ASE2023 Home

xliu234@buffalo.edu ☰

#230 PTdetector: An Automated JavaScript Front-end Library Detector

Main Edit Your submissions (All) Search

☑ **Email notification**

Select to receive email on updates to reviews and comments.

▼ **PC conflicts**

Hadi Hemmati
Maleknaz Nayebi
Song Wang

Accepted

Submission (1.3MB) 5 May 2023 7:00:53am AoE · 74fa318e

▶ Abstract

Identifying what front-end library runs on a web page is challenging. Although many mature detectors exist on the market, they suffer from

[more]

▶ Authors

X. Liu, L. Ziarek [details]

Supplementary material (1.2MB)

▶ Topics

	OveMer	Rec
Review #230A	4	
Review #230B	4	
Review #230C	3	
Review #230D		A

You are an **author** of this submission.

Edit submission

Reviews in plain text

Review #230A

Overall merit

4. Accept

Paper summary

This study presents PTDETECTOR, a browser extension for detecting JavaScript libraries on web pages. Using a propagation tree (pTree)-based detection approach and similarity comparison, PTDETECTOR is able to identify front-end JavaScript libraries at runtime. The study evaluates PTDETECTOR's detection capability compared to LDC and Wappalyzer on real-world web pages. The dataset used consists of 36 different libraries. The results show that PTDETECTOR outperforms LDC and Wappalyzer in terms of detection accuracy. Additionally, the study investigates the best settings for the tool. It also discusses some limitations of the proposal. Overall, PTDETECTOR proves to be an effective solution for detecting JavaScript libraries in web browser environments. Overall, I enjoyed while reading the paper and acknowledge the effort in providing not only the theoretical approach, but also a fully functional web browser-based implementation.

Strengths

The authors have made a significant contribution to the state-of-the-art by providing an implementation of the approach. This plug-in greatly enhances the practicality and potential adoption of the idea.

Although the topic is very specific (perhaps even too specific), I find the authors' motivation to be well-reasoned and convincing. The contribution appears substantial as it addresses the limitations present in the current state-of-the-art proposals.

The paper is well-written and well-organized. I particularly appreciate the background section, with one exception which I will discuss below.

The idea is centered around the detection of front-end JavaScript libraries. However, I believe the proposal can be generalized to other types of uses and even to technologies associated with JavaScript, such as TypeScript. It would have been beneficial if the authors briefly discussed this potential generalization of their proposal.

The paper is verifiable, as the authors provide both a replication package and the plug-in itself for use in the Chrome web browser.

Weaknesses

I have not identified any weaknesses that would hinder the acceptance of the paper. However, I do believe that the paper could benefit from improvements in certain important areas:

- Threats to validity are not addressed.
- While it is commendable that a plug-in is provided to verify the implementation of the proposal, I feel that the authors place too much emphasis on the browser extension itself. It would be beneficial if they made a clearer distinction between the contribution of the proposal and its implementation.
- The empirical validation could be more extensive, particularly in terms of comparing it with the state-of-the-art, considering the sources of the libraries, and examining the geographical dispersion of the websites, among other factors.

Comments for authors

Section II.D (Background > Our tool) does not appropriately serve as background information. Instead, it appears to be a combination of text that should be placed at the beginning of Section III (even before III.A) and within the related work section. This portion of the text does not describe the necessary fundamentals or state-of-the-art concepts required for understanding the proposal. In fact, to fully comprehend the proposal, readers need explanations of certain concepts and terms that have not been provided, leading to the use of forward references (which is not ideal). I suggest moving the first part to Section III (or even to the introduction) and the remaining part to the related work section.

It would be beneficial if you could further elaborate on the motivation behind the proposal's specificity to front-end JavaScript libraries. While the motivation for detecting this type of libraries is understood, my suggestion is to explore the generalization of the proposal. Are there specific characteristics of front-end libraries that limit the application of this idea in other environments?

In Section IV.A (first paragraph), when you refer to "techniques," do you mean "libraries"?

Furthermore, in Section IV.A, when you mention assigning the threshold score t a value of 0, please provide more details about this score and the criteria for assigning it.

I would have appreciated a more in-depth discussion in the experimentation section regarding the influence of dependencies in the analyzed cases.

The "Limitations" section is interesting, but it falls short of being a mere enumeration. For example, when you address the problem of module detection (ES6), you simply state that ES6 modules are not popular. However, it is important to understand that this statement does not offer a solution to the problem. I suggest including a more comprehensive discussion on how your proposal could be adapted to accommodate the new implementation of JS library loading.

Regarding the separation of the method from the implementation, would implementing the idea on TypeScript libraries pose any significant challenges? I do not perceive many limitations (perhaps apart from potential static analysis complexities), but I might be missing something.

Here are some additional minor comments:

- In Figure 2, the references to "Sec. 3.3.1," ..., and "Sec. 3.3.3" do not correspond to any sections in the manuscript. Please use "III.C.3" instead (as you also use "III.C3" in the paper).
- Change "Object.getOwnPeopertyNames()" to "Object.getOwnPropertyNames(n)".
- In the second paragraph of Section III.A (from "Several processing steps..." to "...matching efficiency"), there are frequent references to unexplained terms. It would be helpful to rephrase this section to enhance the comprehensibility of the text.
- Please use formal English and avoid contractions (e.g., "don't" -> "do not").
- In Table IV, add a separator line between "cnt" and "No."

- In the related work section, change "\paragraph{Library detection}" to "\paragraph{Library detection.}" (and the subsequent references).

Review #230B

Overall merit

4. Accept

Paper summary

This paper presents a technique to automatically identify front-end libraries running on a web page. Different from the existing detectors that suffer from the ineffectiveness due to the library wrappers, the proposed technique utilizes a data structure, namely pTree, to feature the characteristics of libraries. The authors implemented their technique as a web browser extension and test it using 200 top-traffic websites. The evaluation results demonstrate the effectiveness of their technique.

Strengths

-- Publicly available dataset and tool
-- Intuitive idea and practical implementations
-- A comprehensive evaluation
-- Well-written and easy to follow

Weaknesses

-- Quite a few typos and grammar errors
-- Marginal improvement in terms of accuracy and precision

Comments for authors

1. Overall, the paper is well-written and easy to follow. The target problem is a practical need and helpful for both industry and academia.

2. The authors clearly demonstrate the limitations of the existing detection techniques using the real-world cases. The motivation examples are clear and intuitive.

3. The idea of the proposed technique is also intuitive. They leverage the property trees of javascript file as the detection feature, and can better distinguish libraries that have similar property names.

4. The authors identified several practical issues during the detection, such as interference of dependencies, random variables, and so on. The approach can comprehensively handle them.

5. The authors make their tool publicly available, which can benefit the future research.

6. The room for improving the precision and accuracy is quite marginal, while the improvement of recall is much more significant.

Other issues:
(1) In Page 1, “a whopping 76.6% of them” ==> ???
(2) There are many inconsistent usages of “Line **” and “line ***”.
(3) In Page 3, “in Listing.3 line6”==>“at Line 6 in Listing. 3”
(4) “But” is misused as an adverb in many places.
(5) In Page 4, “Step I, create a vertex v, and determine its type” ==>“Step I creates a vertex v and determines its type”

Review #230C

Overall merit

3. Weak accept

Paper summary

This paper presents a technique named PTDETECTOR to detect JavaScript libraries behind the web. When detecting a webpage, PTDETECTOR first generates its pTree, and then calculates the similarity score of the libraries based on the pTree similarity comparison algorithm. The authors compare their approach against LDC and Wappalyzer with the result that PTDETECTOR outperforms them and can detect Webpack-bundled libraries.

Strengths

- Well-structured
- Straightforward but novel method

Weaknesses

- Lack of evaluation to assess the effectiveness of this tool in identifying bundled libraries.
- Lack of threats to validity

Comments for authors

This paper targets the problem of identifying JavaScript libraries running behind the web, which resolves the limitation of existing tools in effectively identifying bundled libraries. The approach employed seems innovative with a clear structure.

- However, my primary concern is the lack of justification regarding the effectiveness of identifying bundled libraries in the experiment. I suggest that the authors can specify which libraries are wrapped in local scope in the experimental dataset with the ideal experimental result being that only PTDETECTOR can correctly recognize this part of the libraries.

- The authors should discuss the threats to the validity of the proposed method.

- Additionally, Fig. 2 could be more concise, particularly in the section representing 3.3.1.

Review #230D

Metareview

All reviewers agree to accept the paper. Congratulations! For the camera ready version, we strongly recommend the authors to consider to resolve the concerns pointed out by the reviewers, including discussion of the threats to the validity, clarification of the distinction between the technical contribution and the implementation, and more extensive empirical validation.

Recommendation

A. Accept

HotCRP