2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)

Accessibility Issues in Ad-Driven Web Applications

Abdul Haddi Amjad
Computer Science
Virginia Tech
Blacksburg, USA
hadiamjad@vt.edu

Muhammad Danish
Computer Science
Virginia Tech
Blacksburg, USA
mdanish@vt.edu

Bless Jah
Computer Science
Virginia Tech
Blacksburg, USA
blessj@vt.edu

Muhammad Ali Gulzar
Computer Science
Virginia Tech
Blacksburg, USA
gulzar@cs.vt.edu

***Abstract*—Website accessibility is essential for inclusiveness and regulatory compliance. Although third-party advertisements (ads) are a vital revenue source for free web services, they introduce significant accessibility challenges. Leasing a website's space to ad-serving technologies, like DoubleClick, results in developers losing control over ad content accessibility. Even on highly accessible websites, third-party ads can undermine adherence to Web Content Accessibility Guidelines (WCAG). We conduct the first-of-its-kind large-scale investigation of 430K website elements, including nearly 100K ad elements, to understand the accessibility of ads on websites. We seek to understand the prevalence of inaccessible ads and their overall impact on the accessibility of websites. Our findings show that 67% of websites experience increased accessibility violations due to ads, with common violations including Focus Visible (WCAG 2.4.7) and On Input (WCAG 3.2.2). Popular ad-serving technologies like Taboola, DoubleClick, and RevContent often serve ads that fail to comply with WCAG standards. Even when ads are WCAG compliant, 27% of them have alternative text in ad images that misrepresents information, potentially deceiving users. Manual inspection of a sample of these misleading ads revealed that user-identifiable data is collected on 94% of websites through interactions, such as hovering. Since users with disabilities often rely on tools like screen readers that require hover events to access website content, they have no choice but to compromise their privacy to navigate website ads. Based on our findings, we further dissect the root cause of these violations and provide design guidelines to both website developers and ad-serving technologies to achieve WCAG-compliant ad integration.**

***Index Terms*—accessibility, web, ads, privacy, web development**

I. Introduction

Approximately 15% of the global population has some form of disability, and their access to the Internet is largely dependent on website accessibility. Many of these users rely on assistive technologies such as screen readers, which require highly accessible websites (*e.g.,* properly labeled images) to function effectively [1]. The Web Content Accessibility Guidelines (WCAG) [2], led by the World Wide Web Consortium (W3C), have driven efforts to improve websites' accessibility. As a result, there are now numerous tools and services to test and enhance the accessibility of websites [3]–[6]. Although website accessibility is gradually improving due to these tools, the accessibility of website advertisements (ads) remains largely unexamined.

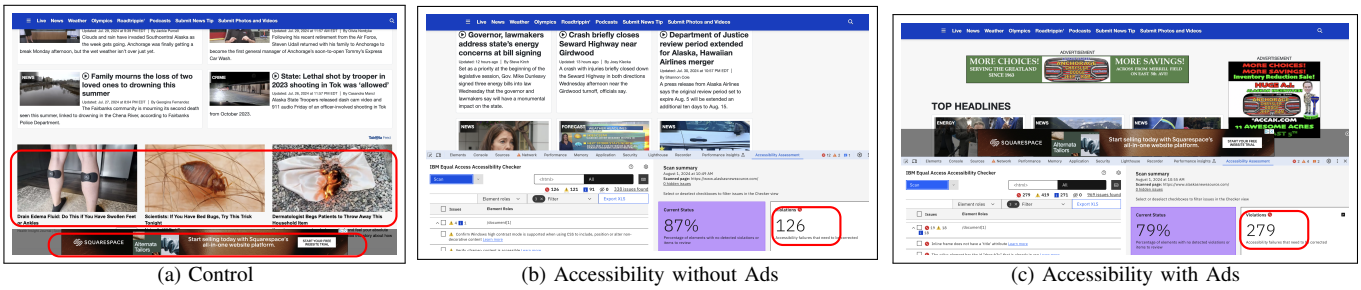
Website ads are the driving force behind free and open Internet [7]. Website developers integrate ads on their websites by leasing space to ad-serving technologies (*e.g.,* Taboola [8]) that serve user-relevant ads when a visitor accesses the website. Consequently, developers give up control over the ad content served on their websites and whether it meets WCAG accessibility guidelines.

Problem. Including ads on a website poses several challenges in evaluating the website's accessibility. First, popular website accessibility tools, like Google Lighthouse [9], do not include ads in their assessments, which overestimates accessibility in evaluation reports. Second, due to the dynamic nature of ad content, website developers cannot predict which ads will be served to users or whether those ads will adhere to WCAG guidelines. Even when a website developer makes the best effort to ensure WCAG compliance, ads can significantly deteriorate the overall accessibility of the website. Third, inaccessible website ads prevent users with disabilities from accessing a significant portion of the website. This is especially alarming when ads pose privacy and security risks [10]–[14], as they are more likely to lead to fraud, phishing, and scams. Fourth, blanket ad blocking harms the free internet and causes fundamental equity issues by making ads completely unreachable. Accessible ads encourage a free and open internet and equip users with disabilities with the necessary context for safe and informed interactions with ads.

Prior work. While website accessibility has been extensively studied [15]–[23], to date, no comprehensive study has examined the accessibility of website ads at a large scale, leaving a significant gap in our understanding of website accessibility as a whole. Only four studies have examined digital ad accessibility, focusing on small datasets of platform-specific ads and solely on blind website users. Yeung et al. [15] conducted an empirical analysis of 90 websites, auditing ads for compliance with WCAG best practices and interviewing 13 blind participants to provide contextual insights. He et al. [24] studied only 500 ad screens within Android apps and introduced a tool for detecting accessibility violations. Similarly, prior work manually examined `alt text` on 67 websites [25] and conducted a survey to understand screen reader users' challenges with website ads [26]. With the prevalence of website ads and their potential safety risks, the impact of website ads on accessibility remains an open yet important question.

Contributions. In this paper, we conduct the first-of-its-kind large-scale, in-depth investigation of website ad accessibility and its impact on overall website accessibility. We perform

1558-1225/25/$31.00 ©2025 IEEE
DOI 10.1109/ICSE55347.2025.00171



(a) Control (b) Accessibility without Ads (c) Accessibility with Ads

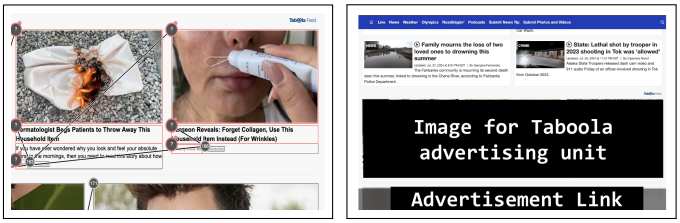
Fig. 1: Accessibility report for `alaskasnewssource.com`, highlighting the count of accessibility violations, which are 126 when the website is visited without ads and 279 when inspected with ads.

an automated analysis of 430K website elements with nearly 100K website ad elements on top 5K news and media websites. We first seek to understand the prevalence of inaccessible ads on the website and how they impact the overall accessibility of websites. We build an automated pipeline to rigorously analyze the accessibility of website elements and ads using IBM Equal Access Accessibility Checker and state-of-the-art ad-blocker uBlockOrigin. Among ads that adversely impact a website's accessibility, we analyze accessibility violations per WCAG guidelines. Identifying such violations provides key insights into commonly overlooked accessibility guidelines in ads and determines whether these violations are prevalent only in ads. To inform developers about accessibility violations across different ad-serving technologies, we further analyze inaccessible ads with respect to the technologies serving them.

Ads are historically more likely to lead to fraud, phishing, and scams [13], [14]. Even when ads are accessible, we investigate whether the provided accessible information truly represents the content and intent of the ad. We utilize open-source CLIP (Contrastive Language–Image Pretraining) model [27] to confirm whether the content of ad images aligns with the provided alternative text for accessible ads. For instance, users with visual impairments solely rely on alternate text to access website content; false and misrepresented alternate text can mislead such users into involuntary interaction with ads. The onus for addressing inaccessible and misleading ads falls on the website developer whose website is hosting such ads.

Findings. Our analysis reveals that 66.7% of websites have ads that frequently fail to comply with WCAG guidelines. On average, we observe 11 more accessibility violations due to ads. We further categorize the most common WCAG guideline violations found in ads to illustrate their potential impact on individuals with disabilities. We find 28 different WCAG guideline violations in ad elements, with 7 of these violations predominantly occurring in only ad elements. These violations are related to missing captions, missing focus order, and missing page language, all of which are critical in ensuring smooth navigation on a website. Our analysis also identifies the ad-serving technologies responsible for the majority of these violations. We find that 52% of websites use one of Google's ad-serving technologies (e.g., DoubleClick), which account for over 14K violations in their ad elements.

We show that the misrepresentation of data in alternate texts compared to the actual image deceives users with visual



(a) Accessibility Violations in Ad Slots of Taboola. (b) Ads are not accessible to users relying on screen readers.

Fig. 2: Violations identified in Taboola ad slots and their impact on how screen reader users experience the website.

impairments and cognitive disabilities, making it difficult for them to interpret the intent of the ads. We find that 27% of ad images misrepresent their content and intent. Furthermore, our manual inspection of ads from top-50 news websites reveals that 94% of the ads collect trackable user identifiers upon `onmouseover`, `onhover`, and `onclick` events. These events help assistive technologies identify website elements that a user with disability desires to access. Thus, navigating ads can compromise the privacy of assistive technology users.

Data Availability: The data artifacts of this work are available at https://github.com/SEED-VT/Web-Ads-Accessibility.

II. Motivating Example

This section presents a case study showing how ads on websites may violate accessibility guidelines and affect the overall website's accessibility. Alaska's News Source, `alaskasnewssource.com` [28], provides local news, weather updates, sports coverage, and community events for the state of Alaska. It is ranked 14th in the top 5K websites in the news and media category. In the stock Chrome browser, this website shows numerous ads in ad slots—three at the top and one at the bottom, as seen in Figure 1(a). The top slots are served by Taboola [8] and the bottom by Google Ads [29].

Website accessibility without ads. In the first step, we generate the accessibility report for `alaskasnewssource.com` by computing the violation count without including any ads. We remove ads from the website to emulate the website development stage, where website developers are unaware of the precise ads that will be served by the ad-serving technologies. This is a common practice employed by many website quality and accessibility checkers like Google Lighthouse [9] that do not include ads in their accessibility evaluations. We use IBM's accessibility checker chrome extension, a tool used

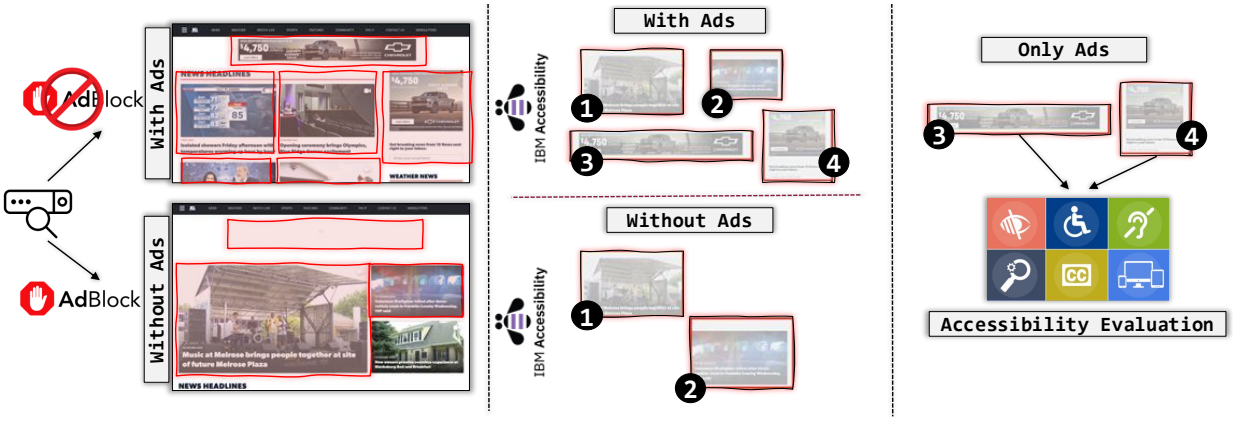


Fig. 3: Overview of our methodology involving generating accessibility reports for each website with ($R_{with-ads}$) and without ads ($R_{without-ads}$), then comparing them to isolate and analyze the accessibility violations caused by only ads ($R_{only-ads}$).

by over 20K developers [30], mainly because it evaluates all website elements, including ad elements. To remove ads from the website, we deploy the uBlock Origin's Chrome extension, which has over 700K users [31], to block all ads on the website. As seen in Figure 1(b), all ads on `alaskasnewssource.com` are removed, and the accessibility checker reports 126 violations. For example, the element in Listing 1 violates the WCAG 2.4.4 (Link Purpose) [32] guideline because it lacks descriptive link text, label, or an image with alternate text. The guideline requires that links must have a clear purpose that users can understand, especially when the link is not accompanied by any other textual context. This can be resolved by adding `aria-label="Download the Alaska's News Source Streaming Apps"` to this tag by developer.

```
<a data-tb-link="" target="\_self" rel="" href="
    https://www.alaskasnewssource.com/page/how-to-
    download-the-alaskas-news-source-streaming-apps/
    " class="link | text-reset d-block h-100 w-100">
```

Listing 1: Anchor tag missing descriptive text

Most accessibility violations can easily be addressed by the developer with simple refactoring of the website's code (*i.e.,* HTML, CSS, and JS).

Website accessibility with ads. Next, we compute accessibility report for `alaskasnewssource.com` by computing the violation count with ads, as displayed to the user. This setting allows us to demonstrate the increase in the violations attributed to ads on the website.

Figure 1(c) shows that ads appear on the website as they did on a normal visit. The accessibility checker reports 279 violations, showing a 121.4% change in the violation count. Given that the only difference between the two websites is the presence of ads, it is reasonable to attribute the increase in accessibility violations to the ads. We further examine specific violations related to ads, particularly concerning the ad slots Taboola serves.

Figure 2(a) shows violations identified by the accessibility checker across three ad slots for Taboola (highlighted with a red circle around their title and short description). This violates the WCAG 2.1.1 (Keyboard) guideline [32] because the element cannot be interacted using a keyboard alone. All ad images are designed with a `link role`, which does not include a tabbable element. Thus, users who rely on keyboard navigation cannot directly access these elements. Additionally, most of these ad images do not include alternate text to make them accessible for visually impaired users. We further test the accessibility of these ads with Google's Chrome Vox screen reader [33]. Vox only announces `image for Taboola advertising unit` on tab focus without reading the title and short description of each ad. Figure 2(b) shows how the website and ads appear to visually impaired users using screen readers, leaving ads' content mostly inaccessible except the two vague alternate texts.

It is difficult for a website developer to anticipate and address ad accessibility violations as these violations are non-deterministic and occur after the deployment of the website. In the case of `alaskasnewssource.com`, ads are negatively impacting accessibility, which raises the concern of whether such an effect is prevalent across all ads: (1) if yes, then what kinds of violations do they cause? and (2) if not, then does the accessible information in these ads adequately represent their content? Similarly, it is useful to know which ad-serving technologies are serving more inaccessible ads.

III. Research Questions

Given the proliferation of ads on popular websites, their impact on vulnerable users, and the gap in developers' understanding of ad accessibility, we aim to answer the following research questions:

1) **RQ1:** How do advertisements impact the accessibility guideline violations of websites?
2) **RQ2:** What are the most common accessibility guideline violations found in advertisements?
3) **RQ3:** Which ad-serving technologies frequently exhibit accessibility violations in their served advertisements?
4) **RQ4:** How common is it for advertisements to have misrepresented alternate text?

IV. Methodology

To address the research questions, we design an automated pipeline to inspect websites, identify accessibility violations

Column	Description	Example
Page Title	Identifies the page or html file that was scanned.	Fox News
Page URL	Identifies the page URL that was scanned.	www.foxnews.com
Element	Type of HTML element where the issue is found.	`<video>`
Element Code	Code of HTML element where the issue is found.	`<video hola-pid="4" loop="" muted="" autoplay="" playsinline="">`
Xpath	Xpath of the HTML element where the issue is found.	`/html[1]/.../.../video[1]`
Check point	WCAG check points this issue falls into.	2.1.1 Keyboard
WCAG level	A, AA or AAA. WCAG level for this issue.	A

TABLE I: Sample accessibility evaluation report detailing elements and associated violation information.

in each element, extract violations specific to ad elements, and categorize them based on WCAG guidelines [2]. Figure 3 illustrates the process used in our empirical investigation, which is completely automated and packaged into a standalone measurement tool.

A. Websites Visited

We visit the top 5K news and media category websites from SimilarWeb [34], using a custom Selenium (version 4.11.2) based automated website navigator running on Google Chrome (version 120.0.6099.129). Prior literature has also focused on this category of websites because they have the most ads [24], [25], [35], offering the most breadth and coverage of ads. For every website, we launch our automated website navigator to first visit the website and then eventually trigger the accessibility evaluation using the IBM accessibility tool, as explained in Section IV-C. On average, it takes a website approximately 30 seconds to fully load (until the onLoad event is fired) and an additional five minutes to ensure all ads are loaded before applying the accessibility checks. We use a stateless crawling process with all cookies and local browser states cleared between consecutive visits. This approach helps ensure that the collected accessibility data accurately reflects the current state of the website without being biased by previous visits. We visit all 5K websites from North America.

We crawl each website twice to obtain two accessibility evaluation reports:

1) $Report(R_{with-ads})$: Accessibility report of the website with ads.
2) $Report(R_{without-ads})$: Accessibility report of the website without ads.

$Report(R_{with-ads})$ contains the accessibility report for all elements on the website, including ad elements. $Report(R_{without-ads})$ contains the accessibility report for all elements on the website, excluding ad elements. Finally, we extract the $Report(R_{only-ads})$ by differencing the $R_{with-ads}$ and $R_{without-ads}$, which represents the accessibility report for the ad elements only:

$$\underbrace{R_{with-ads}}_{\text{Website elements with Ads}} - \underbrace{R_{without-ads}}_{\text{Website elements without Ads}} = \underbrace{R_{only-ads}}_{\text{Ad elements}} \quad (1)$$

Violation	Element	WCAG	Count
Overall	All	-	430,497
	Ad	-	99,422
Levels	Ad	A	80,220
		AA	43,591
Criterion	Ad	2.4.7 Focus Visible	41,110
		2.4.1 Bypass Blocks	37,357
		3.2.2 On Input	26,629
		1.1.1 Non-text Content	16,387
		4.1.2 Name, Role, Value	15,712
		2.4.4 Link Purpose (In Context)	7,714
		1.3.4 Orientation	6,493
		1.3.1 Info and Relationships	6,118
		2.1.1 Keyboard	6,026
		2.4.3 Focus Order	3,575
		1.4.3 Contrast (Minimum)	2,890
		2.5.3 Label in Name	1,190
		1.2.1 Audio-only and Video-only	1,147
		1.2.4 Captions (Live)	1,144
		1.4.1 Use of Color	1,021
		3.3.2 Labels or Instructions	269
		1.3.3 Sensory Characteristics	92
		3.1.1 Language of Page	60
		1.4.12 Text Spacing	26
		1.4.4 Resize text	21
		2.1.2 No Keyboard Trap	17
		3.2.1 On Focus	13
		2.4.2 Page Titled	6
		2.4.6 Headings and Labels	6
		1.3.2 Meaningful Sequence	3
		1.3.5 Identify Input Purpose	2
		2.2.1 Timing Adjustable	2
		1.4.13 Content on Hover	1

TABLE II: Number of website elements with violations during the campaign, detailing ad elements by violation level and criteria.

Why do we need this approach to find the accessibility of ads? To date, there does not exist a tool that can identify and isolate accessibility scores for individual ad elements. Ads can only be evaluated post-deployment. Ad-serving technology requires only reserved ad spaces on the website during the development phase. The ads appear when the website is served to a user. The ads are selected based on user profiles and real-time advertiser bidding. We opt to evaluate these ads post-deployment on the client side to understand the true impact of ads on accessibility.

B. Blocking Website Advertisements

To measure a website's accessibility without ads, we aim to load the website without any ads in our crawler, which requires removing all ads before performing accessibility evaluations. We leverage a widely used, advanced ad-blocking tool called uBlock Origin [31] to remove ads from a website before it completely loads in the browser. uBlock Origin relies on a crowd-sourced filter list [36], [37], which are databases of rules and patterns that identify and block resources from commonly used ad-serving technologies. Almost all ad blocking tools [31], [38]–[42] rely on the same filter lists and, therefore, have more or less similar effectiveness in blocking ads. Internally, when a user visits a website, the ad blocker scans the page and its resources, comparing URLs and scripts against its filter lists. If it detects elements associated with ad-serving technology, it prevents these elements from loading on the page. This can involve blocking requests to known ad servers, hiding ad elements within the DOM, or preventing ad scripts from executing. As a result, the ad content is not

displayed to the user, resulting in an ad-free website, which we use to perform accessibility evaluations, $R_{without-ads}$. Prior work [43] on the effectiveness of ad-blocking tools has shown that uBlock Origin is the most effective ad blocker.

C. Accessibility Evaluation

After the website loads, we automatically perform a rigorous accessibility evaluation using the IBM Equal Access Accessibility Checker [30], a Chrome extension. Initially, we use Google's Lighthouse [9], commonly used for quality checks during development. However, tools like Lighthouse, Axe [44], and Wave [45] exclude ad elements by default, leading to inaccurate measurements for websites with ads. Thus, we use the IBM checker to assess accessibility for all website elements, including ads. Internally, IBM's accessibility checker injects a content script into the website to interact with the website's DOM. The content script scans the website's DOM to identify elements and runs a series of checks based on predefined rules such as WCAG 2.1 [2]. The results are intercepted by our automated website navigator and stored in separate storage for downstream analysis.

Accessibility Report. Table I shows a sample accessibility evaluation report with a detailed analysis of violations, including page title, URL, HTML element type, element code, XPath, WCAG checkpoint, and violation level. Each checkpoint corresponds to a specific WCAG criterion with a description of the violation. The WCAG level indicates the priority and severity of accessibility violations, categorized into three levels:

- **Level A:** These guidelines are fundamental for all websites. Non-compliance makes website content inaccessible to users with disabilities. For example, alternate text for non-text content (WCAG 1.1.1) is crucial for users with visual impairments. Meeting Level A criteria is a top priority.
- **Level AA:** These guidelines should be met by most websites. Conformance to Level AA significantly improves the accessibility of website content. For example, ensuring that text and images of text have a contrast ratio of at least 4.5:1 (WCAG 1.4.3), which is crucial for users with visual impairments, including those with color blindness.
- **Level AAA:** These guidelines represent the highest website accessibility standards for full accessibility. For example, sign language interpretation for pre-recorded audio (WCAG 1.2.6) benefits deaf users. Level AAA conformance is not mandatory for WCAG compliance as it applies to specific content types. *Note: These violations require manual assessment and are excluded from this analysis.*

Extracting Ad-elements. We first generate the two accessibility reports of each website in our dataset, $R_{with-ads}$ and $R_{without-ads}$. $R_{with-ads}$ represents the accessibility report of all website elements, including ads, as illustrated in Figure 3 ❶ - ❹. $R_{without-ads}$ represents the accessibility report of all website elements excluding ads, as illustrated in Figure 3 ❶ - ❷. Finally, we automatically compare the two accessibility reports by first matching website elements based on the HTML element code. As a refinement step, we use XPath location to identify extracted ad elements accurately. This process results in a precise list of website elements that are not ads (❶ and ❷). We exclude these elements from $R_{with-ads}$, which results in an accessibility report of ads only, $R_{only-ads}$ as illustrated in Figure 3 ❸ - ❹. We analyzed 430K unique HTML elements with accessibility violations across 5k websites, of which 23.1% are ad elements. Computing accessibility reports is time-consuming and resource-intensive due to the extensive checks required on the rendered page, styling, and HTML structure. It also cannot be deployed on the cloud, as certain accessibility metrics can only be measured when active display is available for UI rendering. Therefore, analysis on the 5K website consumed more than 484 CPU hours to generate accessibility reports spanning 20.6 GB. Table II lists the number of website elements with violations that we interacted with during this campaign. Specifically, it details the ad elements with violations, categorizing them by the level and criteria of violations.

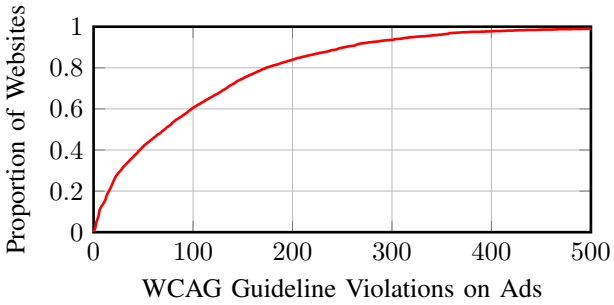


Fig. 4: CDF plot of websites by accessibility guideline violations in ad elements ($R_{only-ads}$).

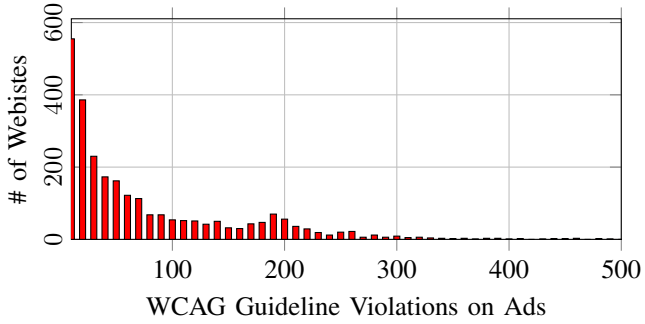


Fig. 5: Histogram of the number of websites by accessibility guideline violations in ad elements ($R_{only-ads}$).

V. Results

A. RQ1: Impact of Advertisements on Accessibility Violations

We first examine the impact of ads on the number of accessibility violations on websites. This analysis demonstrates how third-party ad content can lower a website's accessibility, which has several downstream consequences, such as potentially low search rankings on Google due to accessibility violations [46] in addition to the fundamental inclusivity issues for users with disabilities. Figure 4 presents the cumulative

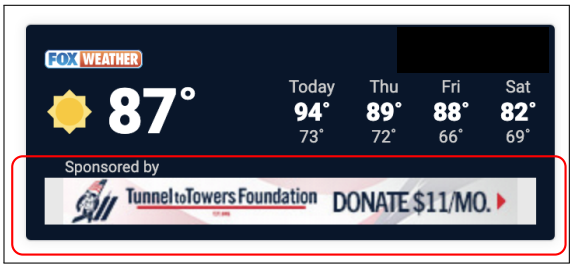


Fig. 6: The Fox News weather widget, featuring a sponsored ad element highlighted in a red box, does not comply with WCAG on input criteria.

Violation in Ad elements ($R_{only-ads}$)	**Websites**
Check the keyboard focus indicator is visible when using CSS declaration for 'border' or 'outline'	2,708
Inform the user when their input action will open a new window or otherwise change their context	2,358
Hyperlink has no link text, label, or image with a text alternative	1,915
Verify `<frame>` content is accessible	1,808
Confirm the element should be tabbable, and if so, it becomes visible when it has keyboard focus	1,632
Content is not within a landmark element	1,605
Verify there is a way to bypass blocks of content that are repeated on multiple Web pages	1,490
Confirm Windows high contrast mode is supported when using CSS to include, position or alter non-decorative content	1,427
Page detected as HTML, but does not have a 'lang' attribute	1,374
Inline frame does not have a 'title' attribute	1,310

TABLE III: Top-10 most prevalent violations in ad elements

distribution of websites and their WCAG guideline violations of ad elements ($R_{only-ads}$). Concretely, for websites without ads ($R_{without-ads}$), the average violation count is 106.2 with a median of 72, For websites with ads ($R_{with-ads}$), the average violation count increases to 118.1, with a median of 81. Overall, the average violation count for ad elements ($R_{only-ads}$) is 11.7. While the magnitude of the increase in violations may not look drastic, it is important to understand that a single violation directly points to an accessibility issue of a website element that makes it inaccessible for at least one type of disability. As mentioned before, WCAG guidelines violations in ads are more challenging to address than the violations in original website content, as a developer can immediately act on violations from original website content, whereas, in the case of ads, it is infeasible for the developer to address the violation due to user-specific dynamic nature of ads.

To further understand the number of violations caused by the ads on websites, we distribute websites with the number of violations caused by ads. Figure 5 displays the distribution of websites for one or more accessibility guideline violations. We find that nearly 66.7% of websites have an increase in violations after including ads ($R_{without-ads} \rightarrow R_{with-ads}$).

(a) Focus Visible (b) Focus Not Visible

Fig. 7: The blue highlighted border shows the focus on the San Diego news block in (a), but it is not visible on the ad element in (b).

> ***Finding.*** Overall, nearly 67% of websites show at least one accessibility violation attributed to ad elements, underscoring the need for further investigation into these violations.

B. RQ2: Common Accessibility Violations in Advertisements

To further understand the reason behind the increase in violations caused by the ad elements on websites, we investigate the different types of violations, when these violations occur, and how common those are in ad elements.

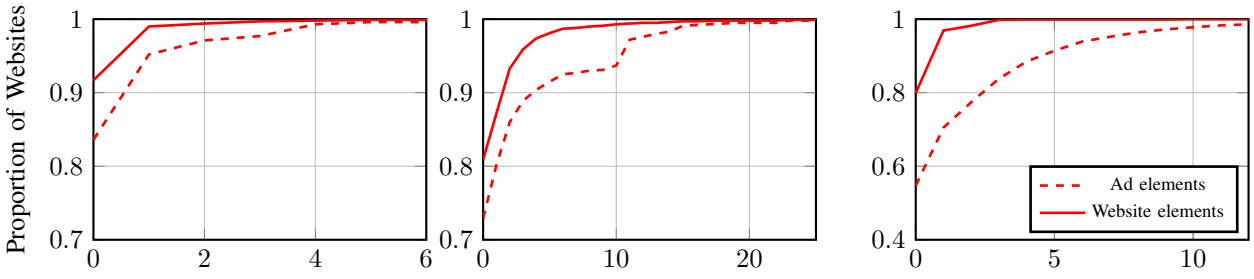
```
<a id="aw0" target="_blank" href="https://googleads.
    g.doubleclick.net/pcs/click?....." onfocus="ss('
    aw0')" onmousedown="st('aw0')" onmouseover="ss('
    aw0')" onclick="ha('aw0')">
        <img src="https://tpc.googlesyndication.com/
    simgad/25226402141320675 27" border="0" width="
    292" height="30" alt="" class="img_ad">
</a>
```

Listing 2: The corresponding `<a>` tag for the aforementioned sponsored ad does not comply with WCAG on input criteria.

1) Prevalent Violations in Ad Elements ($R_{only-ads}$): This analysis provides insights into design-specific violations within ad elements and their potential implications. Table III highlights the most prevalent violations we found in ads across 5K websites. Note that these ad elements with violations are rendered at runtime, preventing developers from addressing them during deployment.

Focus Visible is one of the most common violations. Focus visible enforces that keyboard focus indicators are visible when styling elements like links or ad elements using CSS. In other words, website elements currently focused on should be highlighted. For example, on the website `10news.com`, we find a Google Ads iframe that violates this rule, as depicted in Figure 7(b). In Figure 7(a), the website developer has correctly implemented the focus visible feature by highlighting website cards with a blue border and thus adheres to focus visible. Under WCAG Success Criterion 2.4.7 (Focus Visible) [32], focus indicators must always be visible, which benefits users with motor disabilities, visual impairments (including low vision), cognitive disabilities, and those relying on accessibility tools like screen readers. We observe that 54.1% of the website ad elements have not implemented focus visible.

On Input is the second most common violation in ad elements that fails to inform users when their input action will change the context, i.e., navigating to a different page or opening a new window. This requirement is mandated by WCAG Success Criterion 3.2.2 (On Input) [32]. According to this



(a) Missing Captions (b) Focus-order (c) Language of Page

Fig. 8: Violations are predominantly found in ad elements ($R_{only-ads}$) rather than in website elements ($R_{without-ads}$). The X-axis represents HTML elements associated with specific types of violations, while the Y-axis shows the proportion of websites.

criterion, user actions on an interface component should not automatically change the context without prior notification. Violating "on input" causes confusion or disorientation from unexpected context changes. We observe this violation in ads elements on 47.1% of the websites. For example, on the website `foxnews.com`, we find an ad that violates this rule, as shown in Figure 6. The red-highlighted ad has associated `onclick`, `onmouseover`, `onmousedown`, and `onfocus` events, as shown in Listing 2. These events open a new window (target="_blank"), but there is no indication in the link text or an accessible tag like `aria-label` to inform users that a new window will be opened. It is mandatory to inform users about changes in context to prevent disorientation.

> ***Finding.*** Overall, we observe 28 different types of violations, with the most prevalent being Focus Visible (54%), and On Input (47%) across various websites.

2) Violations in Ad Elements ($R_{only-ads}$) vs. Non-ads Website Elements ($R_{without-ads}$): This analysis highlights the accessibility violations that are specific to ad elements ($R_{only-ads}$) but not quite prevalent in non-ad elements ($R_{without-ads}$). We compare violations from over 99K ad elements and 331K non-ad elements. Figure 8(a-c) shows the distribution of websites with the three most commonly found violations in ad elements ($R_{only-ads}$), which are not common in non-ads website elements ($R_{without-ads}$).

Missing Captions (WCAG 1.2.4 [32]) is one of the most common accessibility violations found more in ad elements ($R_{only-ads}$) than standard website elements ($R_{without-ads}$). This criterion requires that pre-recorded and live audio content in synchronized media (such as videos) includes captions. Captions are essential for users who are deaf, as they provide a text alternative to spoken dialogue and important audio information. On average, 16.4% of ad elements ($R_{only-ads}$) contain at least one instance of this violation, whereas only 8.3% of website elements ($R_{without-ads}$) exhibit the same violation. In other words, ad elements ($R_{only-ads}$) are 2× more likely to be missing captions than non-ads elements ($R_{without-ads}$) on websites.

Focus Order (WCAG 2.4.3 [32]) is found to be more prevalent in ad elements ($R_{only-ads}$) than non-ads website elements ($R_{without-ads}$). This criterion requires that the navigation order of interactive elements on a website is logical and intuitive, ensuring users can navigate through content in a predictable manner. Proper focus order is crucial for users who rely on keyboard navigation, such as individuals with motor impairments or those using assistive technologies like screen readers. Without a logical focus order, users may struggle to understand the structure of the page, miss important information, or find it difficult to interact with the content. On average, 27.1% of ad elements ($R_{only-ads}$) contain at least one instance of this violation, whereas only 19.1% of website elements ($R_{without-ads}$) exhibit the same violation.

Language of Page (WCAG 3.1.1 [32]) is another accessibility violation that is found more in ad elements than non-ad website elements. This criterion requires that the primary language of a website element is clearly defined in the HTML. Declaring the language is essential for users who rely on assistive technologies, such as screen readers, which use this information to provide accurate pronunciation and interpretation of the content. Without the language of the page specified, users with visual impairments or cognitive disabilities may have difficulty understanding the content, leading to a less accessible and non-inclusive website experience. On average, 45.3% of ad elements ($R_{only-ads}$) contain at least one instance of this violation, whereas only 20.1% of website elements ($R_{without-ads}$) exhibit the same violation. Thus, ad elements ($R_{only-ads}$) are 2× more likely to have this violation than non-ad elements ($R_{without-ads}$). This significant difference points to the special ad accessibility case where website developers are not able to enforce language constraints on the ads served on their website due to the user-specific and dynamic nature of websites.

We also find four other violations that are more prevalent in ad elements than non-ad elements. Insufficient text spacing (WCAG 1.4.12) occurs when the minimum spacing requirements in text for readability is not met. This violation is observed in 77 ad elements compared to only 1 non-ad element. Hover content violations (WCAG 1.4.13), where hover-triggered content is not accessible or persistent, are found in 77 ad elements compared to only 8 non-ad elements. Similarly, keyboard accessibility issues (WCAG 2.1.1), such as non-operable elements for users relying on keyboards, affect 9,263 ad elements compared to only 75 non-ad elements are impacted. Lastly, on-focus violations (WCAG 3.2.1), where

Domain	Company	Websites	Violations	WCAG	Description	Count
taboola.com	Taboola	411	17,522	1.1.1	All non-text content have text alternatives to ensure accessibility for users who rely on assistive technologies.	5,699
				3.2.2	User inputs should not trigger unexpected changes or disruptions in content.	5,057
				2.4.7	Any interactive element has a visible indicator of focus to help users navigate and interact with content.	4,658
doubleclick.net	Google	1,577	10,108	3.2.2	User inputs should not trigger unexpected changes or disruptions in content.	3,369
				2.4.4	The purpose of a link is clear from its context, ensuring users understand where the link will take them.	3,344
				2.4.7	Any interactive element has a visible indicator of focus to help users navigate and interact with content.	1,675
revcontent.com	RevContent	325	3,204	3.2.2	User inputs should not trigger unexpected changes or disruptions in content.	1,375
				2.4.1	Users can bypass repetitive content, such as navigation menus, to access the main content more efficiently.	1,113
				1.2.1	Text alternatives/captions for prerecorded audio and video content.	232
googlesyndication.com	Google	1,021	2,339	2.4.7	Any interactive element has a visible indicator of focus to help users navigate and interact with content.	1,844
				1.1.1	All non-text content have text alternatives to ensure accessibility for users who rely on assistive technologies.	410
				4.1.2	User interface elements have proper names, roles, and values to ensure they are accessible and understandable by assistive technologies.	25
google.com	Google	365	2,209	3.2.2	User inputs should not trigger unexpected changes or disruptions in content.	569
				4.1.2	User interface elements have proper names, roles, and values to ensure they are accessible and understandable by assistive technologies.	516
				2.4.7	Any interactive element has a visible indicator of focus to help users navigate and interact with content.	504

TABLE IV: Top five ad-serving technologies with the highest number of violations.

unexpected changes in content or context occur when an element receives focus, are found in 12 ad elements compared to only 1 non-ad element.

Finding. Overall, we observe seven violations more common in ad elements ($R_{only-ads}$) than in website elements ($R_{without-ads}$): missing captions, incorrect focus order, undefined page language, insufficient text spacing, hover content violations, keyboard accessibility issues, and focus-related violations.

C. RQ3: Most Prevalent Ad-Serving Technologies Serving Ads with Accessibility Violations

We investigate the ad-serving technologies that deliver ads on websites with the most accessibility violations. This analysis aims to guide website developers in selecting ad-serving technologies that comply with WCAG guidelines. Additionally, it encourages ad-serving technologies to prioritize accessibility considerations before delivering ads on websites. Table IV lists the top ad-serving technologies with the highest number of violations in ad elements ($R_{only-ads}$).

Taboola is a content discovery platform that promotes personalized content recommendations across websites. In our investigation, it is found on 8.2% of the inspected websites with a total of 18K violations. Common ad violation is related to WCAG guideline 1.1.1 [32], which mandates that all content must have an alternate text for non-text content, found in 5,699 ads. Another common violation is related to guideline 3.2.2, which mandates providing adequate feedback when an action opens a new window or changes the context. This guideline is violated in 5,057 ads. Lastly, guideline 2.4.7, which requires visible focus indicators for interface components to assist users navigating with keyboards or alternative devices, is violated in 4,658 ads served by Taboola.

DoubleClick is Google's ad management platform, optimizing digital ad campaigns. In our investigation, DoubleClick serves one or more ads on 31.5% of the websites. Ads served by Double clicked cause 10K violations. Common ad violations include not informing users when their input action will open a new window or change context (WCAG guideline 3.2.2), which is found in 3,369 ads. Another common violation is related to guideline 2.4.4, which ensures links have clear, context-independent purposes benefiting screen reader users. This guideline is violated in 3,344 ads. Lastly, guideline 2.4.7 mandates visible focus indicators for interface components, which is violated in 1,675 ads.

RevContent is an ad-serving technology that delivers personalized ad recommendations across various websites. In our investigation, it is found on 6.5% of inspected websites with a total of 3K violations. Common ad violations include not informing users when their input action will open a new window or change context (Guideline 3.2.2), which is found in 1,375 ads. Another common violation is related to guideline 2.4.1, which ensures that all functionality is accessible from a keyboard. This guideline is violated in 1,113 ads. Additionally,

guideline 1.2.1, which requires that all multimedia content has a synchronized captioning or a transcript, is violated in 232 ads served by RevContent.

> ***Finding.*** Overall, 52% of websites using Google's ad-serving technologies violate WCAG guidelines across over 14K ad elements. Taboola serves the most ad elements with accessibility violations.

D. Misrepresentation of Alt Text in Advertisements

Among the ads that do not lead to accessibility violation, we investigate the accessible information associated with ads and if they truly represent the intent and content of the ads. Prior work on privacy implications of ads [47]–[51] and malvertising [52]–[54] show that clickbait [55], [56] and false ads [57] can lead to scams, phishing, and fraud. Analyzing alternate texts of otherwise accessible ads is important because it highlights how effective alternate texts are in effectively conveying the content of ads to users with disabilities.

We start by extracting `<img>` tags with alt text from the ads' HTML and collecting their `src` values, which contain the image URLs. Next, we use the CLIP (Contrastive Language–Image Pretraining) model [27] to assess the relevance between images and their alt text. CLIP encodes images and text into a shared embedding space, allowing us to compute the cosine similarity between them. We classify alt text as misrepresented if the cosine similarity is below 0.5, which is the optimal threshold reported in previous research [58]. This approach efficiently evaluates the semantic alignment between visual and textual data.

Overall, across 5K websites, we find 9,827 ad images with alt text. Of these, 27.1% of ads have a cosine similarity score below 0.5, indicating a low level of similarity between the text and the ad image. For example, Figure 9(a) shows an image of shoes on a bench, served by Taboola on `wsfa.com`, but the alt text describes foxes stealing footwear. This alt text mimics a news headline to confuse users, with a cosine similarity score of 0.33. Similarly, Figure 9(b) shows a hotel image on `clickorlando.com`, but the alt text describes hotels turned into apartments for affordable housing. This alt text also resembles a news headline, with a cosine similarity score of 0.28. The alt texts of these ads do not disclose that they are ads and not the content served by the website. Users may only realize they are ads after performing actions like clicking (or pressing enter) to open the link, only to be redirected to a page outside the original domain. Malicious and false ads represented as news in alt text can redirect to scam and phishing websites, compromising the safety of users with disabilities who struggle to assess a website's trustworthiness.

> ***Finding.*** Overall, 27% of the nearly 10K ads have alt text that misrepresents the ad image's content and intent.

(a) Alt-text: Foxes accused of stealing Crocs, sandals from campers after many pairs found at den.

(b) Alt-text: Distressed hotels turned into apartments for affordable housing.

Fig. 9: Misrepresentation of alt-text in image tags of ads.

Privacy implication of misleading advertisements. We manually inspect the consequences of interacting with these misleading ads to determine if any user identifiers are collected for invasive tracking on actions such as `onhover`, `onmouseover`, or when the user presses enter (for tabbable access) on these ads. These three events are key resources for assistive technologies like screen readers to collect accessible information from website elements and deliver it to users.

Prior research has shown that network traffic resulting from interacting with ads may contain user identifiers in headers [59], [60], cookies [41], [61], and parameters [62]. These identifiers can be used to trace a user's browsing activity on the web, violating their privacy. Lists maintained by ad-blockers identify specific headers [63], cookies [64], and parameters [65] as tracking identifiers for users. We compare the network traffic resulting from manually interacting with ads on the top 50 news websites against these lists to identify any tracking of user identifiers. Our results show that on 94% of websites, at least one tracking identifier is sent to an external server by hovering, and on 98% of websites, at least one tracking identifier is sent to an external server by pressing enter on the ads. Both events are essential actions that screen readers perform to read and announce website elements' content. None of the actions are necessary to access ads for regular users.

> ***Finding.*** On 94% of websites, hovering sends tracking data to external servers, forcing users with disabilities to compromise privacy to navigate ad content, unlike regular users.

VI. Discussion and Takeaways

Based on our empirical investigation, we provide insights for website developers and ad-serving technologies to address ad accessibility violations and highlight research avenues for improving the website experience for users with disabilities.

Accessibility-promoting ad integration for website developers. Ad integration is crucial for ad-driven websites. Developers must ensure high accessibility and WCAG compliance, which can be a regulatory requirement [2]. Once published, the developer cannot modify the website to address ad-related accessibility issues since new ads may introduce different violations. Developers can ensure two key aspects to improve ad accessibility during integration. First, they can inject JavaScript to enforce CSS rules (using `!important`)

to implement basic checks on WCAG Type A violations. For instance, a script can be added to ensure all text elements have sufficient color contrast. Such a script can be triggered on `onload` events to detect certain accessibility violations and address them appropriately, if possible. Some Type AA violations can also be addressed by injecting similar checks in the JavaScript, such as ensuring all interactive elements are keyboard accessible. For scenarios involving cross-origin iframes, developers can use transparent overlays in the parent document with ARIA attributes like aria-label to provide alternate text. This method enables developers to communicate the iframe's purpose to users without modifying its content, offering a practical way to improve accessibility in such restricted contexts. Second, developers may also run accessibility checks on their websites post-deployment (similar to our analysis) to analyze ad elements and derive specific self-correcting strategies to address those concerns.

Accessibility strategies for ad-serving technologies. Website developers integrate ads on their websites by leasing ad space to ad-serving technologies. Ad-serving technologies lack incentives to address accessibility issues in ads, as fixing these issues—similar to mitigating malicious ads—can reduce revenue by decreasing clicks and impressions. However, initiatives like the Acceptable Ads program [66] and adherence to guidelines established by the Interactive Advertising Bureau (IAB) [67] represent efforts by some ad-serving technologies to promote more responsible advertising practices. The Acceptable Ads Program promotes non-intrusive ad formats that respect user experience, requiring adherence to specific display guidelines. The IAB sets comprehensive standards for digital advertising, including ad formats, measurement, and data privacy, ensuring compliance with industry best practices. However, none of these current programs and IAB standards address the accessibility of ads, such as readability by screen readers or keyboard navigation. Including accessibility criteria in these guidelines would ensure that ads meet website accessibility standards. Updating IAB standards to cover accessibility requirements would support assistive technologies and enhance compliance with website accessibility guidelines. When leasing an ad slot, the website developer is often provided with the option to enforce restrictions on the type of ads they want to display on their websites. Based on our findings, we advocate that this ad restriction should also include an option to host only WCAG-complaint ads.

User-side accessibility tools. Recent advancements in large language models (LLMs) can significantly benefit accessibility tools by addressing common accessibility violations automatically [68]. For instance, when dealing with missing `alt text` for images, LLMs can analyze the image's content and generate descriptive and contextually relevant `alt text`. These models can provide accurate descriptions by leveraging advanced image recognition and natural language processing capabilities, making the content more accessible to users with visual impairments. However, hosting large-sized models in a browser extension is not feasible. Small, domain-specific language models are an active area of research, which can advance a browser's ability to auto-transform inaccessible ads into WCAG-complaint with minimal runtime cost.

VII. Threat to validity

Internal validity. Our analysis in Section V involves collecting ad elements ($R_{only-ads}$) by differencing website elements without ads ($R_{without-ads}$) from those with ads ($R_{with-ads}$). However, various confounding factors may influence $R_{only-ads}$. For instance, some websites dynamically fetch different elements, including ads, on each visit. Consequently, website elements loaded in $R_{with-ads}$ might not appear in $R_{without-ads}$. Other factors include changes in behavior due to the environment (*e.g.,* browser and host OS), visit time, and location. To minimize this threat, we perform the two visits in quick succession and maintain consistent environments across different states, ensuring the same location, browser, and stateless crawls. The web experience is often personalized; covering all possible personalized browsing instances for a website is challenging to account for all variations in the inspected websites due to different personas. To address this, we analyze nearly 100K ads across diverse types and personas to ensure broad coverage. To avoid biasing our analysis toward a single persona, we employ stateless crawling, clearing session data (cookies and local storage) between visits.

External validity. The IBM Equal Access Accessibility Checker [30] on different browsers might produce different accessibility violations due to the browser's built-in features, such as Safari's default blocking of certain ad-serving technologies [69]. We conduct our experiments using the Chrome browser with a Chrome-based extension, which is the most widely used browser, to minimize this threat.

Construct validity. We collect the website's elements at page load time and do not capture changes in accessibility violations triggered by user interactions such as scrolling and clicking. This is a general limitation of dynamic analysis that is challenging to overcome due to the numerous navigation and interaction scenarios on websites. However, fuzzing techniques [70] can help address this limitation by performing forced execution of various interaction paths and simulating user behaviors such as clicks, scrolls, and form submissions. This approach increases coverage of potential accessibility violations caused by dynamic changes during user interactions.

VIII. Related Work

Accessibility of websites and mobile applications has been extensively studied [16]–[23]. Hackett et al. [17] used the Internet Archive's Wayback Machine to demonstrate that websites have become progressively less accessible as their complexity has increased. Teixeira et al. [20] conducted a systematic literature review focusing on website accessibility within the tourism industry. Similarly, Moreno et al. [71] proposed a method to improve website design for accessibility by addressing identified violations. Fok et al. [21] conducted a large-scale analysis of mobile applications to investigate how inaccessible user interfaces are. Similarly, Milne et al. [22] and

Ross et al. [23] found that missing or vague labels in mobile applications hinder screen readers from conveying information to visually impaired users.

Digital ads on websites and mobile applications are mainly studied for privacy-invasive targeted advertising mechanisms (*e.g.,* [10]–[12]), malicious ads spreading malware or click fraud and phishing attacks (*e.g.,* [13], [14]), and deceptive or manipulative content (*e.g.,* [35]). However, there is very limited research specifically focused on the accessibility of online ads. Yeung et al. [15] conducted a qualitative analysis of 90 websites over a month, auditing ads for compliance with WCAG best practices and interviewing 13 blind participants to provide contextual insights. In contrast, our investigation offers a large-scale quantitative exploration of the ad accessibility problem. He et al. [24] conducted an empirical study on the accessibility of mobile ads for blind users in Android apps, identifying significant challenges such as missing `alt text` for images, complex gestures problematic for screen readers, and ads that disrupt app functionality. However, their analysis was limited to only 500 ad elements on mobile applications, so their insights are not applicable to website ads broadly. Similarly, Thompson et al. [25] manually examined `alt text` for ad elements on 67 news websites, while Chiou et al. [26] automatically detected reflow accessibility violations in website ads for keyboard users. All studies [15], [24]–[26], [72], [73] are small-scale and on limited accessibility violations.

Several tools [3]–[6] have been developed to address accessibility issues. Twitter A11y [4] is a browser extension designed to add alternative text to images on Twitter. Zhang et al. [5] created a machine-learning model that infers accessibility metadata from mobile app screens, improving iOS VoiceOver functionality and enhancing accessibility for existing apps. InstaFetch [6] is a browser extension that consolidates website content into a screen reader-friendly interface, allowing blind users to access information across multiple pages efficiently. While these tools may improve the accessibility of general website content, it is unclear if they can address the accessibility violation caused by the ads.

IX. Conclusion

Website ads are an integral part of a website and are essential for supporting free Internet; however, they often fail to meet basic accessibility standards. This paper presents the first extensive study on the accessibility issues of website ads. Our research reveals that 67% of websites show a noticeable increase in accessibility violations due to ads. We also discover that popular ad-serving technologies do not adhere to the WCAG guidelines when displaying ads, and nearly one-quarter of ads that do comply with WCAG guidelines have misleading alt text. Our findings provide insight into the challenges faced by website developers in integrating ads while maintaining accessibility. This research suggests ways for developers to enforce accessibility standards to lessen the impact of ads on overall website accessibility.

Acknowledgment

We thank anonymous reviewers for providing valuable and constructive feedback to help improve the quality of this work. This work was supported in part by the Commonwealth Cyber Initiative (an investment in the advancement of cyber R&D, innovation, and workforce development), National Science Foundation award 2106420, and the summer research program by the Center for the Enhancement of Engineering Diversity (CEED) at Virginia Tech.

References

[1] C. Brown, "Assistive technology computers and persons with disabilities," *Communications of the ACM*, vol. 35, no. 5, pp. 36–45, 1992.
[2] "Wcag-21," https://www.w3.org/TR/WCAG21/, 2024.
[3] S. Kumar, J. Shree DV, and P. Biswas, "Comparing ten wcag tools for accessibility evaluation of websites," *Technology and Disability*, vol. 33, no. 3, pp. 163–185, 2021.
[4] C. Gleason, A. Pavel, E. McCamey, C. Low, P. Carrington, K. M. Kitani, and J. P. Bigham, "Twitter a11y: A browser extension to make twitter images accessible," in *Proceedings of the 2020 chi conference on human factors in computing systems*, 2020, pp. 1–12.
[5] X. Zhang, L. De Greef, A. Swearngin, S. White, K. Murray, L. Yu, Q. Shan, J. Nichols, J. Wu, C. Fleizach *et al.*, "Screen recognition: Creating accessibility metadata for mobile applications from pixels," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–15.
[6] Y. Prakash, A. K. Nayak, M. Sunkara, S. Jayarathna, H.-N. Lee, and V. Ashok, "All in one place: Ensuring usable access to online shopping items for blind users," *Proceedings of the ACM on Human-Computer Interaction*, vol. 8, no. EICS, pp. 1–25, 2024.
[7] D. S. Evans, "The online advertising industry: Economics, evolution, and privacy," *Journal of economic perspectives*, vol. 23, no. 3, pp. 37–60, 2009.
[8] "Taboola," https://www.taboola.com/, 2024.
[9] "Google-lighthouse," https://developer.chrome.com/docs/lighthouse/overview, 2024.
[10] M. A. Bashir, S. Arshad, W. Robertson, and C. Wilson, "Tracing information flows between ad exchanges using retargeted ads," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 481–496.
[11] S. Englehardt and A. Narayanan, "Online tracking: A 1-million-site measurement and analysis," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 1388–1401.
[12] M. Eslami, S. R. Krishna Kumaran, C. Sandvig, and K. Karahalios, "Communicating algorithmic process in online behavioral advertising," in *Proceedings of the 2018 CHI conference on human factors in computing systems*, 2018, pp. 1–13.
[13] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang, "Knowing your enemy: understanding and detecting malicious web advertising," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 674–686.
[14] V. Rastogi, R. Shao, Y. Chen, X. Pan, S. Zou, and R. D. Riley, "Are these ads safe: Detecting hidden attacks through the mobile app-web interfaces." in *NDSS*, 2016.
[15] C. Yeung, T. Kohno, and F. Roesner, "Analyzing the (in)accessibility of online advertisements," in *Proceedings of the 2024 ACM on Internet Measurement Conference*, ser. IMC '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 92–106. [Online]. Available: https://doi.org/10.1145/3646547.3688427
[16] S. Hackett, B. Parmanto, and X. Zeng, "Accessibility of internet websites through time," in *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility*, 2003, pp. 32–39.
[17] M. Campoverde-Molina, S. Luján-Mora, and L. Valverde, "Accessibility of university websites worldwide: a systematic literature review," *Universal Access in the Information Society*, vol. 22, no. 1, pp. 133–168, 2023.
[18] M. K. Mohammadi, V. Esichaikul, and A. Mohammadi, "Accessibility and usability evaluation of university websites in afghanistan: a comparison between public and private universities," *Universal Access in the Information Society*, vol. 23, no. 2, pp. 955–974, 2024.

[19] G. Agrawal, D. Kumar, M. Singh, and D. Dani, "Evaluating accessibility and usability of airline websites," in *Advances in Computing and Data Sciences: Third International Conference, ICACDS 2019, Ghaziabad, India, April 12–13, 2019, Revised Selected Papers, Part I 3*. Springer, 2019, pp. 392–402.

[20] P. Teixeira, C. Eusébio, and L. Teixeira, "Diversity of web accessibility in tourism: Evidence based on a literature review," *Technology and Disability*, vol. 33, no. 4, pp. 253–272, 2021.

[21] R. Fok, M. Zhong, A. S. Ross, J. Fogarty, and J. O. Wobbrock, "A large-scale longitudinal analysis of missing label accessibility failures in android apps," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–16.

[22] L. R. Milne, C. L. Bennett, and R. E. Ladner, "The accessibility of mobile health sensors for blind users.(dec. 2014)," 2014.

[23] A. S. Ross, X. Zhang, J. Fogarty, and J. O. Wobbrock, "Examining image-based button labeling for accessibility in android apps through large-scale analysis," in *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, 2018, pp. 119–130.

[24] Z. He, S. F. Huq, and S. Malek, "" i tend to view ads almost like a pestilence": On the accessibility implications of mobile ads for blind users," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.

[25] D. Thompson and B. Wassmuth, "Accessibility of online advertising: a content analysis of alternative text for banner ad images in online newspapers," *Disability Studies Quarterly*, vol. 21, no. 2, 2001.

[26] P. T. Chiou, R. Winn, A. S. Alotaibi, and W. G. J. Halfond, "Automatically detecting reflow accessibility issues in responsive web pages," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ser. ICSE '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3597503.3639229

[27] "Openai-clip," https://openai.com/index/clip/, 2024.

[28] "alaskanews," https://www.alaskasnewssource.com/, 2024.

[29] "Google-ads," https://ads.google.com/home/, 2024.

[30] "Ibm-accessibility," https://chromewebstore.google.com/detail/ibm-equal-access-accessib/lkcagbfjnkomcinoddgooolagloogehp?hl=en-US, 2024.

[31] "ublock-origin," https://chromewebstore.google.com/detail/ublock-origin/cjpalhdlnbpafiamejdnhcphjbkeiagm?hl=en, 2024.

[32] "Wcag-guidelines," https://www.w3.org/TR/2023/REC-WCAG21-20230921/, 2024.

[33] "Chrome-vox," https://chromewebstore.google.com/detail/screen-reader/kgejglhpjiefppelpmljglcjbhoiplfn, 2024.

[34] "Similar-web," https://www.similarweb.com/top-websites/news-and-media/, 2024.

[35] E. Zeng, T. Kohno, and F. Roesner, "Bad news: Clickbait and deceptive ads on news and misinformation websites," in *Workshop on Technology and Consumer Protection*, 2020, pp. 1–11.

[36] "easylist," https://github.com/easylist/easylist, 2024.

[37] "easyprivacy," https://github.com/easylist/easylist/tree/master/easyprivacy, 2024.

[38] "adblockplus," https://adblockplus.org/, 2024.

[39] "adguard," https://adguard.com/en/welcome.html, 2024.

[40] A. H. Amjad, Z. Shafiq, and M. A. Gulzar, "Blocking javascript without breaking the web: An empirical investigation," 2023. [Online]. Available: https://arxiv.org/abs/2302.01182

[41] A. H. Amjad, S. Munir, Z. Shafiq, and M. A. Gulzar, "Blocking tracking javascript at the function granularity," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2177–2191. [Online]. Available: https://doi.org/10.1145/3658644.3670329

[42] A. H. Amjad, D. Saleem, M. A. Gulzar, Z. Shafiq, and F. Zaffar, "Trackersift: untangling mixed tracking and functional web resources," in *Proceedings of the 21st ACM Internet Measurement Conference*, ser. IMC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 569–576. [Online]. Available: https://doi.org/10.1145/3487552.3487855

[43] I. Castell-Uroz, R. Sanz-García, J. Solé-Pareta, and P. Barlet-Ros, "Demystifying content-blockers: Measuring their impact on performance and quality of experience," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3562–3573, 2022.

[44] "axe," https://www.deque.com/axe/, 2024.

[45] "wave," https://wave.webaim.org/, 2024.

[46] "Seo-accessibility," https://adasitecompliance.com/website-accessibility-seo-impact/, 2024.

[47] T. Book, A. Pridgen, and D. S. Wallach, "Longitudinal analysis of android ad library permissions," *arXiv preprint arXiv:1303.0857*, 2013.

[48] T. Book and D. S. Wallach, "An empirical study of mobile ad targeting," *arXiv preprint arXiv:1502.06577*, 2015.

[49] M. Alrizah, S. Zhu, X. Xing, and G. Wang, "Errors, Misunderstandings, and Attacks: Analyzing the Crowdsourcing Process of Ad-blocking Systems," in *ACM Internet Measurement Conference (IMC)*, 2019.

[50] H. Le, A. Markopoulou, and Z. Shafiq, "Cv-inspector: Towards automating detection of adblock circumvention," in *Network and Distributed System Security Symposium (NDSS)*, 2021.

[51] M. A. Bashir, S. Arshad, E. Kirda, W. Robertson, and C. Wilson, "How tracking companies circumvented ad blockers using websockets," in *Proceedings of the Internet Measurement Conference (IMC)*, 2018.

[52] A. K. Sood and R. J. Enbody, "Malvertising–exploiting web advertising," *Computer Fraud & Security*, vol. 2011, no. 4, pp. 11–16, 2011.

[53] X. Xing, W. Meng, B. Lee, U. Weinsberg, A. Sheth, R. Perdisci, and W. Lee, "Understanding malvertising through ad-injecting browser extensions," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1286–1295.

[54] A. Arrate, J. González-Cabañas, Á. Cuevas, and R. Cuevas, "Malvertising in facebook: Analysis, quantification and solution," *Electronics*, vol. 9, no. 8, p. 1332, 2020.

[55] X. Li, J. Zhou, H. Xiang, and J. Cao, "Attention grabbing through forward reference: An erp study on clickbait and top news stories," *International Journal of Human–Computer Interaction*, vol. 40, no. 11, pp. 3014–3029, 2024.

[56] A. A. Ahmad Azam, N. F. F. Hasbullah, N. M. Abdul Sookor, W. Mohamad Hanafi, M. S. Hassan, and N. A. N. Ibrahim, "A review of the effects of clickbait on online platforms among society," *e-Journal of Media and Society (e-JOMS)*, vol. 7, no. 1, pp. 97–117, 2024.

[57] M. Handler, "False and misleading advertising," *Yale LJ*, vol. 39, p. 22, 1929.

[58] M. Ahmad and M. Mazzara, "Scs-net: Sharpend cosine similarity based neural network for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, 2024.

[59] W. Rieder, P. Raschke, and T. Cory, "Beyond the request: Harnessing http response headers for cross-browser web tracker classification in an imbalanced setting," *arXiv preprint arXiv:2402.01240*, 2024.

[60] T. Bujlow, V. Carela-Español, J. Solé-Pareta, and P. Barlet-Ros, "Web tracking: Mechanisms, implications, and defenses," *arXiv preprint arXiv:1507.07872*, 2015.

[61] S. Munir, S. Siby, U. Iqbal, S. Englehardt, Z. Shafiq, and C. Troncoso, "Cookiegraph: Understanding and detecting first-party tracking cookies," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 3490–3504.

[62] S. Munir, P. Lee, U. Iqbal, Z. Shafiq, and S. Siby, "PURL: Safe and effective sanitization of link decoration," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 4103–4120. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/munir

[63] "headers," https://github.com/uBlockOrigin/uAssets/blob/master/filters/badlists.txt, 2024.

[64] "Cookielist," https://raw.githubusercontent.com/cookiegraph/CookieGraph/main/Data/ublock_filterlist.txt, 2024.

[65] "Adguardblockparam," https://raw.githubusercontent.com/AdguardTeam/FiltersRegistry/master/filters/filter_17_TrackParam/filter.txt, 2024.

[66] "Acceptable-ads," https://acceptableads.com/, 2024.

[67] "Iab-ads," https://www.iab.com/guidelines/iab-new-ad-portfolio, 2024.

[68] J. Gorniak, Y. Kim, D. Wei, and N. W. Kim, "Vizability: Enhancing chart accessibility with llm-based conversational interaction," in *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3654777.3676414

[69] "Safari-itp," https://clearcode.cc/blog/intelligent-tracking-prevention/, 2024.

[70] K. Kim, I. L. Kim, C. H. Kim, Y. Kwon, Y. Zheng, X. Zhang, and D. Xu, "J-force: Forced execution on javascript," in *Proceedings of the 26th international conference on World Wide Web*, 2017, pp. 897–906.

[71] L. Moreno, P. Martínez, and B. Ruiz, "Guiding accessibility issues in the design of websites," in *Proceedings of the 26th annual ACM international conference on Design of communication*, 2008, pp. 65–72.

[72] A. S. Nengroo and K. S. Kuppusamy, "'advertisements or adverse-tisements?'—an accessibility barrier for persons with visual impairments," *The Computer Journal*, vol. 62, no. 6, pp. 855–868, 10 2018. [Online]. Available: https://doi.org/10.1093/comjnl/bxy104

[73] A. H. Amjad and M. A. Gulzar, "Poster: How do visually impaired users navigate accessibility challenges in an ad-driven web?" in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 5021–5023. [Online]. Available: https://doi.org/10.1145/3658644.3691389