichl, T. Hoßfeld, and R. Schatz. "Time is bandwidth"? Narrowing the gap between subjective time perception and quality of experience. In *IEEE ICC 2012*, pages 1325–1330, 2012.

[15] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[16] T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, and P. Tran-Gia. Best practices for QoE crowdtesting: QoE assessment with crowdsourcing. *IEEE Transactions on Multimedia*, 16(2):541–558, 2014.

[17] A. Iyengar, E. Nahum, A. Shaikh, and R. Tewari. Web caching, consistency, and content distribution. *The Practical Handbook of Internet Computing*, 2005.

[18] C. Kelton, J. Ryoo, A. Balasubramanian, and S. R. Das. Improving user perceived page load times using Gaze. In *USENIX NSDI 17*, pages 545–559.

[19] R. Netravali, A. Goyal, J. Mickens, and H. Balakrishnan. Polaris: Faster page loads using fine-grained dependency tracking. In *USENIX NSDI 2016*, 2016.

[20] M. Varvello, J. Blackburn, D. Naylor, and K. Papagiannaki. EYEORG: A platform for crowdsourcing web quality of experience measurements. In *ACM CoNEXT*, pages 399–412, 2016.

[21] X. S. Wang, A. Krishnamurthy, and D. Wetherall. Speeding up web page loads with Shandian. In *USENIX NSDI*, 2016.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

APPENDIX

The *SpeedPerception* Phase-1 benchmark [3] contains large-scale crowdsourced benchmark data about human end-user A/B judgements of perceived webpage performance (speed). All Phase-1 webpages were from Internet Retailer top-500 list, collected via Webpagetest (WPT) during June 2016, using Chrome browser in desktop rendering mode. This dataset and the associated analysis scripts (written using Python 2.0 & Jupyter 4.2.1) are made public under the MIT license.