Automatic detection on CMPs and the journey into the patterns of darkness

Marius Alexander Pedersen

Master's Thesis

Supervisor: Marija Slavkovik

Department of Information Science and Media Studies

University of Bergen

June 1, 2022

Acknowledgements

I would like to express my gratitude to Prof. Marija Slavkovik, for excellent help by always being available when I had questions, needed to discuss sections or needed someone to read what I had written. I would also like to thank the people part of my study environment and family, with special regards to SGI and study room 642 for reading through some of the sections I wrote and coming with tips in how to make it sound clearer.

Marius A. Pedersen
Bergen, 26.05.2021

Abstract

This thesis will study how to automatically classify different dark pattern types for CMPs. After completing research on automatic detection of dark patterns, it uses a scraper to extract different features on CMPs that a program will use to classify five types of dark patterns defined from the research. The program is evaluated using four different statistical measures and achieves adequate results. After the evaluation, the factors that caused misclassifications and how to potentially avoid them is brought up.

Contents

List of Figures

Chapter 1

Introduction

This thesis is concerned with the problem of automatically classifying *dark patterns* from *consent management platforms*, that are also called *cookie* banners/notices.

A cookie is a short text file that a Web server stores on a user's hard drive [4]. A cookie is used for websites to remember the user and their preferences [23]. Cookies are considered personal information and as such are under the same regulation that governs the use and processing of other personal data, such as for example identification number or home address.

A cookie can either be a session cookie or a persistent cookie.[1] Session cookies are only available as long as the session is active and are deleted when the session ends. A persistent cookie also expires, but after the session ends, at a set expiration date.

A cookie may not be from the same domain as the website that is being visited. A domain is a collection of websites under the control of the same entity, like for example an organisation or a person. An information cookie that is from a different domain than the one actively being visited is called a Third-Party cookie[2].

Cookies can be used by advertisers for providing personalized advertising. A study by Yan et, al 2009[34] has shown that the number of clicks on personalised advertisements can improve the click-rate of those advertisements by an average of 670%. Click-rate is a measure used by advertisers to estimate how successful an advertisement is. An ad click-rate indicates that a user clicked the ad and engaged with it, rather than just ignoring it. For some companies, selling advertisements is the main source of revenue. 80% of the income of ALPHABET, Googles parent company, in 2020 came from personalized ads[3].

The regulation of personal data depends on where the user is situated (jurisdiction).

[1] https://www.cisco.com/c/en/us/support/docs/security/web-security-appliance/117925-technote-csc-00.html

[2] https://www.cookiepro.com/knowledge/what-is-a-third-party-cookie/

[3] https://www.cnbc.com/2021/05/18/how-does-google-make-money-advertising-business-breakdown.html

Within the EU and the EEA area the regulation from the EU General Data Protection Regulation (GDPR)[2] applies. The GDPR is a collection of guidelines and regulations that describe the different situations under which a company or an organization is allowed to collect or reuse personal information[4]. In other jurisdictions, other regulations may apply. For example, the California Consumer Privacy Act (CCPA) [1] governs the collection and processing of personal data in the State of California, USA.

To acquire the users consent, a website will have to use a *Consent management platform* (CMP) to interact with the user and present them with the information on which data is collected and how it is used. Some aspects of the CMP design are regulated by the GDPR. For example, the CMP must contain the option for the user to decline the websites data usage and still be able to avail themselves of a minimum website service[5].

There are some differences across different regulations governing the processing and use of data. The CCPA does not require the CMP to contain any functionality other than informing the user that their data will be used [1]. The GDPR however requires that the user must consent to their data being used and that it should be as easy to decline as to accept [2]. However as the case between the French institution CNIL, on one side, and the two tech giants Google and Facebook, on the opposing side, have shown, having the option to reject hidden behind a settings button is a prevalent malpractice. The GDPR states that it is prohibited to require more steps to reject consent in the CMP than to accept to it [6]. Google and Facebook did not comply with this prohibition and were consequently fined with 150 million and 60 million Euros for their offence.

This thesis limits its focus on the GDPR regulation. GDPR requires that anyone, such as a company or an organization, that intends to collect and use personal data, must acquire consent to do so[7]. GDPR states certain specifications on how that consent is to be elicited. For example, it is not sufficient to present the the user with an opt-out option. Instead, the company or organization eliciting consent should offer an opt-in alternative [2]. This is because the elicited consent must be informed and that means the user should actively choose if they want their data collected and used.

Having personal data from its users can be directly profitable to website owners [8, 34]. Consequently, there exists an incentive to prefer that users consent to cookie use. Bauer et, al 2021 [5] show that small changes in the design of a CMP significantly

[4] https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_en.html

[5] https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_en.htm

[6] https://www.cnil.fr/en/cookies-cnil-fines-google-total-150-million-euros-and-facebook-60-million-euros-non-compliance

[7] https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_en.htm

increase or decrease the rate of consent. Thus the website owners have an incentive to seek out and use CMP structures that make it more likely that the user consents to its data being processed. Utz et, al 2019 [30] showed that when the CMP's default design is to offer an opt-in consent choice, as required by the GDPR, only 0.1% of users would choose to actively opt-in for allowing third-party cookies.

In an attempt to guide users into giving consent, websites can use various design elements. Design elements and patterns that are created to serve the intent of nudging a user towards a choice that is not in their best interest are called *dark patterns* [12].

A dark pattern exists in a legally grey area. While the use of dark patterns may be regarded unethical, it may not violate existing laws and regulations. An example of a dark pattern that is prohibited by the GDPR regulation is *pre-checked checkboxes*:

> "Consent should be given by a clear affirmative act establishing a freely given, specific, informed and unambiguous indication of the data subjects agreement to the processing of personal data relating to him or her, such as by a written statement, including by electronic means, or an oral statement. This could include ticking a box when visiting an internet website, choosing technical settings for information society services or another statement or conduct which clearly indicates in this context the data subjects acceptance of the proposed processing of his or her personal data. Silence, pre-ticked boxes or inactivity should not therefore constitute consent." (Recital 32 of Regulation 2016/679).

This recital from the GDPR, together with other GDPR articles, was used in a court case against Planet49 by the German Federation of Consumer Organisations[8]. The Planet49 case is the first time a business lost in court for breaking the GDPR regulation on the grounds of CMP design[9].

Today there are millions of websites[10] and only a few organizations helping to uphold the regulation by reporting those that break it. Some of these enforcers are NOYB[11], and CNIL[12]. However, with todays tools available they will not manage to check each website for dark patterns, or submit lawsuits against each offender, when necessary. For a regulation to be effective there must be a possibility of its efficient enforcement.

A possible solution for enforcing consent regulations is being able to check websites for dark patterns automatically. *Automation* would allow for fast, continuous detection

[8] https://curia.europa.eu/jcms/upload/docs/application/pdf/2019-10/cp190125en.pdf
[9] https://www.cookiebot.com/en/planet49/
[10] https://siteefy.com/how-many-websites-are-there/
[11] https://noyb.eu/en
[12] https://www.cnil.fr/

of dark patterns compared to humans. The automated checker would only need URLs of websites to visit them and identify if there are any dark patterns in their CMP. More specifically, as dark patterns differ in type, an automated checker would identify if a dark pattern type is present.

The automation of dark pattern detection has been considered in the literature [13, 18, 19, 21, 27]. This thesis also attempts to classify dark pattern types automatically.

[13, 18, 19, 21, 27] all attempt to detect the dark pattern types defined by [12]. These types were defined to capture not only manipulation in CMP but in other aspects of web services as well. Soe et, al 2020 [26] define dark pattern types that are specific to CMP. In this thesis we are interested in automatically detecting these.

A dark pattern is essentially a manipulation aimed to affect a human user. An automated classifier can only process data, not the emotion or impression a human has when encountering a CMP. Thus to identify a dark pattern type, the automated checker needs to be informed which data of the CMP is related to which dark pattern type.

One way to automatically detect dark pattern types is to describe them manually through a set of human identified relevant features. Soe et, al 2022 [27] used a machine learning based approach on such human labelled features with a moderate success.

An alternative to human labelled feature detection is to use features of the software code of the CMPs. This is the approach we take in the thesis[13].

The work in this thesis had the following approach. First, I analyzed the available definitions of dark pattern types in the literature and selected the ones were most precise and most pertinent to the CMP context. The dark pattern type definitions used in this thesis are those proposed by Soe et, al 2020 [26]. Second, I identified how to extract data from the CMP and website code. Third, I identified which are the relevant code data features that relate to the Soe et, al 2022 [27] dark pattern type definitions. I constructed a prototype automated dark pattern checker and evaluated it on 2000 websites from the Open Page Rank Initiative[14] list of most visited webpages in the world.

To explain the data extraction from the CMP and website code requires some introduction to HyperText Markup Language (HTML) and how a CMP is coded. A CMP is not an own element on a website but rather a generic tag called “div”. Each element of a CMP is often just given an incomprehensible name of numbers and letters consisting of the generic “div” tags. This can make it difficult to detect a CMP and is one of the reasons why earlier research with automatic detection of CMPs has had such a low recall [21]. The features that I decided to use from the code extracted data were screen-

[13]The research for this thesis follows an empirical methodology using a quantitative research method called "Descriptive research". Descriptive research is a method which requires a phenomena that can be assessed with statistical data [33]

[14]https://www.domcop.com/openpagerank/frequently-asked-questions

shots of the CMP, the text available, the button, the potential “settings” page and if it used checkboxes / sliders, if any of the checkboxes or sliders started in a “checked” position. I will refer to these as simply “the features” from now on. I chose to use these features, after some initial preliminary analysis, because they made it possible to detect five out of the eight dark pattern types defined in Soe et, al 2020[26].

To extract the data from the CMP and their webpages, we clearly needed to be able to enter websites automatically. For this job, a web scraper is typically used. After some testing and research I decided to use a scraper from Liljedahl and Nyquist 2021[17], which I only needed to modify slightly. This scraper attempts to extract the features and will always take a screenshot of the website and if it is there, the CMP.

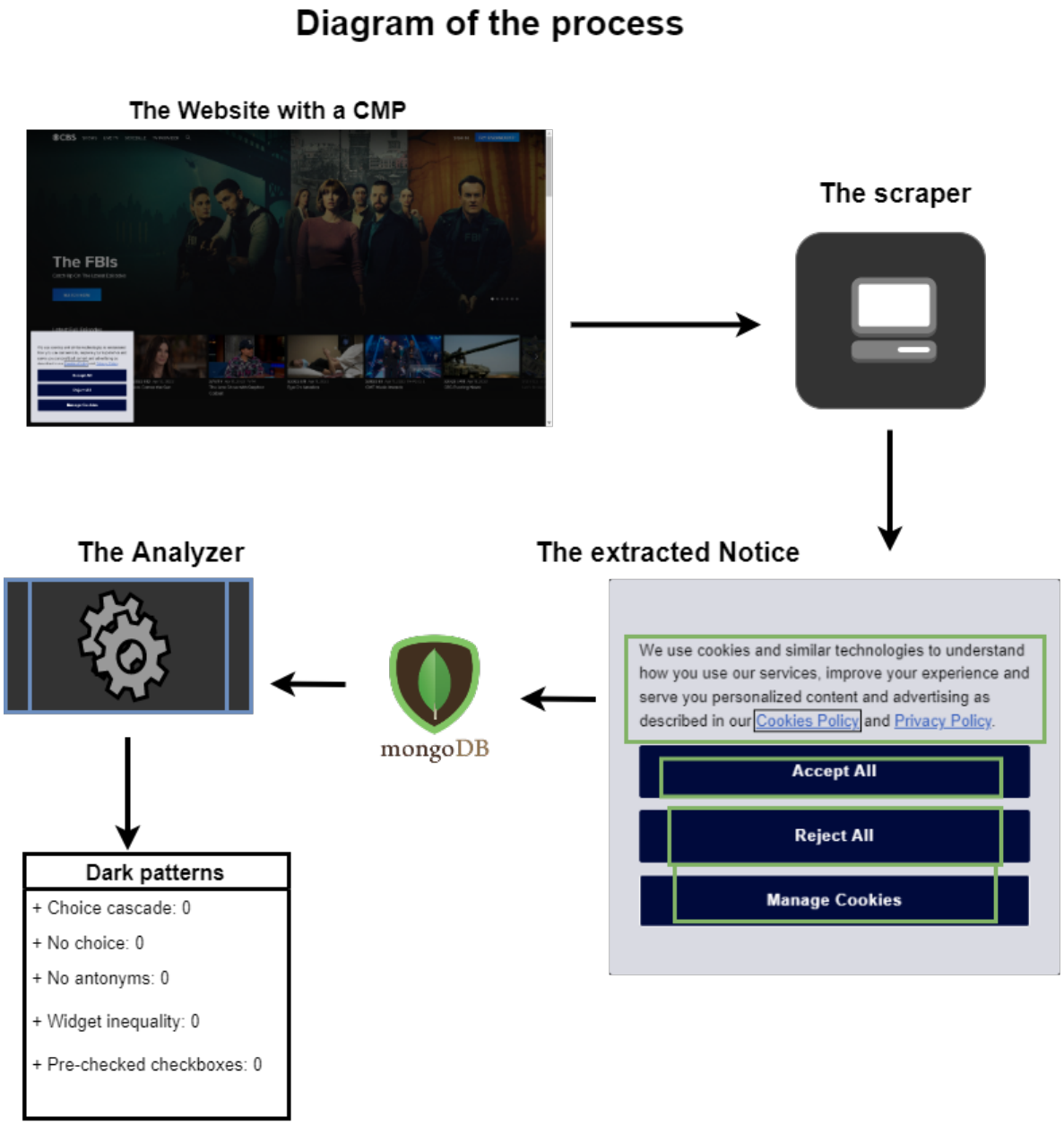

Figure 1.1: Diagram of the automated dark pattern checker architecture.

To evaluate the automated checker prototype, I wanted to use websites that are very likely to be visited. I used a list of such website provided by Open Page Rank Initiative[15]. The list of the most popular websites from Amazon could also have been an option, but that would stop being available on the 1st of May[16], making the reproducibility of this work, a little bit more challenging. I decided to limit my evaluation to the top 2000 websites of the used list, because of the time the verification would take. The number 2000 I considered sufficient to obtain a statistically relevant evaluation.

The diagram in figure 1.1 is the resulting architecture of the automated dark pattern checker. The data from the scraper is stored on a cloud database called MongoDB[17]. I made a python program to extract the information and analyze the data then give a classification of which dark pattern it believes is present on a CMP. The python program referred to as the analyzer attempts to classify five of the eight dark pattern types from Soe et, al 2020[26]. The reason why not all the eight dark patterns were included is discussed in Section 2 The green boxes on the CMP in the diagram highlight which features the scraper extracts.

To summarise, the research question tackled in this thesis is
RQ: Can we automatically detect dark patterns in CMPs from the web code? To answer this research question, we need to solve the following problems:

- find a way to collect data from CMP's of websites,
- find a way to analyze the data to classify the different dark pattern types,
- evaluate how well the dark pattern automatic identification works.

Contribution. This research has found that it is possible to create a program which can automatically extract features from websites and use them to automatically classify different dark pattern types in CMPs. This tool is then evaluated and entries that cause misclassification are presented and discussed.

Thesis outline The thesis is structured as follows.

Chapter2 on dark patterns explains what they are. The chapter goes from when they first were introduced to later definitions of sub types of dark patterns. The chapter explains which different dark pattern types will be focused for this thesis and why some dark pattern types are not being classified in this thesis.

[15] https://www.domcop.com/openpagerank/frequently-asked-questions
[16] https://www.alexa.com/topsites
[17] mongodb.com

Chapter 3 gives an explanation on what Consent Management Platforms is and what they are used for, it also discusses how one can use web scraping to extract the information on seeks from websites, in addition to the related methods the chapters uses.

Chapter 4 presents how the data were extracted and classified. From how to create the necessary programs to extracting the results.

Chapter 5 presents the results with examples of what worked and what didn't including why something did not work.

Chapter 6 presents related work on automatic detection of dark patterns on CMPs.

Chapter 7 presents what we can conclude from the results but also how to improve them and how this could have been done differently. It presents a different approach that uses more machine learning but are less dependent on what can be extracted from a web scraper.

Chapter 2

Dark Patterns

The chapter on dark patterns explains what dark patterns are and several available definitions for the term. The chapter also presents problems regarding automated detection of various dark patterns and how to tackle these issues so that the dark pattern types may be automatically classified.

2.1 History

Dark patterns are a neologism introduced by Harry Brignull[1]. Brignull defined it as “A user interface that has been carefully crafted to trick users into doing things (...) they do not have the user’s interest in mind." Dark patterns are created to trick the user into choosing an option that is not what they would choose on their own. The website from Brignull is still active and includes a “hall of shame with examples of dark pattern design. The hall of shame members includes Amazon, that hides the functionality of unsubscribe to some of their services behind multiple pop-ups with dark patterns. As a consequence Amazon received a legal complaint from the Norwegian Consumer Council 2021[20].

Brignull also classified 12 different types of dark patterns. I give their definitions in the table 2.1.

All of the dark pattern types mentioned in table 2.1 can be read in more depth on the website linked in the footnote.[2] To make these twelve patterns more tractable for use and interrogation by practitioners Gray et, al 2018[12] created five categories which these twelve split into. The five categories have become a staple in dark pattern research and are explained in table 2.2.

[1] https://www.darkpatterns.org/about-us
[2] https://www.darkpatterns.org/types-of-dark-pattern

Table 2.1: The initial twelve different dark pattern types defined

Dark pattern	Explanation
Trick question	Is a question framed with the intention to confuse the reader and nudge them into answer in a particular way. Without giving it much thought, you think to answer your opinion you shall click option 'A', but the question is framed in such a manner that you should answer 'B'. An example of this can be "would you *not not like to consent* to our cookie settings?" In this example there is a double negative which is a tactic used to create a confusing question[12].
Sneak into basket	Upon attempting to purchase something on a website, but in your shopping cart you find the website have added a product you have not clicked on.
Roach model	When a service is designed for it being very easily for you to join/buy their services/product but difficult to get out of/cancel. Amazon mentioned earlier as an example on how it is very difficult to unsubscribe.
Privacy Zuckering	When you share more private information than you would want to share. Inspired by the namesake, Mark Zuckerberg with Facebook.
Price Comparison Prevention	Design that have intentionally made it difficult to compare the price of the product you are looking at with another similar item.
Misdirection	Website design that attracts your attention away from anything the website doesn't want you to notice or spend a lot of time thinking about.
Hidden Costs	When you are at the last step for an online purchase but there have been added costs that until now had been hidden from you. Therefore, making your purchase more expensive than what you intended to, but you are now so far in the process you go along with it.
Bait and Switch	The user set out to do one thing but through the design of the website the user end up doing something else often more undesirable, but possibly more beneficiary for the website.
Confirmshaming	The user is guilted into consenting to what the website asks of them due to the wording of the question and/or the answer options.
Disguised Ads	Advertisement which hides that they are advertisement for a product or service. For example the "article ads" which are ads on newspaper websites that looks like other articles, but is in fact just an article for the product or service that they are selling.
Forced Continuity	When the period from free trial goes to paying member, changes without any notification.
Friend Spam	When a service asks for your email or social media for the website/product/service to be in some way beneficial to you, but what ends up happening is that they use it to send 'spam' mails to all your contacts claiming to be from you.

Table 2.2: The five dark pattern categories from [12]

Category name	Explanation
Nagging	A design element that is present only to be a nuisance until the user interacts with it in a desired way.
Obstruction	Design that makes a process more difficult than it would inherently be due to either blocking design elements or functionality.
Sneaking	Design whose purpose is to hide or delay information that is relevant to the user. A common place this happens is on online stores, where the store may add an item to the users shopping cart without the user knowing about it.
Interface Interference	Design that highlights the parts of the interface the website wants you to use and hides other information that is not equally beneficial to the website. Can be seen on CMPs where it will often be a visible button to accept, and the button to decline will not even be on the first page of the CMP, but on another "settings" page.
Forced Action	Design that forces the user to perform a specific action to use or continuing to use certain functionality of the website. Examples of this can for example be websites which requires the user to make an account to use the website.

There are some issues with the five categories introduced in[12]. One issue is that the dark pattern categories are somewhat ambiguous. which even the authors themselves mentions. This thesis revolves around CMPs, and it can be difficult to differentiate between Sneaking and Interface interference in that context. How does the reject button has to look for it to be considered as “hidden” so the sneaking category applies and, not “less highlighted” than the accept button, for it to be Interface interference instead? Or is it the case that all cases of the sneaking pattern are also a case of interface interference? This poses problems for this thesis that attempt to automatically classify different dark pattern types. When it is unclear how to differentiate the different types, creating code and logic that classifies the pattern types and differentiate between them such that each pattern is unique is difficult. It would not be an accurate depiction of the pattern if the differentiating features between them was for example size difference. Therefore, this thesis needs categories of dark patterns that are more precise in the definition within a CMP context.

2.1.1 CMP feature based definitions

We can find definitions of dark patterns within a CMP setting in the article from Soe et, al 2020 [26]. In the article they manually analyzed 300 CMPs with a focus on GDPR. Defined different types of dark patterns relevant for consent management platforms. The pattern type definitions have a closer connection to how each pattern is represented by features on a CMP than those proposed by [12]. The eight different dark pattern types that are introduced by Soe et, al 2020 [26] are explained in depth in the following paragraphs.

Does not Count is a pattern that emerged after it was shown many websites will still use your data if you deny their consent request. This was shown in [6, 19, 22] and is articles that will be talked more in depth about in related works. To summarize their findings is that consent is taken as given, regardless of if you click to reject to a websites request to use your data. The legalities of this are brought up in Matte et, al 2020[19] with a more legal focused article. Papadogiannakis et al, 2021 [22] found using more advanced techniques of ID and information leaking that more than 75% of all websites have shared the user's information after they have rejected all cookies. It was discovered that for many websites it is better to ignore the CMP than to click deny. Bollinger et, al 2022 [6] took a different approach and made a classifier which can classify what each cookie is, then used this classifier on cookies active after rejecting a CMP. They found that 21% of websites still share with third parties after you click reject all.

No choice is when the user is not given any option to actually deny consent. A common method is that the CMP will only inform the user that the website will use the users personal data. We can see this in figure 2.1 as an example. In this example there is a section that requires the user to write in their date of their birth as it is a site with age restricted content, and then there is text that if you proceed you agree to their privacy and terms & conditions policy. On figure 2.1 there are no option to reject their privacy policy, and therefore this example contains the dark pattern no choice.

Multiple choice panels is, as the name implies, multiple choice panels for the user to consent. From figure 2.2 we see there is a CMP in the middle of the screen giving the user the option to consent or do not consent. At the lower right corner, we see there is another CMP that is also about cookies giving you the option to agree or to read more about them. With these two CMPs it is unclear if the user has to answer both of them or if only one is enough.

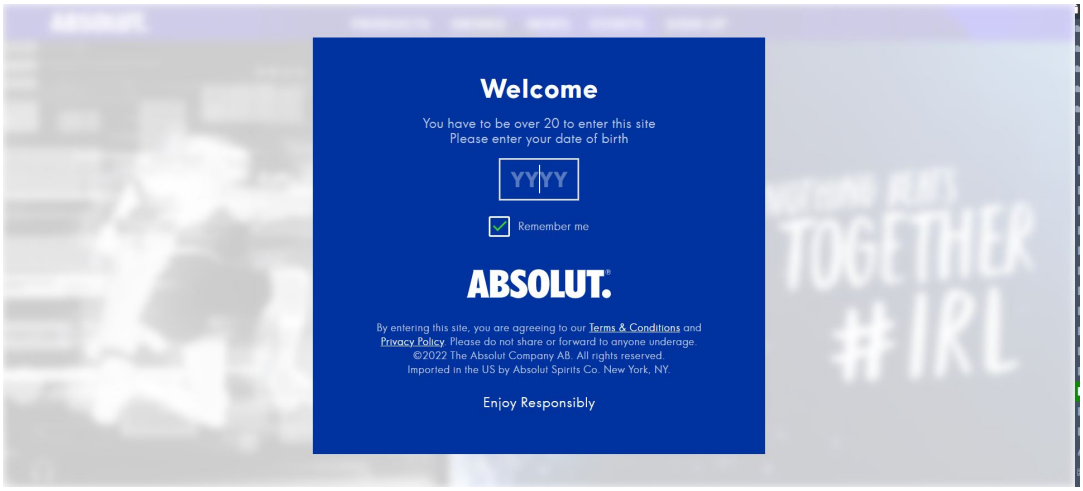

Figure 2.1: An example from https://www.absolut.com/en/ where the user has to accept their cookie policy to enable usage of the website

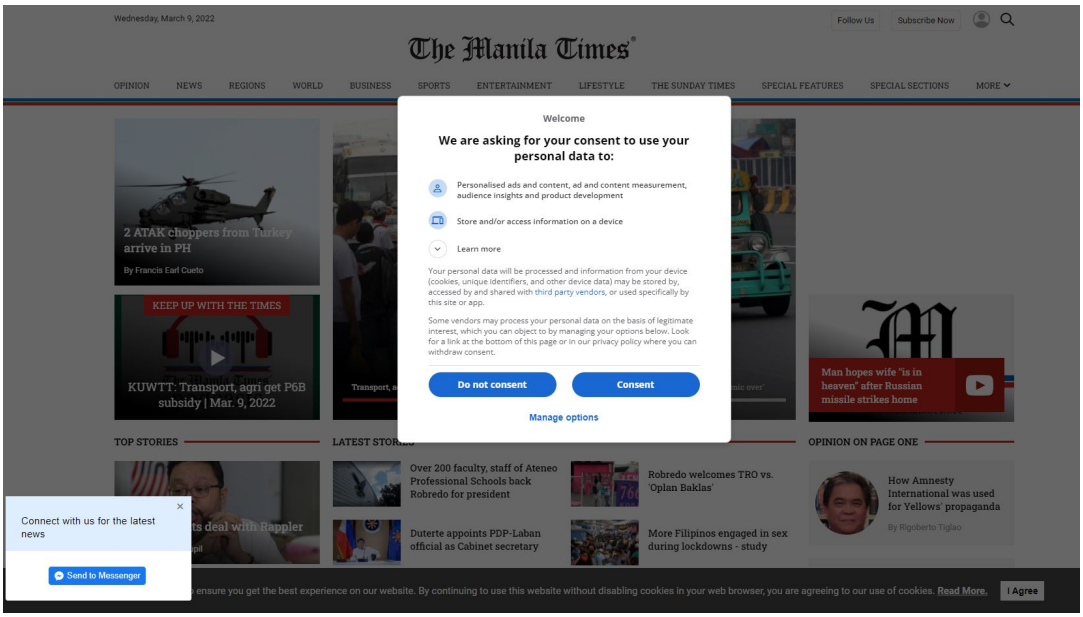

Figure 2.2: Here we see multiple popup CMPs https://www.manilatimes.net/

Choice cascade is when the user has to navigate through multiple buttons or links that offer more information to be able to deny the CMP. This can be seen where the website will have a learn more / settings button instead of a reject on the first page of the CMP. On the learn more page, the user may have to navigate further and then eventually find the deny button on the settings page. From figure 2.3 the user is first presented the options to Agree or go to a more options page. After the user click the more options button there is information on which cookies that is used and reject all button next to a accept all button.

The pattern **widget inequality** refers to a disparity between the consent and refuse options that favors the acceptance choice. This may be accomplished through a variety

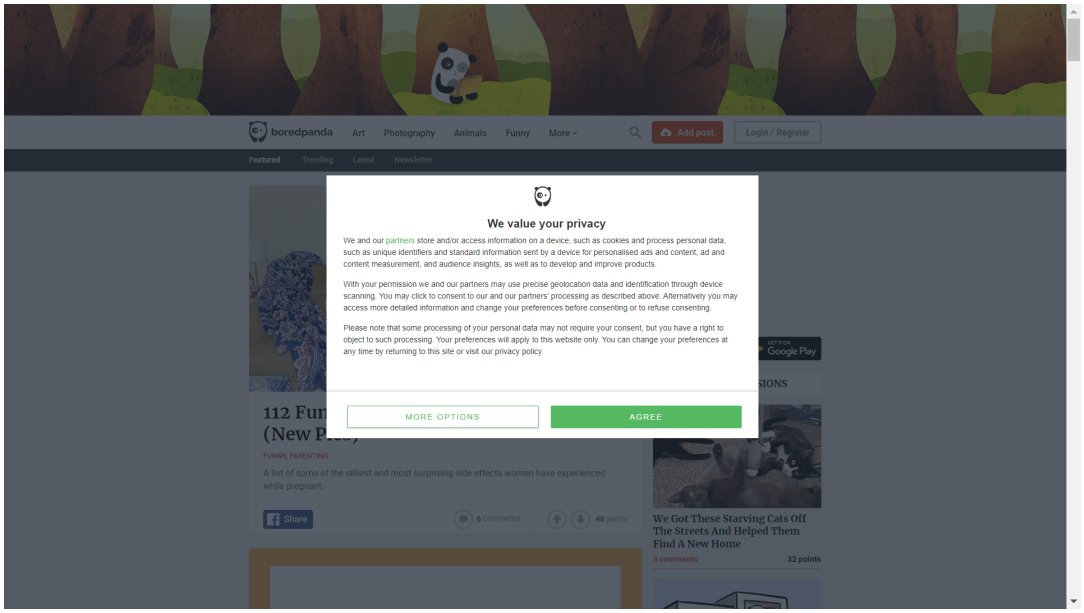

Figure 2.3: An example of choice cascade where the user has to go to 'more options' to reject their cookies *https://boredpanda.com*

of design elements, but it is most evident visually, when the accept button is intended to be prominent with appealing coloring while the reject button, if there, virtually blends in with its surroundings. This do not have to be color differences. However, as long there is a design aspect that favors the accept option over the reject option visually it is a case of Widget inequality. From figure 2.4 we have a case where the accept button is bright blue and larger making it more visible than the deny button. These color variations may seem trivial but has shown to have an effect. For example, when Google tested 41 shades of blue to use on their links. They reported to have earned 200$ million on going with a more purple blue coloring[3].

Unlabeled sliders is a dark pattern type of when the CMP is not labelled clearly what each position a slider can be in, represent. This pattern will also include the checkboxes that often appear cannot be pre-checked when they first appear. As the court case against Planet 49 showed in the Court of Justice of the European Union[4]. On figure 2.5 there is an example of a slider that can be confusing as it is not clear what represents on and what represents off.

Unmarked X is a type of dark pattern that emerges when there is a "X" to close the CMP. However, it is not explained whether closing the panel this way will consent to the CMP or not. Thus, the close function can be misleading especially if there is a slider

[3] https://www.theguardian.com/technology/2014/feb/05/why-google-engineers-designers

[4] https://curia.europa.eu/juris/document/document.jsf;?&docid=218462&doclang=EN&cid=8679428

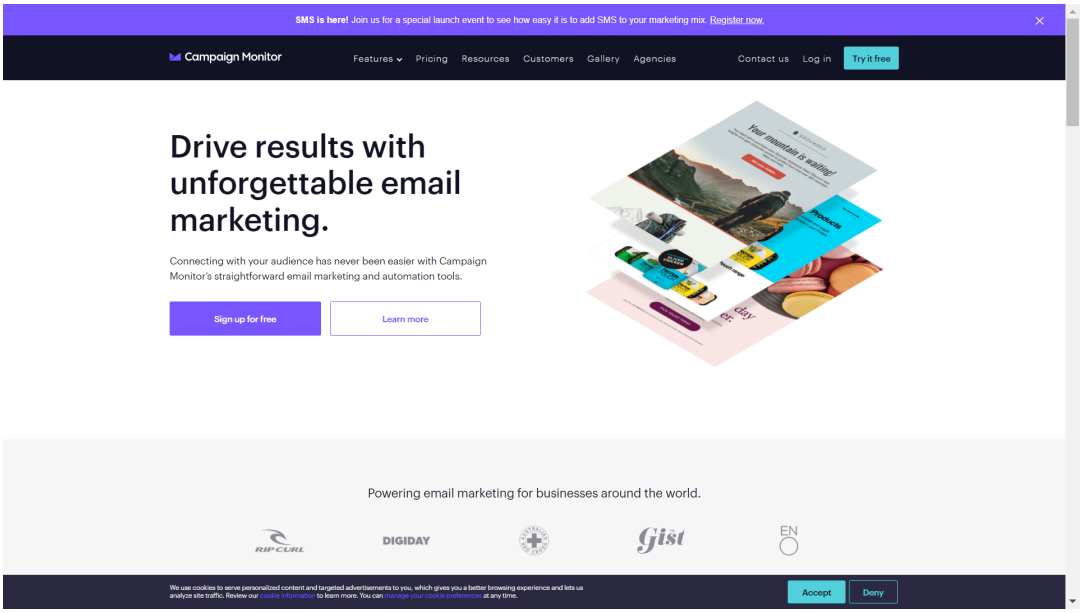

Figure 2.4: An example of widget inequality from https://campaignmonitor.com

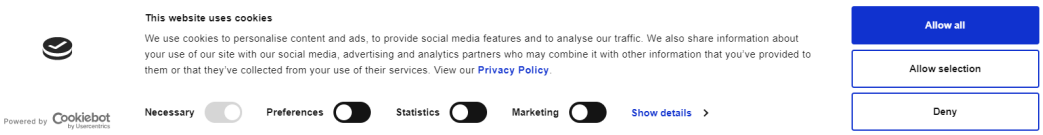

Figure 2.5: An example of a unlabeled slider. From https://www.findhorn.org/

which starts in a 'off' position. What can happen is that the X button is interpreted as another form of accept. In the example given in figure 2.6 we can see the closing X button in the right corner. How this button is interpreted in this example is unknown as they do not write how it will be interpreted either.

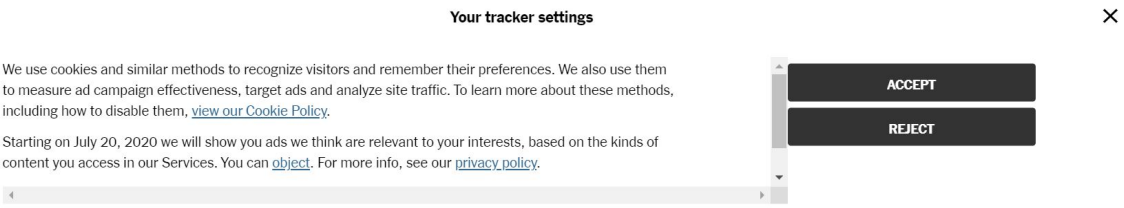

Figure 2.6: An example of an unmarked x from The New York Times. It is unclear what will happen or how it will interpret the 'X'.

The pattern **no antonyms** is used when there are options for both accept and deny but instead of having the two buttons being antonyms of each other, they use different terms. An example can be seen in figure 2.7. Here we can see that the option for declining the CMP is represented in the option "Proceed with required cookies only". It is not clear whether it will decline the cookie policy, without reading the small text above the options. To avoid this dark pattern, it would be preferred they used for

example decline and accept for the two options.

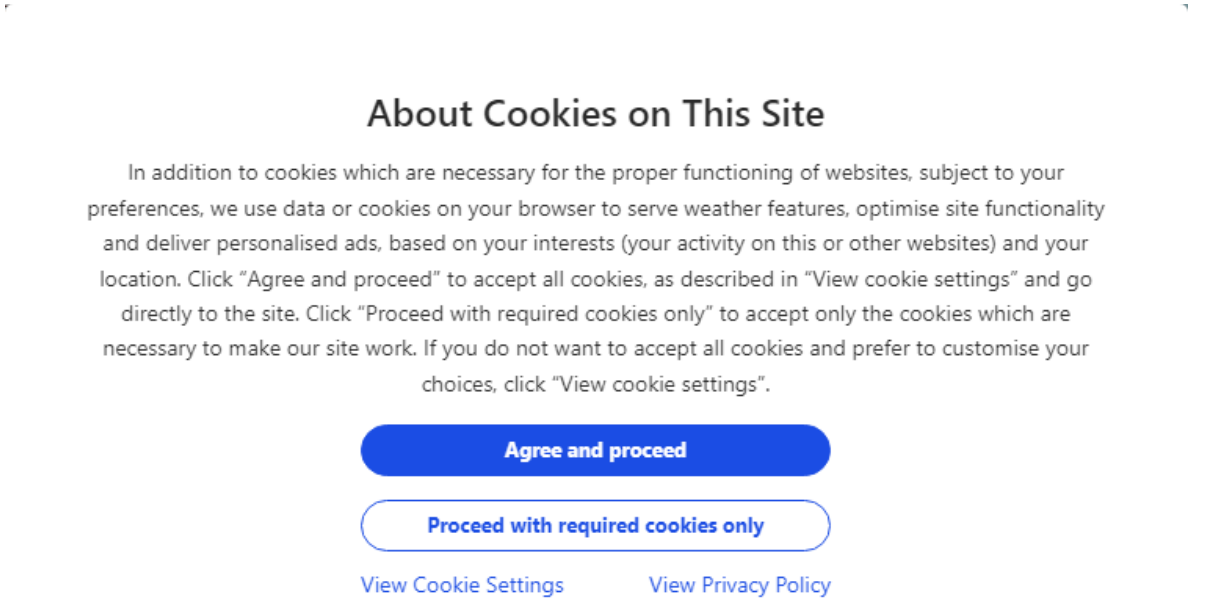

Figure 2.7: An example of no antonyms here from https://www.weather.com

2.1.2 Automatic Detection Considerations

With the definitions presented it is important we balance them in the code between "strict" and "soft". With a "stricter" definition it is easier to lock down what is defined by the term, but you may lose some edge cases that the definition should cover. With a "softer" definition you catch the edge cases, but you may add cases that should not be part of it, and it becomes harder to classify what is and what is not. In this work there will be used stricter definitions on the different dark pattern types to have it clearer defined for the program. For example, is choice cascade interpreted as if the reject button is on another page after the accept, it is a choice cascade. This is regardless of the x closing button being interpreted as a reject option. This is due to know if the "x" is interpreted as a reject button would have to demand an investigation of which cookies the website stores after clicking "x". Or it would have to say in the text, at which we would have to run natural language processing (NLP) and analyze the text to know that it was stated in the text. The program must balance how strict it can interpret the definitions by which features it has available.

An automatic detection program needs a choice of features to use. For example, if one uses only screenshots of the CMP or website one would manage to extract and identify a high percentage of CMPs and would not be reliant on how the website have

coded their CMP. However, one loses direct link to multiple useful features that is important for some of the dark pattern types. For example, the text feature. You can still extract some text from a screenshot, but you must account for some errors and it not being fully accurate for all CMPs. This is due to there is no tool which manages to be 100% accurate in deciphering text from pictures. The text feature is a flexible feature and can be used with natural language processing to attempt to find some interesting patterns, example if there are some terms and expressions that may indicate a dark pattern. The automatic detection and analysis in this thesis attempt to use a balance of using both extracted textual features from the CMP code and features extracted from screenshots taken of the CMPs.

2.2 Types which will be automatic detected

This section presents which dark pattern types will be attempted to be classified and some problems each of the pattern has and how that will be dealt with.

2.2.1 No choice

Number of challenges may impede the automated detection of legitimate “no choice” situations. Some issues arise when websites employ phrases for refusing consent other than “deny” and other one-word refusals, such as “only allow essentials” or “I do not accept”. When attempting to automatically recognize the buttons indicating accept and reject, it will be necessary to search for keywords. If “not allow” is searched for, “allow” will also appear on the same button, which might add complexity. For “only allow”, it may lead to misunderstanding not only because of the same issue as “not allow”, but also because we would need to watch the next word. If the option reads “only allow advertising cookies” or “only allow marketing cookies”, it is no longer a refuse button. Then, a cutoff threshold for the number of words representing an option must be determined. To have an easier time coming up with synonyms for the keyword search, I’ve decided to just use one word in this work; as a result, the findings may have some error margins, as it may have missed certain sites employing multi-word words to describe a refusal or accept option. Thus, to classify the pattern for “no choice”, classification will depend on whether the scraper and the analyzer can discover a decline choice by searching with one-word synonyms for “decline”. If none of them locates a synonym on the initial notification page and the settings page, it is categorized as no choice.

The “no choice” pattern is also heavily reliant on the classifiers assumptions. For when deciding if there is an actual no choice it need to know how the settings page look like. Before the Planet 49 case there was a practice of having the button for accepting various third-party cookies as “on”, then the user had to click reject. However, after this there are not many websites keeping up this practice for users from a GDPR regulated area. Therefore, when the user enters the settings page, and the user click a save button it will often be as if they clicked a reject all button. Since all the checkboxes and cookie settings are set in a “off” position. Therefore, synonyms for decline on the settings page should also include “Confirm my choices” and “save”. See figure 2.8 for an instance where clicking confirm will be interpreted as clicking reject.

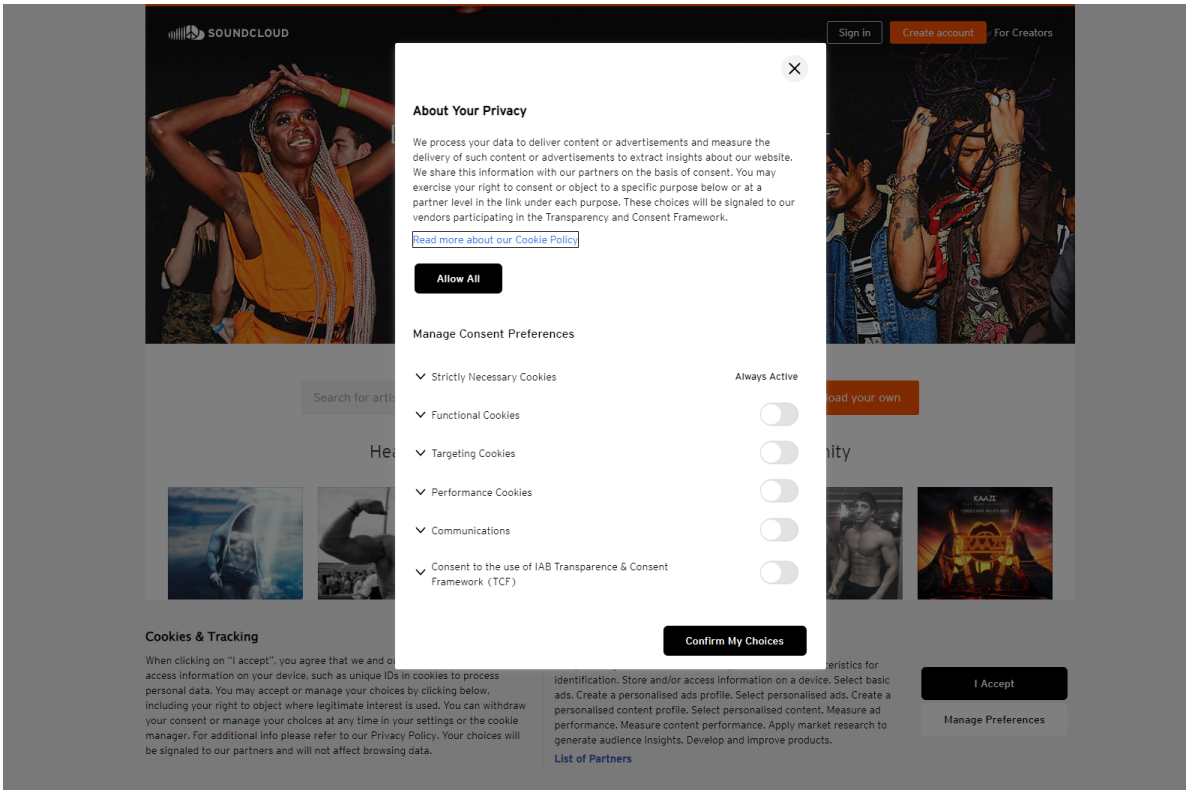

Figure 2.8: An example of where confirm is used for reject, since the options come in an off position https://soundcloud.com

2.2.2 Choice cascade

Some problems for this type of dark pattern are that there might be difficult to properly track how and where the reject button is on the “learn more" or “settings page”. Therefore, it is wise to settle with a definition of “if you must click a “learn more" button or similar, which takes you to another page to be able to give a denial of the websites use of your data, it is a type of choice cascade””. This demands that there is an option of decline as that is the difference between this pattern and no choice Choice cascade and

no choice is less dependent on human interpretation than other dark patterns, making it better suited for machine identification. As is the case with the existing definition, the checker only needs to determine where on the CMP the choice to reject is located. If the option to refuse appears on the initial CMP page, it is not a choice cascade. However, if the scraper/user need to visit a different page to reject, this is categorized as a choice cascade. In the original definition "The denial of consent is only reached by following a number of links or buttons that offer more information"[26] the number of buttons or links the user may click can have an effect on the outcome, but for this thesis we ignore the severity of a offense and classify all offenses equally by using a strict interpretation.

To automatically detect this pattern one need to locate both the decline option and the settings / learn more buttons. If one cant locate the decline button of the first page on the CMP, look on the page that comes after clicking the settings / learn more button. If one finds the decline option here or another settings page / learn more page, and a decline option on the next page, it is classified as a choice cascade. Do note however that after the planet 49 court case, websites are required to have not pre-checked boxes.[5] Therefore when you enter a setting page if you find a "save" option it should save your preferences as if you had clicked deny. See figure 2.8 as an example. This is important to take note of when automatically classifying these types to not misclassify a website with the patterns "choice cascade" and "no choice". This may be avoided by using a different set of search terms on the settings page than on the first. If we believe that every website is following the court case, we may expect confirm, save and store, as well as the other reject synonyms, to indicate the choice to deny cookies. We can however not use "confirm" on the first initial page of the CMP as that can be used as a valid option to accept the CMP on the first page.

2.2.3 Widget inequality

The main problem with automatically classifying the widget inequality pattern, is that in contrast with the two previous discussed dark patterns, is that it is more dependent on human interpretation. An example is when the accept button is green and decline is red. This can be considered a dark pattern because the colors are different, but it can also be seen as intuitive. As red is associated with stop and green go in a traffic light. Therefor the color paring of red and green is an intuitive way of conveying what each button stands for. This also applies for other color pairings as well, with how they

[5] https://curia.europa.eu/juris/document/document.jsf;?&docid=218462&doclang=EN&cid=8679428

fit with the cultural context. If we gather that both buttons are different color than the background, it is hard to classify which is more visible in any reliant automatic way.

One approach to classify on color would either have to map all different color pairings we think are acceptable. Then from those exclude those that do not fit the background and then add some "slack" for the colors. The method would need some "slack" so that color pairing 008000 and ff0000 (green and red) is accepted with similar color variations such that 008001 and ff0001. These colors are very similar for us humans, but for a computer they are different as they have different hex color values.

A design element we can notice making an impact on human impression is size difference. That is why in this thesis it is settled that if there is a significant size difference there is a dark pattern. A size comparison between the decline button and accept button would need to come up with what is a good balance for acceptable differences. I work under the assumption that the buttons don't have to be of equal size, but if the accept button is more than 10% the size of the decline button we have dark pattern. 10% was chosen as a good balance point as it allows sufficient room for design variations, while not allowing accept option be to much larger than the decline button. 10% may not be enough however but finding a correct significance threshold requires further research with human experiments and is out of the scope of this thesis. Changing the size difference parameter will be made easier in the future, but for now it is possible by changing a numeric variable in the code from 1.1 representing 10% to desired size in the analyzer code.

2.2.4 Unlabeled sliders

When the sliders for different options start as "on" it can be confusing if clicking the slider will turn it off for the user as well as demanding more actions from the user for opting out. For automatic detection there are a variety of problems emerging here, such as how the buttons are labelled as "on and "off" as different websites have different setups for this. Some may have "on" and "off" just on each side of the button always visible others may have two pictures placed on top of each other, then when you click the slider, it switches the pictures. It is also the problem with what impression the human would get from the button setup. Some may say it is clearly labeled while others may think its not. An example of this can be seen on figure 2.9. Here it is clearly labeled on and off but there is so many options that it becomes confusing.

When attempting to detect the “unlabeled slider” pattern, one have to detect a slider or the checkboxes available on the CMP. This can be done as a checkbox is its own often used web-element type called “checkbox” with Selenium. That means that Selenium has a built-in method to capture and retrieve checkboxes from websites. However, when you are to detect how they are labelled, problems start emerging. Some websites may have written “on” and “off” outside the checkbox/slider element. To locate the strings then, you must search for a text of “on” and “off”. However, if these are located, there is a cascade of features that now need to be defined. Example how far from the checkbox is acceptable, what size difference between the text and the checkbox is acceptable, Which color is acceptable for the text with regards to the background and the checkbox color, is it acceptable that the text is below or to one of the sides of the checkbox? These concerns are dependent on human interpretation and therefore for this attempt on detecting this pattern we will settle for checking if the slider/checkbox comes pre-checked which as explained earlier is illegal if the user comes from an EU or EEA regulated area.

2.2.5 No antonyms

To automatically detect whether the “accept” and “decline” buttons are antonyms we need to first retrieve the buttons. When buttons are located with their text one need to perform the antonym check. For locating the buttons, I foresee some issues as different pages have different setups. Only searching for a web-element with the button tag as was done with the checkboxes on a website may not be enough and the scraper used for automatic detection should not be reliant on the buttons being tagged properly, so should use multiple different methods for locating any of the buttons. A method is using a screenshot of the website. With the screenshot it should be possible to use an Object character recognizer (OCR), to check whether the OCR can locate decline or accept in case the other methods failed. An OCR is an algorithm that attempts in this context extract what is written from a picture. It will be explained further in the background section of this thesis.

When the buttons are located, the issue of actually finding if they are antonyms starts. How are the algorithm considering what is an antonym to what? With a strict algorithm we lower the number of words we consider to a target word. For example, what is the antonym for “accept”? The first that comes to my mind is “decline” but what about “disallow, reject, refuse, deny, etc...”? To balance the algorithm, I propose to use three antonyms for each word that is a synonym to decline and accept or a term that is frequently used in a CMP setting. So, in total there is nine synonyms used for

"accept" and "decline" each with three antonyms. Another way of detecting if there are antonyms could be to use a dictionary API and when it locates the accept and decline option it will use what the API lists as antonyms. One such API is pydictionary[6]. The pyDictionary and using other dictionaries was considered but were found to be lacking in antonyms as figure 2.10. The figure shows the output from the pydictionarys antonyms for accept, decline, and confirm.

2.3 Excluded types

This section discusses different dark pattern types from Soe et, al 2020[26] that is not classified in this thesis.

2.3.1 Does not count

The dark pattern type "does not count" is not classified in this work as it is the pattern of the eight that have been classified the most before in earlier research. The reader can consult for a legal focused article Matte et, al 2020 [19], if the reader wants to read a more technical focused article on the "does not count" pattern there is Papadogiannakis et, al 2021 for example [22]. They used advance techniques and found the vast number of websites will ignore whether you reject or not to some extent. There is also Bollinger et, al 2022 [6] which took a cookie focused approach where they trained a classifier to learn what cookies was used for what, then used the classifier on which cookies are active after you reject a CMP to register if they ignored the users rejection or not.

2.3.2 Multiple choice panels

The multiple-choice panels pattern provides problems which are harder to solve than the other patterns. The most apparent is that when there are multiple CMPs with buttons for allow or deny it is hard to capture which CMP is the one that provides the valid reject or accept. What happens if you click accept on one and decline on the other? To solve this one would have to monitor which cookies are in use before and after as with the pattern "Does not count". I have only seen it on the https://www.manilatimes.net/ and https://www.bbc.com/. For both websites there seems to be one CMP that is unique, and one created by a CMP provider that is registered at IAB Europe Transparency and Consent Framework[7] which may be a contributing factor in these two websites having two CMPs. To classify this pattern, one will have

[6] https://pypi.org/project/PyDictionary/1.3.4/
[7] https://iabeurope.eu/transparency-consent-framework/

to create a scraper that locates a CMP, then when it has located one it must store the location of that CMP such that it does not locate that CMP again. Thus, after extracting information from the first located CMP, it needs to exclude the elements under the area of the stored positions as for these elements not interfering from the scrapers search for a second CMP. It then would search for a CMP, however now a decision of which CMP should the other dark pattern types be classified from. If for example the first CMP does use antonyms but the second does not, is it a case of "no antonyms"? This pattern is possible to automatically classify but due to time restrictions this will have to be saved for future work for this thesis. I have included an example on how a website looks with multiple choice panels in figure 2.11

2.3.3 Unmarked X

The dark pattern type unmarked is outside of the scope of this thesis as well. The exclusion is due to the rarity that there is an actual x to close the CMP as well as issues extracting the button, on the websites it was present. Some of these issues are due to it being often stored as a "close" button, where the "X" can be an image. See for example figure 2.6. Another issue was to know how closing the CMP with the close button is interpreted it have to say in the CMP which would need natural language processing to extract what the function of the button was if it stood there. I attempted to use OCR to extract the x button. OCR did however struggle and it did not manage to properly extract the button if it was an X. Therefore, due to the issues with the scraper extracting the close button where it was present and the OCR not managing it as well this type was decided to be dropped.

Another way it could be possible to extract the x or close button could be using a trained Convulutional neural network(CNN). The neural net would have to used labelled CMPs however which is the main issue on why it was not attempted in this thesis, as labelling a data set is time demanding. This method on using trained machine learning models to identify different features is elaborated further in the future work section in chapter 7.1.

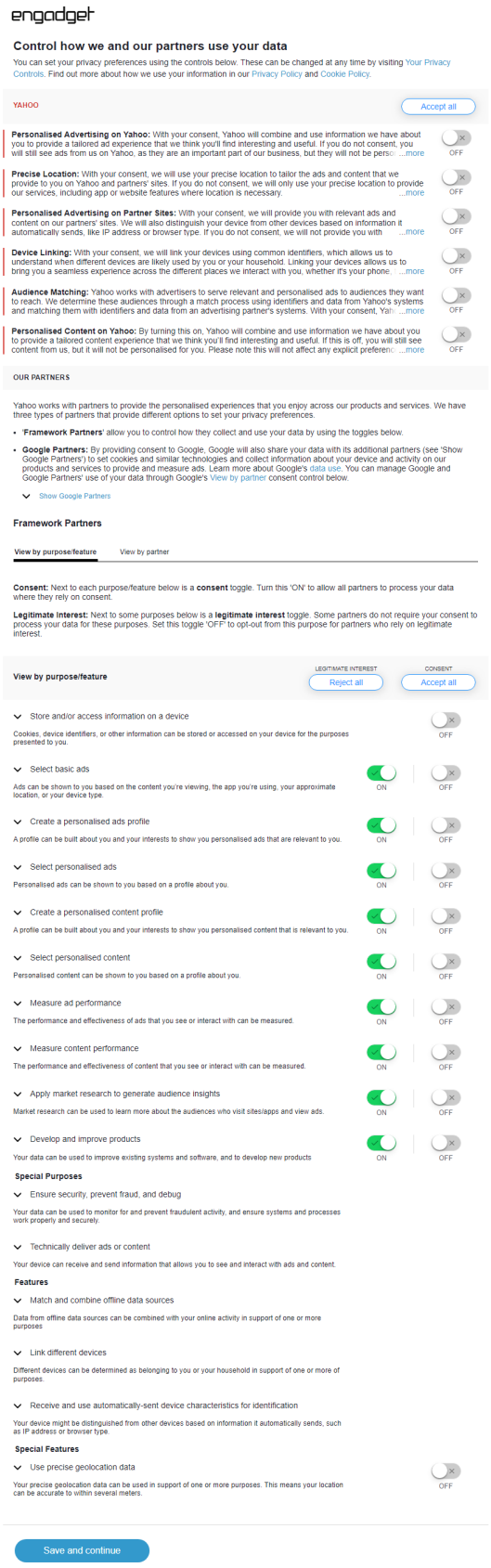

Figure 2.9: Example from engadget.com

```
from PyDictionary import PyDictionary
dictionary=PyDictionary()
```

```
print(dictionary.antonym("Accept"))
print(dictionary.antonym("Decline"))
print(dictionary.antonym("Confirm"))
```

```
Accept has no Antonyms in the API
None
Decline has no Antonyms in the API
None
Confirm has no Antonyms in the API
None
```

Figure 2.10: Running pydictionary to locate antonyms

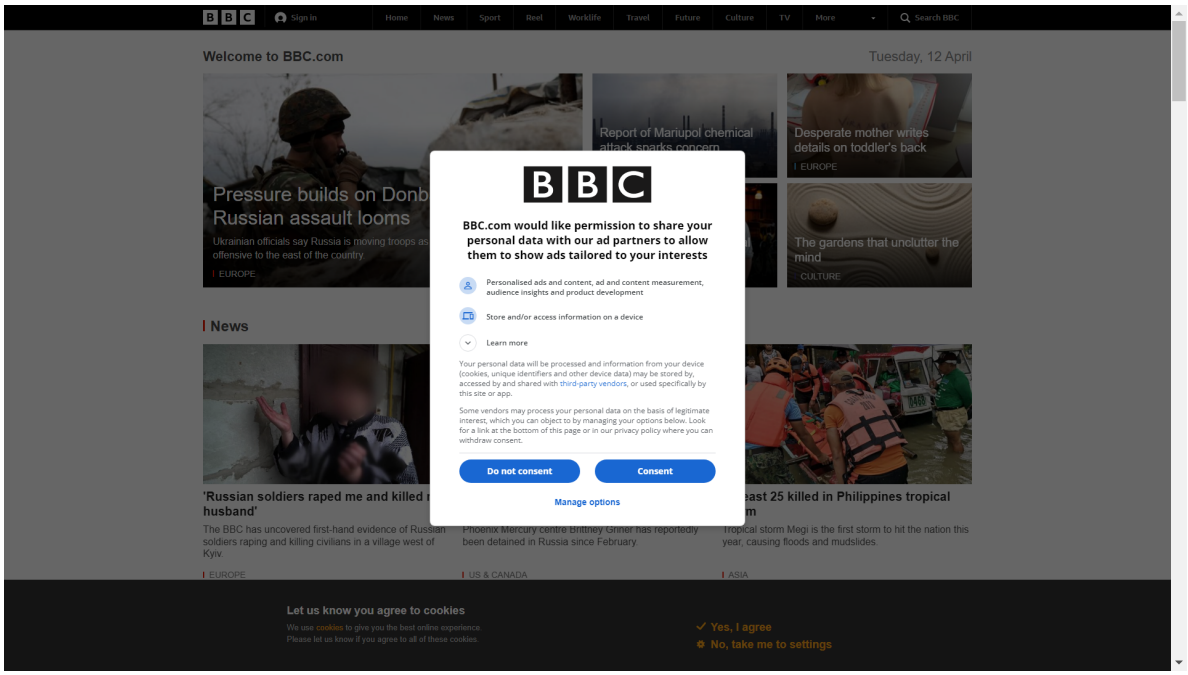

Figure 2.11: BBC have two CMPs, you can spot the second one at the bottom darker than the one upfront https://www.bbc.com/

Chapter 3

Background

3.1 Web Scraping

In this section it will explain what web scraping is and how it is used to collect the data used in this research.

This thesis will use webscraping to extract CMP features from 2000 websites and will have to use a web scraper to complete this task automatically. In the article from Glez-Peña et, al 2013[11], the authors define web scraping as “the process of extracting and combining contents of interest from the Web in a systematic way". This definition captures well what a scraper is. When one does web scraping the user creates/uses a program to extract what information they want and how to get it from websites. The program then follows an algorithm to extract the information that user has defined in an automatic fashion. After the information is extracted, it is stored for further use.

There are many tools available for web scraping that are free and open source. Some of the most popular are Beatuiful Soup[1], Selenium[2] and Scrapy[3]. Beautiful Soup assists in searching and navigating parse trees from HTML and XML files. It has well organized comprehensive documentation which is great for learning how the library works. Selenium has a well-known web driver element which is an object like a browser which you can write methods and functions with on how it should interact with elements on a website. This web driver element can also interact with Javascript features on a website. Scrapy is the fastest of the three in terms of processing and extracting information. Scrapy is the fastest as it can schedule, and process request asynchronously. Thus,

[1] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[2] https://www.selenium.dev/about/
[3] https://scrapy.org/

Scrapy will process other things after it has sent a request and are waiting must wait[4].

Websites are written in HyperText Markup Language (HTML). HTML is a language with a layered structure, where elements can have sub elements which can have sub elements[5]. A sub elements parent is the element it is under in the hierarchy. Each element can have a tag which can indicate what kind of element they are. There are not any CMP-tags therefore CMPs are contained with a generic div tag. See figure 3.1 for an example of how a CMP look in HTML code on a website. This is one of the clearest CMP code I have discovered during the work of this thesis, as they call the CMP a class name related to the GDPR.

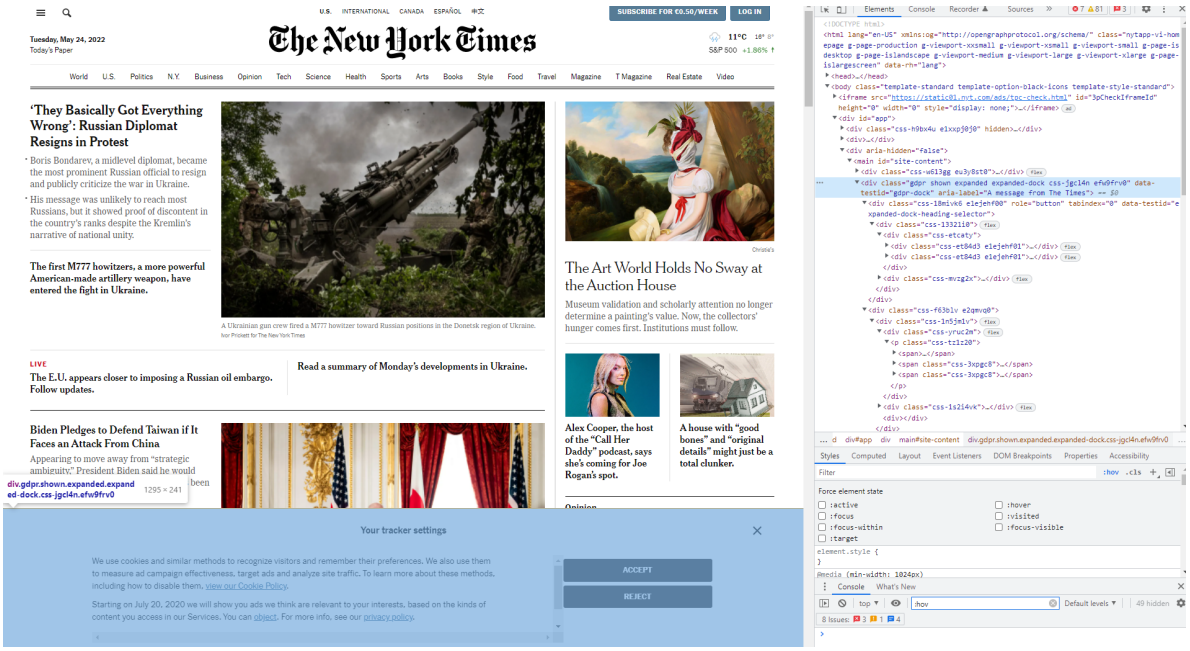

Figure 3.1: An example of HTML CMP code, from nytimes.com

Websites can use CSS to style HTML elements. From the figure 3.1 we can see there are elements called css followed by numbers and letters. For example, css-et84d3 e1ejehf01. This element is then defined and styled in a specific way in the CSS file the site uses. As these elements are not named it convolutes finding the element we are looking for. For example, the text you can see on the figure 3.1 in the CMP, is in a div element called css-yruc2m. With such a naming scheme the scraper cannot use a method that relies on class names but rather what each element has as its value.

The scraper used in this thesis is the scraper created by Liljedahl and Nyquist, 2021 [17] in their thesis. They used the scraper to extract the parameters: the size of the

[4] https://docs.scrapy.org/en/latest/intro/overview.html
[5] https://www.w3schools.com/html/html_intro.asp

CMP, size of the buttons, the color of the buttons, amount of pre-checked checkboxes, the readability level of the CMP, how long the website saves cookies and if the website redirects the user when the user rejects the cookies. The scraper takes a screenshot of each website it visits and a screenshot for both the initial page of the CMP and the settings page for the CMP. This feature along with it using four different methods for locating a CMP made it an ideal scraper to use for the data collection needed on this research. I go more in depth in the implementation chapter on how the scraper is implemented with some modifications and which features it extracts that are of use for the work in this research.

3.2 Machine Learning

> Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience
>
> *Tom Mitchell*
> *Machine Learning 1997*

Here I intend to explain what machine learning and aspects of it that is used in this thesis means. The thesis solution was for a long time of the development period, using a machine learning classification algorithm. However, it was eventually mostly dropped and only one section still uses machine learning.

From the definition in the epigraph, we can gather that machine learning is algorithms. When these algorithms improve their results from an experience or from data it is called learning or optimization. If we then run these algorithm multiple times so that they improve their previous answers it is called training. The algorithms may be quite complex and can contain multiple algorithms for different purposes in the learning phase. The algorithm will also often need multiple variables or units that can represent values and help adjust the function, these units are called nodes or sometimes artificial neurons.

When we have multiple connected nodes we have what is called an neural network also called Artificial neural network (ANN). Each node in a network has a weight assigned to them. The weight adjusts the output one node has to one of the other nodes it is connected to. In a neural net the nodes are arranged into layers. A layer is defined by all the neurons in a section have all their output connections to the following section, and all their input connections are from the preceding section. The only exceptions

being the input layer and the output layer. The input layer is what comes first and therefor gets its signal from external connections. The output layer is the final layer so have no output connections to other nodes. The layer(s) in between are whats called hidden layers. If the nodes in a neural network can create a cycle can be connected to themselves the network is called a recurrent neural network. If the nodes in the neural net cannot create a cycle, the network are called feed forward. The function that connects all these nodes together is called the activation function which will decide what signal the node outputs[6]. After the final output layer and the network have a result, there is a back-propagation function that will adjust the weights for nodes to achieve a better result. An example of how a artifical neural network can look can be seen in figure 3.2

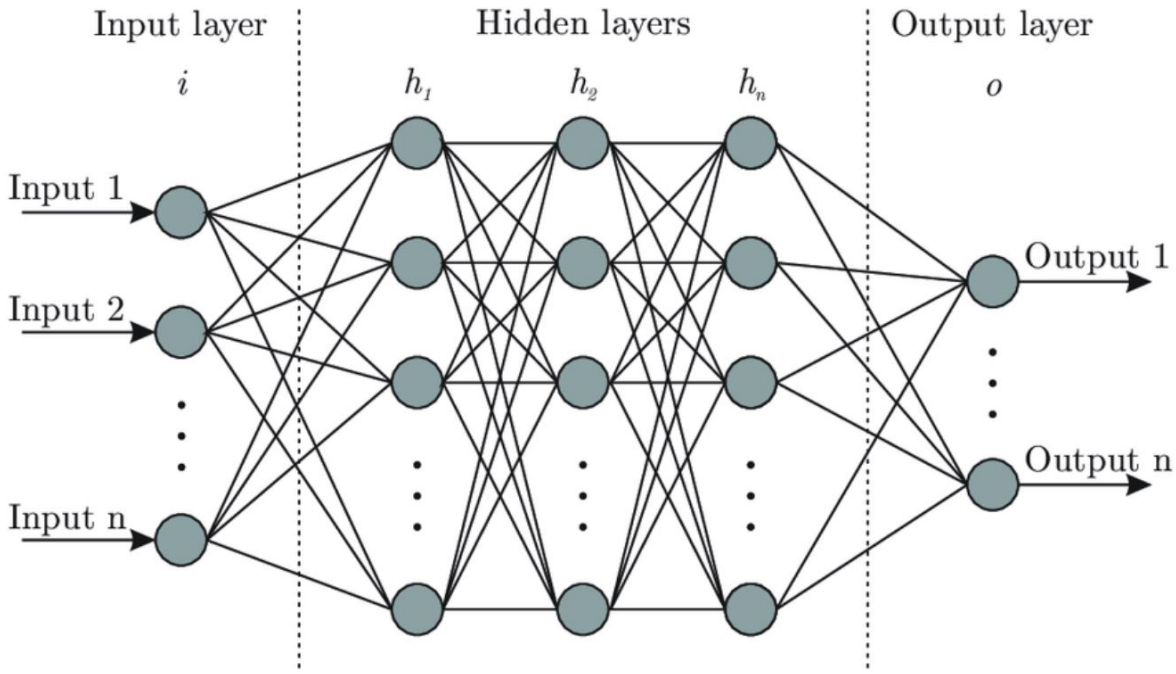

Figure 3.2: An example of a ANN, from [7]

This thesis will utilize 'tesseract' a object character recognizer(OCR) which after version 4 utilizes a RNN with LSTM architecture[7]. An object character recognizer also known as an optical character recognizer is "the process of programmatically identifying characters visually and converting that to the best-guess equivalent computer code."Liedle 2018[16] It was HP which started developing it in the late 1980s, but for a long while it stood mostly untouched until it was released under open source where Google started developing the OCR algorithm further[8].The model scores among the

[6] https://www.v7labs.com/blog/neural-networks-activation-functions
[7] https://github.com/tesseract-ocr/tesseract/blob/main/ChangeLog
[8] https://web.archive.org/web/20061026075310/http://google-code-updates.blogspot.com/2006/08/announcing-tesseract-ocr.html

best OCR libraries struggling a little bit more than the very best on handwritten images, but as we will not be using any handwritten images it will be adequate for this thesis[9].

Recurrent neural network (RNN) is the type of machine learning algorithm this thesis will utilize by using the OCR library tesseract. The RNN feature of having nodes being able to be connected in cycles makes a RNN better at remembering than other neural nets that are feed forward. However, a plain RNN will still struggle with whats called the vanishing gradient problem Hochreiter 1998[14]. The vanishing gradient problem is that older data will get less important due to the values used to update weights, in each layer gets smaller and smaller the bigger or longer the incoming data is. Resulting in the earliest layers achieving very little “learning/change" and remembering of what is important. example of this can be if there is say a sentence “Alice likes to jump on a trampoline. (...) After work Alice likes to jump on ..." the algorithm will struggle on suggesting that Alice want to jump on a trampoline after work since more data have gone through, after the initial sentence and it struggles inferring what is important and not important. However, with a "Long short-term memory"(LSTM) which was introduced in the article called “Long Short-term Memory" Hochreiter 1997[15] it solves this problem. Figure 3.3 is an overview of how this is implemented[10].

3.2.1 Statistics

After an automated tool is built it is evaluated for precision, recall and accuracy of the classification. Calculating, precision, recall and F1 Score have become standard for classification related tasks and originates from Sørensen 1948[28] and Dice 1945[10] independently. Precision is the number it correctly classifies as one type of dark pattern (TP) divided on the summation of TP and the number of incidents the tool misclassifies as an incident of a dark pattern when there is not a dark pattern (FP). Precision has a maximum score of 1. The score of 1 is only true when the classifier classifies nothing as one of the types of dark pattern when they are really not (FP). It can be written in a mathematical fashion as $Precision = \frac{TP}{TP+FP}$Precision is important as it handles the

[9] https://research.aimultiple.com/ocr-accuracy/ between OCR libraries

[10] The solution is to introduces gated logic gates to nodes which goal is to help monitoring what is important to remember and what can be forgotten. The gated logic gates functions in a simple way by normalizing the values it receives as input and with the use of multiplication with weights it checks whether the new value crosses a threshold. Normalizing the values means that all the incoming data in numbers are between 0 and 1 in this case. First if the incoming input is something the node at the gate is connected to needs. The next gate will then calculate the new value’s importance level in using the input from the first gate. With the importance value and the value from gate one it calculates a cell state. Before moving forward it passes a output gate which will calculate the node’s hidden state value based on the cell state and what it previously have stored.Sak et, al 2014[25]

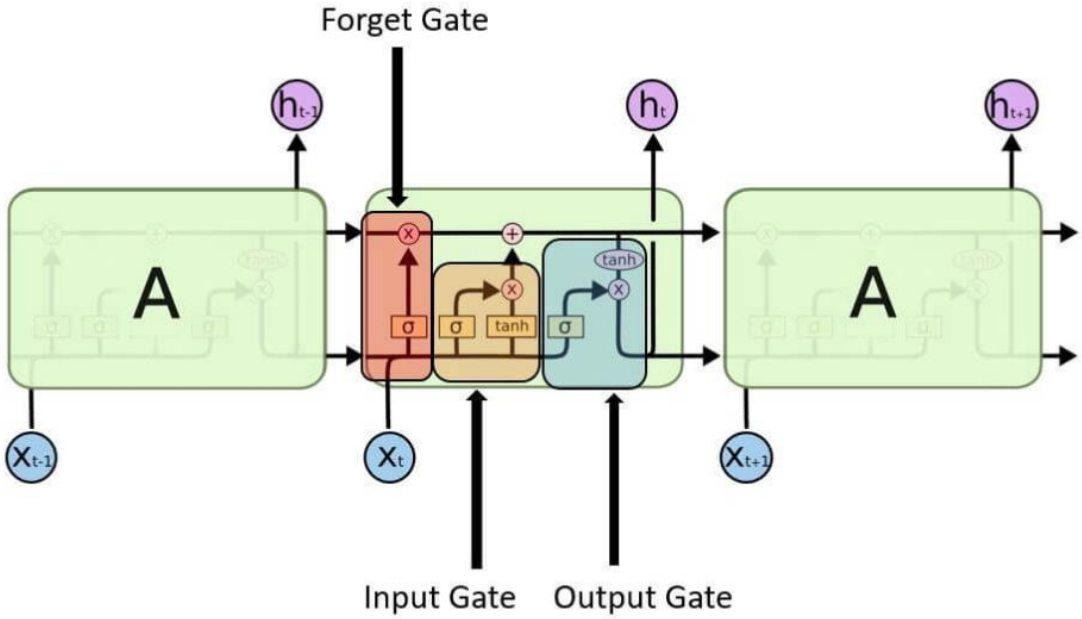

Figure 3.3: An LSTM example, from https://www.javatpoint.com/long-short-term-memory-rnn-in-tensorflow

balance between true positives and false positives. Recall is the number it correctly classifies as dark pattern types (TP) divided on the summation of TP and the number of instances of dark patterns it classifies as not there when it actually was there (False negative). Recall can be written mathematically as $Recall = \frac{TP}{TP+FN}$ Recall is important and informs on which ratio of true positives is classified. However, it is not enough as if one where to classify every object in the data one would get 100% recall and is therefore not useful alone.

With Precision and recall we can calculate the F1-score. The F1-score gives a number between precision and recall and is regarded as a better indicator on how well the checker classifies. The F1-score is calculated by dividing two with the summation of the ratios of both precision and recall. It can be written mathematically as: $F1 = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$ F1-score is however not a perfect measure and have been proved to have a bias as it considers false positives and false negatives as equal and do not use or take in to consideration the True negative cases Powers 2008[24]. It was decided however that the measure could still be useful but other than F1-score, informedness or "Youden's statistic" Youden 1950[35] was considered for the evaluation process.

To have some data that uses the true negative cases it will be calculated accuracy for the different dark pattern types. Accuracy is a measure which is calculated by sum-

marizing what is classified correctly (TP + TN), divided on the total of classification options. (TP + TN + FP + FN).

3.2.2 Expectations

My expectations to the classification of the different dark pattern types is that there will be a high false positive and false negative on the CMPs recall, as they come in a lot of different shapes and sizes with different content. This will contribute to more false positives on the different pattern type classification. For example, I believe no choice will have the lowest accuracy as it is more reliant on detecting the reject option than the others.

Chapter 4

Implementation

This section will go through how the program is set up from collecting the data to analyzing it. A diagram of this process can be seen on figure 4.1.

The process the diagram on figure 4.1 depicts is of the process from website to dark pattern classification. The scraper enters a website in an incognito mode to avoid having any previously stored cookies. When on a website it waits 5 seconds to let the website load, then takes a screenshot of the website before it attempts to locate a CMP. From the figure we can see the CMP in the left-hand corner. In this example, the scraper detects the accept, reject, manage cookies button and the text with the hyperlinks. It then takes a screenshot of the CMP before it clicks the manage Cookies button. When the button is clicked, the settings for changing the cookie policy appears and now the scraper extracts the new buttons, the new text and takes a screenshot. All the detected data is stored in a connected MongoDB cloud database. From the database the analyzer utilizes the features with the screenshot of the CMP to determine which dark pattern is present. In the case for this CMP there were none. This is an overall description of the process and I go more in depth in this chapter on the different parts of it.

4.1 The scraper

A web scraper is software that is made to gather information from websites. When the scraper enters multiple websites in a row automatically, it is called a “crawl". A web scraper is often used by researchers to gather data for their research, but they can also be used by companies. Some may only extract small amount of information, while others like for example Clearview AI scraped more than 3 billion personal photos without asking for consent[1]. There is no limit in how much or what kind of information one

[1] https://www.emarketer.com/content/facebook-promises-delete-over-1-billion-face-scans-law-enforcement-still-has-data

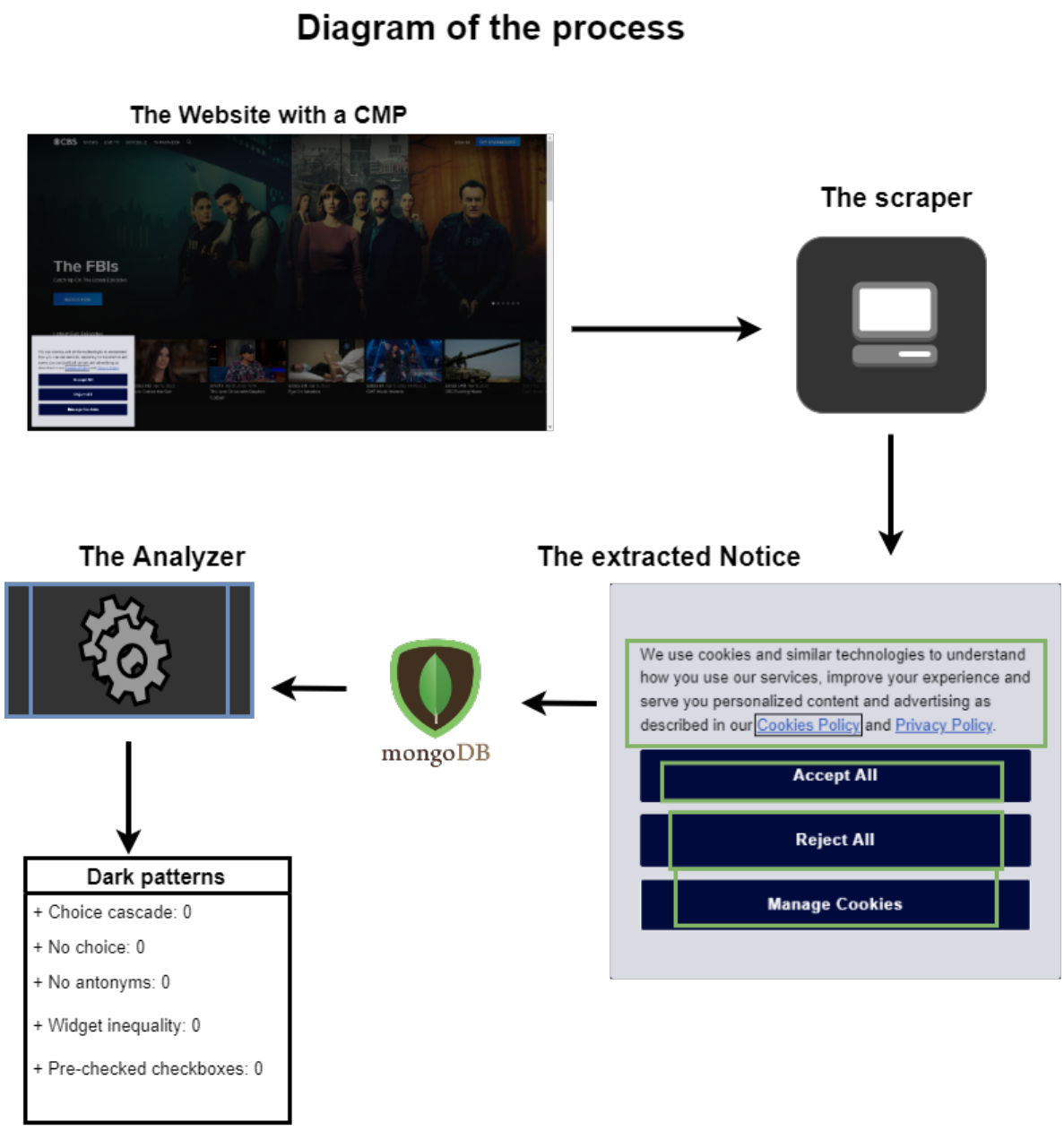

Figure 4.1: Diagram of how the program extracts the data till it is classified

can scrape if it is publicly available, and you have the resources.

The scraper used in this thesis can be accessed with the code for the analyzer with the GitHub link given in the appendix. The scraper is a slightly modified version of a scraper provided in the thesis [17].How the scraper finds the location of the buttons has been altered. The modifications have increased the number of searchable terms for each type of button; in addition to searching for refuse and accept on both the initial and settings pages, additional searchable links have been added when searching for the cookie policy link.

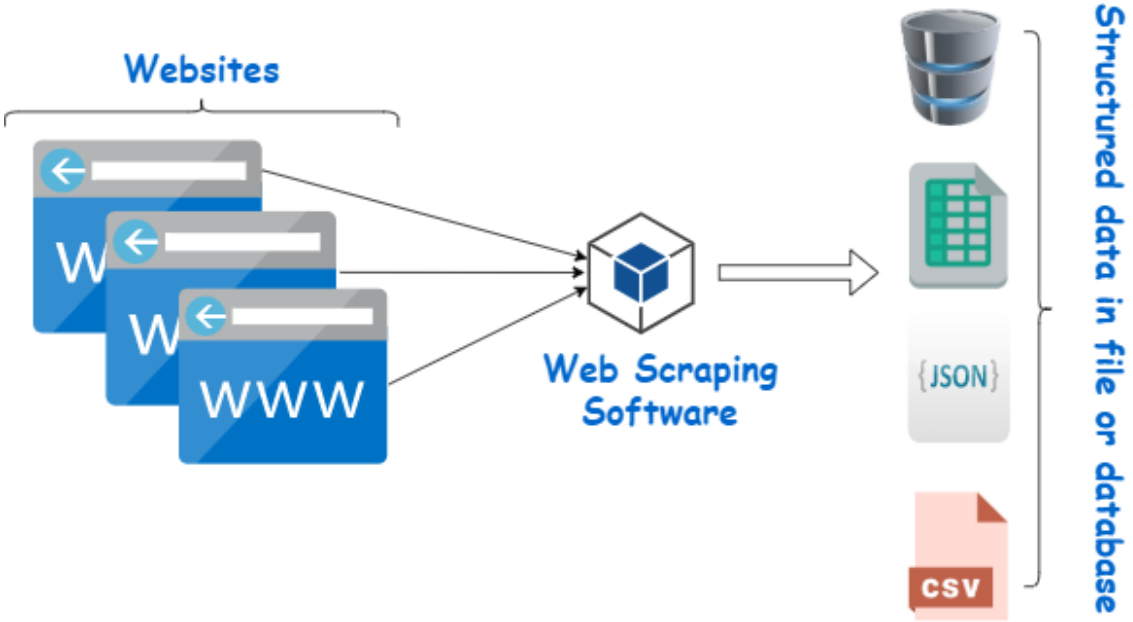

Figure 4.2: A representation of web scraping from https://www.webharvy.com/articles/what-is-web-scraping.html

During testing, the modified scraper from Liljedahl and Nyquist was shown to be one of the most effective in collecting CMPs, it had a higher CMP return rate than the other scrapers it was compared to. The other scrapers that were investigated was the one used in Nouwens et, al 2020[21] and a self-made one using some of the same techniques of searching for keywords as those used in Hausner and Gertz 2021[13]. This scraper yielded 12 CMPs when tested on 20 popular websites. The number of actual available CMPs was 15, according to a manual inspection of the websites. 12/15 possible was a very high recall rate, however these was popular websites, and it was expected to get a much lower recall rate when completed on a larger number of less popular websites. I expect to have a higher recall rate on popular websites than less popular ones, as the popular websites have more visitors and risk more of not following clear regulation as for example containing a CMP. The other scrapers were all eliminated because they ei-

ther featured all of the strategies used in the modified Liljedahl and Nyquist scraper [17] but didn't employ all of the other techniques and/or had a lower recall and precision. From now when the text mentions the scraper it refers to this modified scraper from Liljedahl and Nyquist 2021.

The features of the CMPs that are of interest is:

- The text of the CMP
- The optional "settings" / "learn more" page
- A consent button the CMP and the "settings" page if available
- A decline/reject button on the CMP and the "settings" page if available
- If it is used checkboxes whether any of these start in a "checked" position

Before the features can be located however, the scraper must locate the CMP. Locating the CMP may be difficult, as it is no standard way of setting a CMP up on a website. With 20 different web pages we can have 20 different ways they create the CMP for their website. Therefore, there is bound to be some CMPs, that the scraper misses, and the results are of a subsection of the number of websites that were initially set as it does not manage to gather all the available CMPs.

The scraper is built on python code and uses splinter to create a selenium browser object. Splinter is a "tool for testing web applications (...) it lets you automate browser actions, such as visiting URLs and interacting with their items"[2]. Splinter provides useful functions with the elementAPI and driverAPI which allows the scraper to locate a checkbox and whether it is checked. Selenium is the browser automation tool[3]. Selenium is what controls the browser and can create the browser element while Splinter provides useful functions of what one can extract from a website.

The scraper connects to the MongoDB database I have set up to store the extracted data. MongoDB is a type of noSQL database. Non-relational, distributed, open-source, and horizontally scalable are all characteristics of the NoSQL database management system [3]. MongoDB is used for its ability to store features with embedded features all in one data entry. For example, we have a feature called ”notice' which then contains seven other features one of them is ”moreBtn” which contains 10 features, where some of them also contains some features. With all these embedded in a relational

[2] https://splinter.readthedocs.io/en/latest/
[3] https://selenium-python.readthedocs.io/

database it can be difficult maintaining order and extracting information, but in the NoSQL format one can extract all information with a simple query on a JSON object. For example If I wanted the text on the moreBTN I would write: query = database['notice']['moreBtn']['text']. Where database is the connection to the MongoDB database the data is stored.

The scrapers crawl consist of 2000 websites from the top 10 000 websites from Domcop 10 million most visited websites[4]. This list is created by The Open PageRank initiative[5]. The Open Page Rank initiative list was chosen as it provides a large quantity of websites that one most likely access. The list is not a list of which sites are the most popular however, and this shows when some of the highest rated websites are sites such as s.w.org which is a site for creating websites. This website is highly ranked as all websites that create a website using wordpress will have a link or a connection back to this website or a another wordpress website. I will not use all the 10 000 websites the site has available due time concerns on how long it would take to manually verify the CMPs that is returned. This could however be done in the future with help of services as Amazon turk.

The first method the scraper uses, is to locate a CMP is by searching for all strings containing the word cookie using a xpath expression. This is a similar method that Hausner and Gertz 2021[13] uses, however they have not released their finished product yet so it is unknown which keywords they use. A Xpath expression is a method of selecting/pointing to a value that matches your query. Xpath allows for better navigation and referencing in XML-like files, this includes HTML which websites uses[6]. When the scraper has located elements with the word cookie or words with cookie as part of the word. The scraper will start searching through those elements parents looking for those elements that have a CSS position of 'fixed' regardless of scroll. This can locate a CMP since if it contains anything with the word cookie and has the position fixed, for most websites it will be about the cookie policy. Unless there is a bakery website that for some reasons have the need to have you always read that they do sell cookies on a fixed position on their website regardless of how you scroll.

Second method the scraper uses, is to search for a approve button in English, if found it looks for an HTML element that is double the size of the button + 50px. This is similar to the first method but instead of cookie it is a synonym for accept. The method

[4] https://www.domcop.com/top-10-million-websites
[5] https://www.domcop.com/openpagerank/what-is-openpagerank
[6] https://developer.mozilla.org/en-US/docs/Web/XPath

will check if it can find a div element that contain a accept synonym, and the parent element must be big enough to have room for the accept button and another element that is similar to the accept button. It is set up this way to find an element with an accept button and room for a reject button. This is an effective method for locating CMPs that is not a full banner and that do not mention “cookie".

The third method the scraper uses is the same method as the first, but from the the “cookie" elements it searches for parents with the width of the full page. This is for locating the types of CMP’s that are banner-like on the bottom and sometimes top of the websites. This is useful for CMPs that are implemented more as a part of the website and not as a pop-up.

The fourth method uses a AdBlock add-on in the browser. AdBlock is a browser extension made for blocking ads when you visit websites, but they have also other functions for CMPs[7]. AdBlock have a setting that attempts to block and decline all CMPs. This feature uses a list of often used CSS tags and ids for CMPs. This method searches if any of these tags or ids are present and visible on the website it currently is visiting. If some of them are and has a visible feature to the user, it extracts that feature and tries to screenshot it. This is a good method for locating well known CMPs.

The scraper also tries to locate the accept, deny and settings buttons. For accept and deny the scraper searches through elements of types: “button", “input" and “a", and compares their text element with that of a list of synonyms that the button can have. The lists consist of synonyms and words frequently used to represent “accept", “decline" and “learn more / settings".

If the scraper is able to locate a CMP or the settings page, but not a accept or decline button, there is an backup method added in the analysis step in the analyzer program that uses the screenshots the scraper takes. With the screenshot it applies an Object Character Recognizer algorithm (OCR) which is explained in the preliminaries chapter but is a machine learning method for recognizing objects from images. The objects being text in this setting. Then from the string that results from this the analysis program will search for synonyms using the same list the scraper uses accept and deny.

To locate the "learn more" or "settings" page we have two approaches. One looks for synonyms for "settings / learn more" and then follows that button when it leads either to a drop-down menu or another page. The other searches for a link containing either, cookie, policy or learn more.

[7] https://getadblock.com/en/

4.2 How the analysis program is implemented

This section will cover how the analysis part of the project is implemented. The analysis program is written in python code. Python was selected as that was the language, I had the most proficiency in. I used Jupyter notebook as it works well for performing data science tasks by allowing the user to run selected parts of code independently of other parts.

4.2.1 Libraries used by analyzer

The analysis uses nine libraries to perform its tasks of extracting the information stored from the scraper, performing object character recognition and other database comparisons and functions to classify which dark patterns are present on a CMP. The libraries that play a key role is explained in the following paragraphs.

4.2.2 pymongo

Pymongo and the submodule called MongoClient[8] are used to connect to the database setup in MongoDb. Pymongo is the official python driver for MongoDB and is needed for python application that interacts with a MongoDB database.

4.2.3 Pandas

Pandas and its DataFrame module created by Wes Mckinney [32] provides tools and data structures for data analysis. DataFrame is used as a data structure with vast amounts of useful commands. Pandas receives continuously updates and it is pandas version 1.3.4 that is used in this thesis. Pandas will be used as its tools and data structures fits well for the data, and the data manipulation needed.

4.2.4 Pytesseract

Pytesseract is used to perform the OCR and as the module uses a Recurent neural network (RNN) with long-short-term memory (LSTM) to identify the text from the CMP and the setting page of the CMP. An explanation of RNN and LSTM can be found in the background chapter.

[8] https://pymongo.readthedocs.io/en/stable/index.html

4.2.5 numpy

Numpy is imported which is library which focuses on performing numeric calculations with python[9]. It is used in the program for it capabilities to perform intersection between two lists for the classification.

4.3 The analyzer

The analysis program begins by creating a connection to the MongoDb database where the data was saved by the scraper. Using pymongo, a connection is created. After establishing a connection, it pulls data from the same MongoDb collection that the scraper provided the data to and retrieves them in JSON format. JSON "stands for JavaScript Object Notation and is a lightweight data-interchange format"[10]. Data-interchange format means that the data is stored in rows and columns[11].

When the program has obtained its chosen data, mostly textual features and screenshots. It utilizes OCR to retrieve the text from the screenshot. From the OCR, a lengthy string of text is retrieved. The application then utilizes a regular expression (regex) to tokenize both the text from the notification that the scraper collected, and the text taken from the screenshot. The tokenized text is then stored in an array. The program then intersects the tokenized text arrays with arrays containing synonyms for accept and decline. This returns a highly efficient way of discovering if any of the CMPs contained a option to decline or accept. We then know where and which word the CMP has used from the intersection. The arrays of accept and decline contain 18 words that where either known for being used in CMPs or known as synonyms for accept and decline. Due to the fact that certain websites force the language to be the same as where the user's IP-address originates, regardless of browser language settings. The list of synonyms required Norwegian synonyms as the CMPs of these webpages were written in Norwegian. This method of locating accept and decline, although effective will also causes some misclassifications. When the CMP of the websites write about decline while not offering a decline option, this method will wrongly classify the CMP of containing a decline option.

For the antonyms check, the program uses the words that have been identified in use for accept and decline from the method described in the paragraph above. Then

[9] https://numpy.org/about/

[10] https://www.json.org/json-en.html

[11] https://www.ibm.com/docs/en/personal-communications/12.0?topic=types-data-interchange-format-files

compares them with a created dictionary where each word of the synonyms used have three associated antonyms. If either one of the antonyms have a match with either of the options in the text it will be classified as using antonyms.

Having both the option to decline and accept in the initial page of the CMP, will be much more frequent now that both Google and Facebook was fined 150 million euros and 60 million euros respectively for not having both decline and accept in the first page of the CMP[12]. They were fined by the Commission Nationale de l'Informatique et des Libertés in France. With this reinforcement it bears resemblance with the planet 49 court case[13] and may cause a shift in how cookie CMPs look. The resemblance being that there is an active enforcement from a government in how the design of the CMP should be. Now that it is being enforced to have both accept and decline options on the first page of the CMP we will only consider the CMP of using antonyms if both options are on the same page.

For widget inequality, the program uses the button sizes that the scraper has extracted and compares if the decline option, that can either be in the settings page or the first CMP page, is bigger than 90% of the size of the accept button. The pre-checked checkboxes, for the unlabeled slider dark pattern type is information from the scraper. For the scraper it will look if there are any pre-checked checkboxes inside the CMP element and if it is classify that as the dark pattern type.

Features that was considered but were eventually not used include the CMP's full text ontology, the x/closing button, and the background color of the CMP. The CMP's full ontology that is the full text of the CMP was speculated that could be used with machine learning to predict whether a CMP contained dark pattern or not. The hypothesis was that the CMPs that would try to trick the user into just clicking accept would have a long and advanced text with a more complex vocabulary than those that did not. It was however dropped as it became unclear why it was needed to use more computation for a feature that would at best say there was a dark pattern present and that the text was complex. It can however be useful when stricter regulations of CMPs become more apparent forcing all CMPs that uses third party cookies to have an accept and decline option present on the first page or equally easy to decline as accept. However, I would hypothesize that the text would be easier and more "joyful / happy" to trick the user into accepting rather than overwhelm the user with complex language, when the user

[12] https://www.cnil.fr/en/cookies-cnil-fines-google-total-150-million-euros-and-facebook-60-million-euros-non-compliance

[13] https://curia.europa.eu/jcms/upload/docs/application/pdf/2019-10/cp190125en.pdf

can easily decline. We can in part already see this in some websites for example from hostgator.com which uses "got it" as an option to accept. This is a unformal way of presenting a legal decision. See figure 4.3.

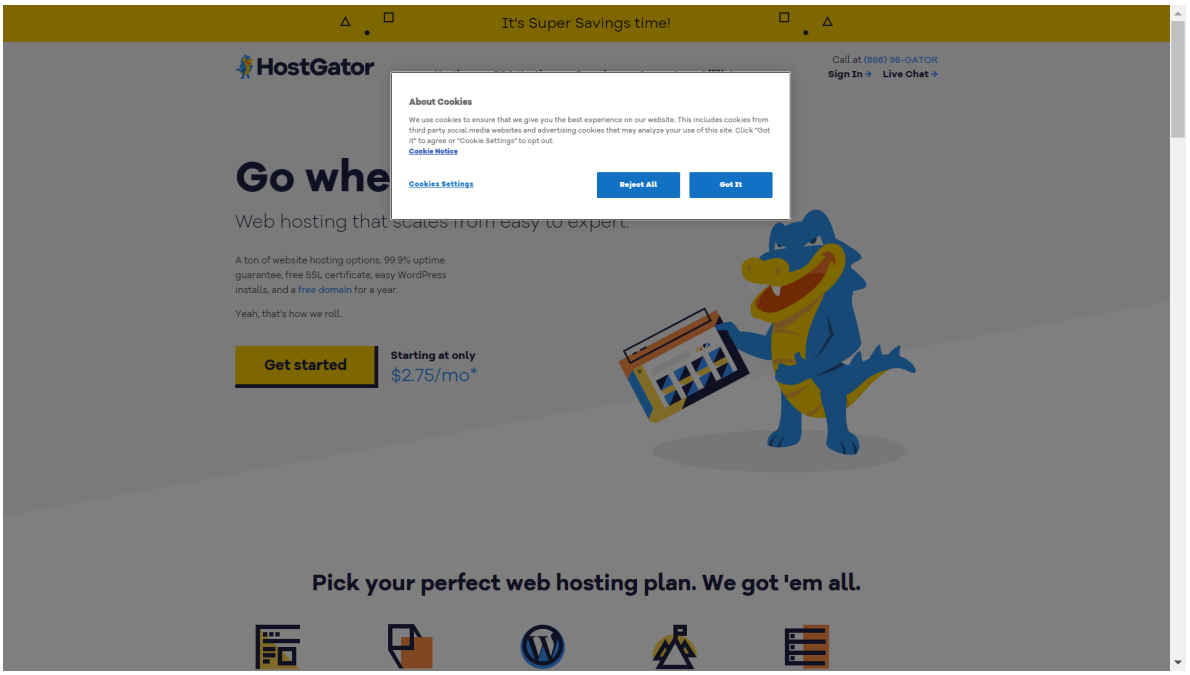

Figure 4.3: An example of a unformal CMP with a accept and decline option on the first page https://www.hostgator.com/

The x or close button was dropped as it became hard to get a method that managed to extract the button in a reliable fashion. Methods that searched for class =’close-button’ or buttons with close in its name did not give any adequate results as it rarely returned the button. The problem of how to interpret close button was also brought up and discussed earlier in this thesis. Background color was dropped early as many CMPs will have the same background as the website and if they did not have the same color, how was that to be interpreted became a problem. It was considered a feature since, if a website attempted to hide the CMP by having it blend in with the background. The website could interpret that if the user continued to use the website, they had given consent. Such a website described would have to be classified as using a dark pattern. However, if the CMP and the website shared a color, example "white" then one could not conclude they were attempting to hide the CMP rather than have it mix in with the design of the website.

The program from scraper to the analyzer was run on 2000 websites from the open page rank initiative the 12th of April 2022. The scraper took roughly 16 hours to complete on a computer with 8 GB RAM and an intel core i5 CPU. The analyzing part

of the program was only run on the CMPs the scraper found and took roughly three hours to run. The scraper returned 629 unique CMPs from the 2000 websites.

Chapter 5

Evaluation

In this chapter the result from the analysis program is evaluated with statistical measures and the incidents that caused misclassification are presented.

5.1 Scraper evaluation

From the 629 CMPs returned from the scraper it was manually investigated whether the websites actually did contain the CMPs or if it was a false positive instance. Each website was accessed in an incognito google Chrome browser window which is what the scraper also uses. It uses incognito mode which is a mode available for most browser such that it does not store any cookies from one website to another. The URLs of the websites along with their data from the checker program was entered in a csv spreadsheet with information from the manual investigation and can be accessed on this GitHub repository explained in the appendix. The manual investigation checked the screenshots from each website that the scraper took as well in case, the website had updated their CMP from the time the scraper ran, to the time the investigation took place. The investigation stored whether it was an actual CMP there, what kind of dark pattern is prevalent in the CMP, and if it was from Google. Checking if it was from Google may seem like an odd feature, but it was early in the investigation noted to be very useful as Google had many different websites present on the URL list, which all shared CMPs which where misclassified. It became more important as the checker would manage to correctly classify the new CMP Google will roll out soon after the CNIL case mentioned earlier. The new CMP can be seen in this footnote [1].

From the investigation of the 629 CMPs returned from the scraper, the actual number of CMPs is 560. That means there are 69 false positives. Using the statistical measures

[1] https://blog.google/around-the-globe/google-europe/new-cookie-choices-in-europe/

from the preliminaries chapter we would ideally calculate recall, precision and F1 score to verify how accurate the scraper classifies that there is a CMP present. The precision is $Precision = \frac{560}{560+69} = 0.890$. However, due to time constraints it was not possible to manually verify all the 2000 websites visited meaning we could not calculate the recall and F1-score of the scraper as we did not have the number of false negatives. That is the CMPs the scraper did not discover.

There was performed an investigation of websites where the scraper did not find any CMPs. As the scraper also takes a screenshot regardless of there is a CMP there or not, it was possible to verify how accurate the scraper was in extracting CMPs. A random sample of 101 websites from the 1 372 websites the scraper could not find any CMPs in was selected. However, it was quickly noticed that some of the main problems apparent was websites that blocked the scraper or took too long to load the CMP. From the 101 randomly sampled websites 14 CMPs was found when visiting their websites. However, when investigated with the screenshot the scraper took it turns out that for 12 of these the scraper was either blocked access, or there were no CMP present on the scraper's screenshot. For the remaining two that had a CMP present in both the screenshot and the manually check, one of them was in fully German and the other the scraper missed for unknown reasons. They can both be seen in figure 5.1 and figure 5.2.

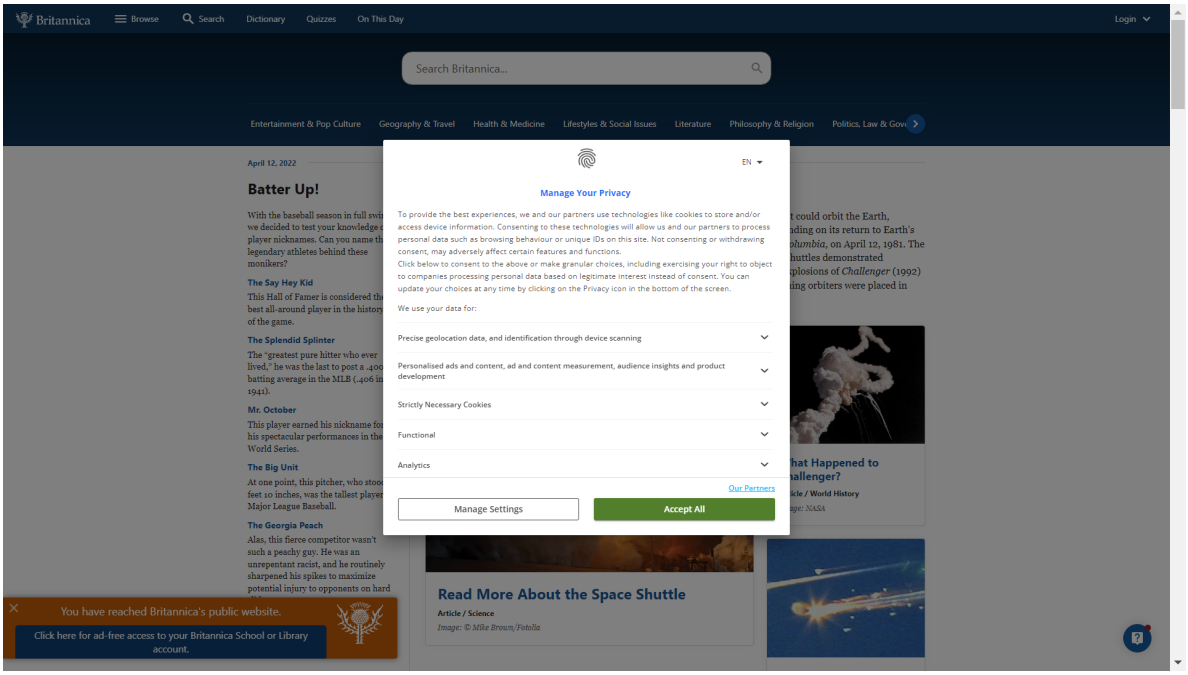

Figure 5.1: Example from https://www.britannica.com/ unknown why the scraper miss this

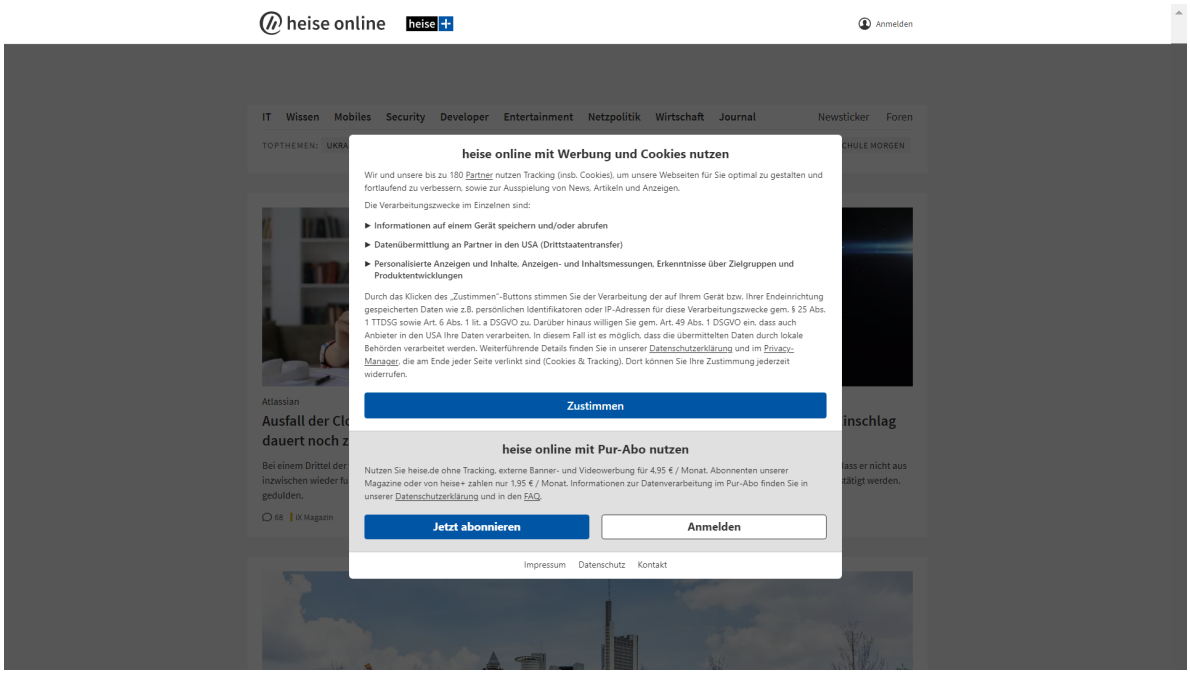

Figure 5.2: Example from https://heise.de/ of the scraper missing this CMP as it does not trigger any of the words the scraper searches with

There were discovered multiple causes for errors among the 69 false positive cases. One prevalent cause was pages with a link to their privacy and cookie policy on a fixed spot either on the bottom or top of the page. This set up was appearing on all the American government pages and there were 25 incidents from American government related pages alone. Unfortunately, there were also discovered CMPs which had not been detect had it not been for the method that locates privacy or cookie policy links with fixed positions. See figure 5.3 for an example of CMP that had not been detected and figure 5.4, for an example of a misclassification. One solution for this problem presented here could be to require a CMP to contain at least one option for "accept" or "decline" for it to be considered a CMP. In this situation we have to either, settle by excluding some CMPs and have more false negatives or include websites without CMPs and have more false positives.

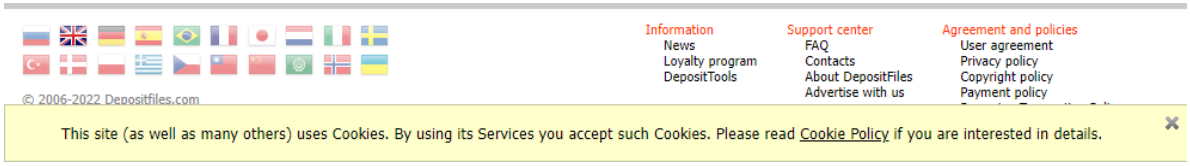

Figure 5.3: Example from https://depositfiles.com/

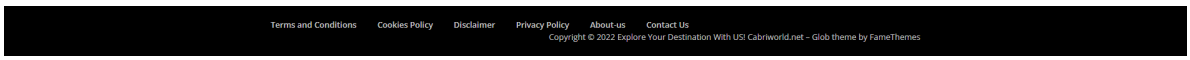

Figure 5.4: Example from https://cabriworld.net/

There are also a few instances where there were not CMPs, but the website wanted to use a notification feature of the browser which requires the user to manually allow a pop up. An example of this misclassification can be seen on figure 5.5. This can however be prevented in the future as there is possible to exclude such pop ups, but this was not done here due to time limitations and a larger focus on the analysis of the dark patterns rather than a focus to improve on the scraper.

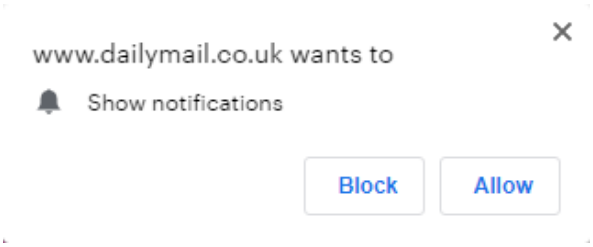

Figure 5.5: Example from https://dailymail.co.uk/

5.2 The analysis evaluation

Before the removal of the confirmed false positives from the data, the checker returned there was 238 types of choice cascade, 116 types of no choice, 428 types of there not being used antonyms, 21 types of there being pre-checked checkboxes and 17 types of widget inequality. After false positives are removed there are left 208 cases of the dark pattern type Choice cascade, 162 types of no choice, 362 types of no antonyms used, 62 cases of pre-checked checkboxes and 15 cases of widget inequality from the 560 CMPs. From the data it is apparent that the accuracy score is higher than the

Table 5.1: Statistics with Google related CMPs

Type	Accuracy	Precision	Recall	F1-score
Choice Cascade	0.63	0.55	0.57	0.56
No Choice	0.74	0.65	0.42	0.51
No antonyms	0.83	0.95	0.82	0.88
Pre-checked checkboxes	0.92	0.53	0.66	0.59
Widget inequality	0.99	0.87	1.00	0.93

others. This comes from the checker being much better at classifying those that do not contain the dark pattern type than those that it thinks does. The true negative is only used in the calculation of the accuracy. From the data it is apparent that the no antonyms and widget inequalities has the highest accuracy. These patterns had a strict definition. With this strict definition and the nature of the pattern types, it is simple to distinguish between those that contain the pattern and those that do not. To check if the CMP used an antonym, it was clearly defined how accept and decline could look.

It also helped that most website that had an option to decline had it on the settings page. 560 CMPs - (231 true incidents of choice cascade + 181 true incidents of no choice) = 148. This gave it only 148 potential websites where there could have been potential for it to wrongly classify the pattern. Widget inequality was a pattern harder to manually investigate as the size difference was set to 10% it was hard to spot if the accept button was 10% or bigger than the decline button. This posed a dilemma as it was such barely noticeable, it could be better to have a higher threshold. This was however expected as the size threshold for it to be a dark pattern or not will require more research. A discovery was that often the word used for "accept" is larger than that of decline. Example "approve" and "deny".

Features that caused misclassification, where often text based. A good example is in stagram.com which the checker classified as having "no choice" and "no antonyms". It was classified as such since Instagram have two options on their CMP See figure 5.6. Instagram use "Only Allow Essential Cookies" and "Allow Essential and Optional Cookies". So, the trigger for a decline option never triggers. This way of posing the options is deceiving in a variety of ways. From having the accept button start the same way as the decline option, letting the accept option be in a bright blue with a bold font while the decline option is the same color and style of the text explaining the cookies above. This can make the option be easily mistaken for just some text of the explanations of cookies and not an option of itself. This is however not a standalone CMP and there are multiple of these CMPs with two options of "only allow". Other terms used is "use essentials", "proceed with required" or "use necessary". These are bi-terms and with more time and a bit more natural language processing some of these could be classified as a decline option. This gives room for improving the false positives for the pattern "no choice" and the false negatives for "choice cascade".

The world of CMPs moves fast. From running the scraper to coming around to verify the results from the checker, only two weeks later, multiple CMPs where different. These are among others Youtube and Linkedin. Youtube is under ALPHABET, Youtube and Googles parent company which as explained earlier got fined by the CNIL for their CMP[2]. Linkedin did not get fined but changed their CMP so that it too follows the regulation that CNIL enforced on Facebook and Alphabet. There is expected many more websites will soon follow and provide an option to decline in their first initial CMP page. Alphabets new CMPs have been tested with the program in this thesis and classified by the checker correctly, as not containing any dark patterns. However, this

[2] https://www.cnil.fr/en/use-google-analytics-and-data-transfers-united-states-cnil-orders-website-manageroperator-comply

research uses how it classified the CMP present at the time of the 12th of April 2022 which were not changed yet.

5.3 Discussion

The decision to use Open Page Rank Initiative over other more popular website rankings such as the Alexa top websites[3] was decided as Alexa have more non English websites on their top 2000. It was also discovered that Amazon will close the service 1st of May 2022[4] which may have caused problems for reproducing the work in this project. A problem with the Open Page Rank Initiative became apparent in the investigation was the way it ranks websites. As it is not entirely dependent on popularity but how much it is linked. Therefore, there were some websites that looked quite old and that probably has less than 2000 visitors monthly. Some of these websites have unique design that may cause some issues for the scraper. As can be seen on figure 5.7. On this website there is not a CMP available, but the website is triggered as one with a CMP due to the fixed sized banner and later down on the site there is hyperlinks with text of "policy" and "continue reading".

An issue that can be raised against the checker is how it only uses three antonyms for each of the synonyms for accept and decline. This was a decision based on practicality, and to have a stricter threshold for what is an antonym or not. The stricter threshold helps differentiate between the CMPs that uses clear antonyms for accept and decline, and those that do not but have a decline and accept option. With a too soft threshold on the number of antonyms one can end up classifying every CMP that have a accept and decline option of using antonyms which would make the type obsolete. As it would be equal to "choice cascade", informing the user if the decline option is behind a settings page. At least how it is defined in this thesis where both accept and decline must be on the initial CMP page. However, having it too strict with few synonyms and antonyms we may end up not locating the "accept" and "decline" option. It is believed that if we included some more synonyms and increased the number of antonyms for each to five, we may have increased the accuracy scores.

Another problem for the checker is websites that write about rejecting the cookies in their CMP while not offering a proper reject option on the CMP. If the text available to the checker contain a synonym for "reject" it will conclude that there is reject an option. This causes a problem since if it were to require the reject option to be on the decline

[3] https://www.alexa.com/topsites
[4] https://support.alexa.com/hc/en-us/articles/4410503838999

button it would require a scraper that was much more accurate in extracting the decline button for those websites that have it available. If the checker instead required there to be more than one reject synonym, the analyzer would miss a large portion of those that have been classified of having a decline option and verified as many CMPs only write a synonym for reject once, and that is on the reject button. Unfortunately, some of the CMP type providers that only write about the reject option is Google. This is unfortunate as Google have 113 related CMPs in the extracted CMPs from the scraper. These websites follow three different, but similar CMP set ups. The CMPs have a misclassification of having a reject option in the first CMP page which is an antonym. This classifies then the CMP for the most common Google CMP type as not containing any dark patterns. I have highlighted "reject" in one of Googles CMP in figure 5.8, do keep in mind however that this will soon go out of date as by the court ruling that happened in France.[5]

The other CMP set ups Google uses do not have an option to reject and should be classified as a no choice. On these CMPs Google follows a trend with CMPs of only informing the user that they use cookies, and when the user clicks on their cookie policy they are sent to a long page with "legal speak" with no option to decline. We can see this from the Figure 5.9. Here I have highlighted Google, but this type of just informing the user before sending them to a legal page was an overarching trend and one of the main reasons the patterns "No choice" and "choice cascade" had such low score. This can potentially be solved by being stricter with the definitions and requiring all potential options being on the initial CMP page or when entering a "settings page" requiring more incidents of "reject" synonyms.

There are websites that do not offer any save or reject options on their "settings" page but pre-checked checkboxes or sliders. how are these websites going to be interpreted? For example Figure 5.10 from Linkedin. As can be seen they do start with un-checked checkboxes which is required but uncertainty arises when there is no option to either save or confirm those settings. If you enter the page again are those settings saved? If you click "back" in the top corner, do you have the CMP again? In the current setup for the checker, it classifies these as having "no choice" and "no antonyms" as it cant locate a decline button. In one way that is correct as there is not a reject button but instead you have to see to that the checkboxes are un-checked. However, it can be argued that in a way the user has the option to reject by clicking on to this kind of settings page then entering the website again if it then stores the user as having rejected. This is however

[5] https://www.cnil.fr/en/cookies-cnil-fines-google-total-150-million-euros-and-facebook-60-million-euros-non-compliance

uncertain and for this thesis we classify these websites as “no choice” but some may argue otherwise.

What is the perseverance of the different dark pattern types for CMPs over time? If we are to learn from the past, we should expect these patterns to evolve over time. In fact, to be so bold some of these patterns are already up for a change. The pattern for unlabeled sliders or pre-checked checkboxes I would argue that at minimum for a user from the EEA or EU area they are in decline. If we investigate the data when there are pre-checked checkboxes from the analyzer, we see that, they are pre-checked, but it has pre-checked options for using first party cookies and have a clear reject button. For example, figure 5.11 from WeTransfer. Here we see there are three pre-checked checkboxes which are all on using first party cookies, and that there is a reject option that will reject all at the top. The decline of the popularity of this pattern can likely be caused by the Planet 49 case which thoroughly marked that having pre-checked checkboxes, the user would have to check out from, is prohibited. Another common way this pattern appears is as shown in figure 2.5 where it has prechecked the option for only using necessary cookies. We can likely suspect that most websites will also have a reject option on their first CMP page after the CNIL case against Facebook and Google in a year as that is what has happened with the planet 49 case.

From the pre-checked checkboxes pattern originally called “unlabeled sliders” there is the ambiguity of how to interpret the CMPs that have checked for “legitimate interest”. There is uncertainty in classifying these as it is uncertain what the website will say can be accounted for “legitimate interest”, but they are allowed to have these. However, to have the “legitimate interest” checked, one of the requirements from the GDPR regulation is that “ organizations that use it must thoroughly justify it in their documentation”[6] If this requirement is followed up on and how this is enforced should be researched. For this research they are classified as the dark pattern pre-checked checkboxes.

An issue that was discovered during the manual investigation was that even if the scraper’s browser language was set to English there where multiple websites that gave the CMP in the language of the website or the language from where the IP-address of the user was (Norway). A solution for when the CMPs was in Norwegian was fixed by extending the vocabulary used by the scraper to contain Norwegian synonyms and antonyms, but for the other languages it was not implemented due to time limitations

[6] https://www.itgovernance.eu/blog/en/the-gdpr-legitimate-interest-what-is-it-and-when-does-it-apply

of adding synonyms and antonyms to multiple languages the researcher did not know. This gave a significant hit on the accuracy scores as well as there was detected at least 33 websites with a different language from Norwegian and English in the 560 CMPs the scraper found. From the 1 361 the scraper did not find a CMP it can possible be many more CMPs.

The investigation stored which CMP was from Google and as the checker correctly classified the soon to be CMPs of Google. It was of interest to see how well the checker classified without a Google related CMP. This was calculated using the same statistics of accuracy, recall, precision, and F1-score in table 5.2. These scores are significantly better than those with Google related CMPs and may indicate how the accuracy of this program may evolve over time as the CMPs change.

Table 5.2: Statics without any Google CMPs

Type	Accuracy	Precision	Recall	F1-score
Choice Cascade	0.76	0.65	0.76	0.70
No Choice	0.76	0.65	0.53	0.58
No antonyms	0.93	0.95	0.95	0.95
Pre-checked checkboxes	0.90	0.53	0.68	0.60
Widget inequality	0.99	0.87	1.00	0.93

5.4 The websites blocking scraping

An interesting aspect that caused misclassification for the scraper is websites that blocked the scraper. Some try as Facebook where they will have a legal warning in both the robots.txt file[7] and will give you a pop-up warning in your scraper program see figure 5.12. However, these warnings have been deemed not accurate by US supreme court in a court ruling where Linkedin attempted to stop a rivaling company scraping information on users from their website[8]. It was ruled that web scraping is allowed and legal if the information is publicly available. The Robots.txt file mentioned is a file that is standard for websites to have to tell web crawlers and other automated programs on the web such as web robots how they should process the website and whats allowed and not allowed. Another way websites blocked the scraper can be seen on figure 5.13. The website sends the web scraper a Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA). A CAPTCHA is used to prevent access by computers as it is very difficult for them to interpret what is written in a CAPTCHA.[31].

[7] https://www.facebook.com/robots.txt
[8] https://techcrunch.com/2022/04/18/web-scraping-legal-court/

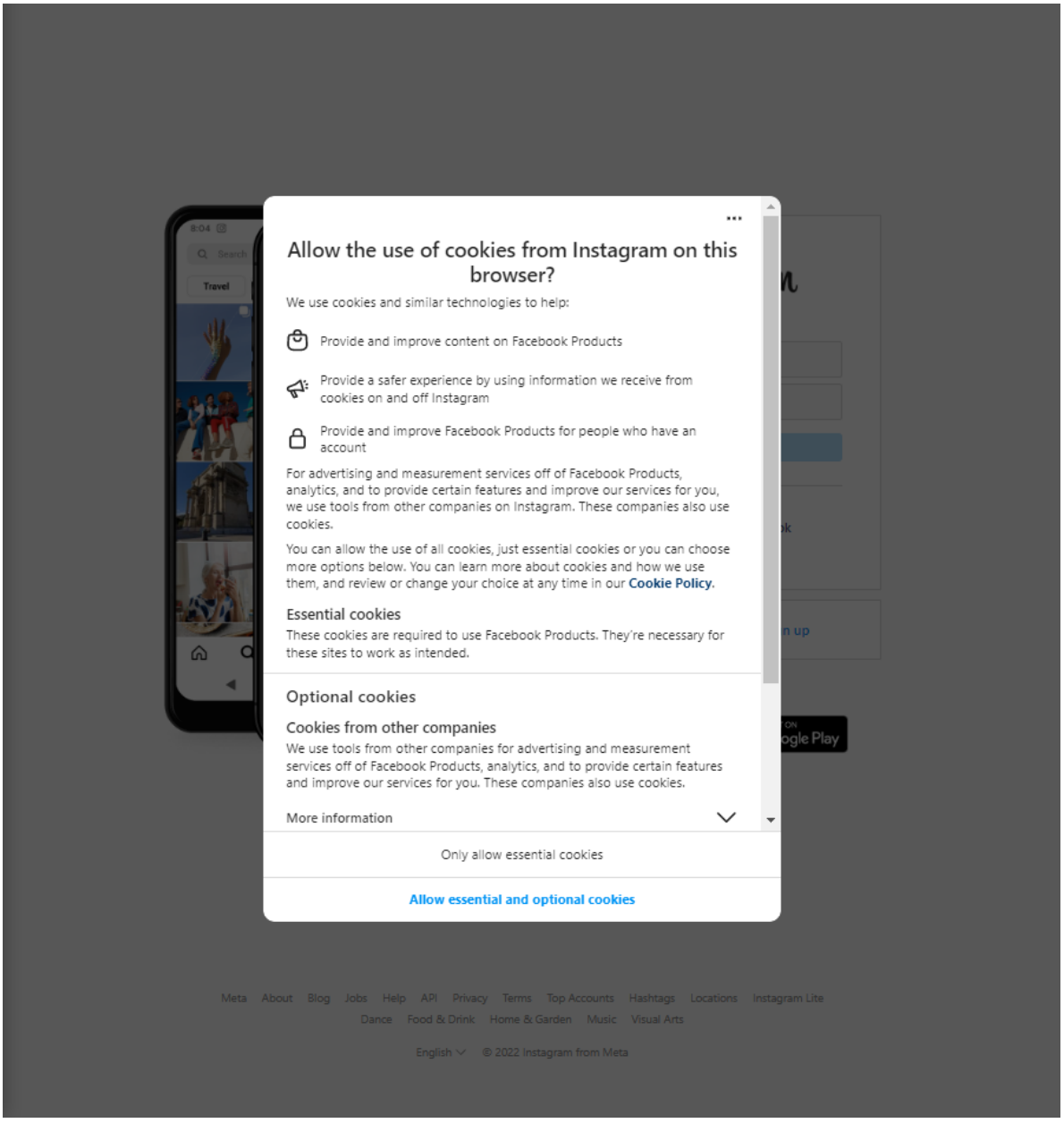

Figure 5.6: Example from https://instagram.com

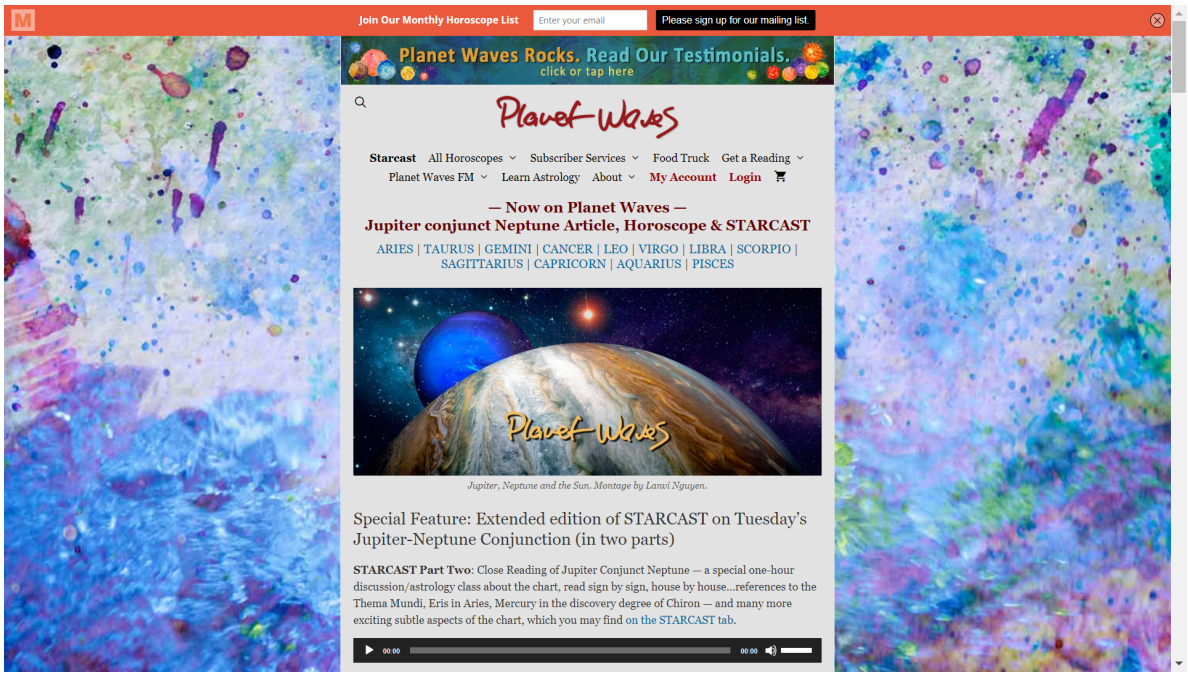

Figure 5.7: Example from https://planetwaves.net/

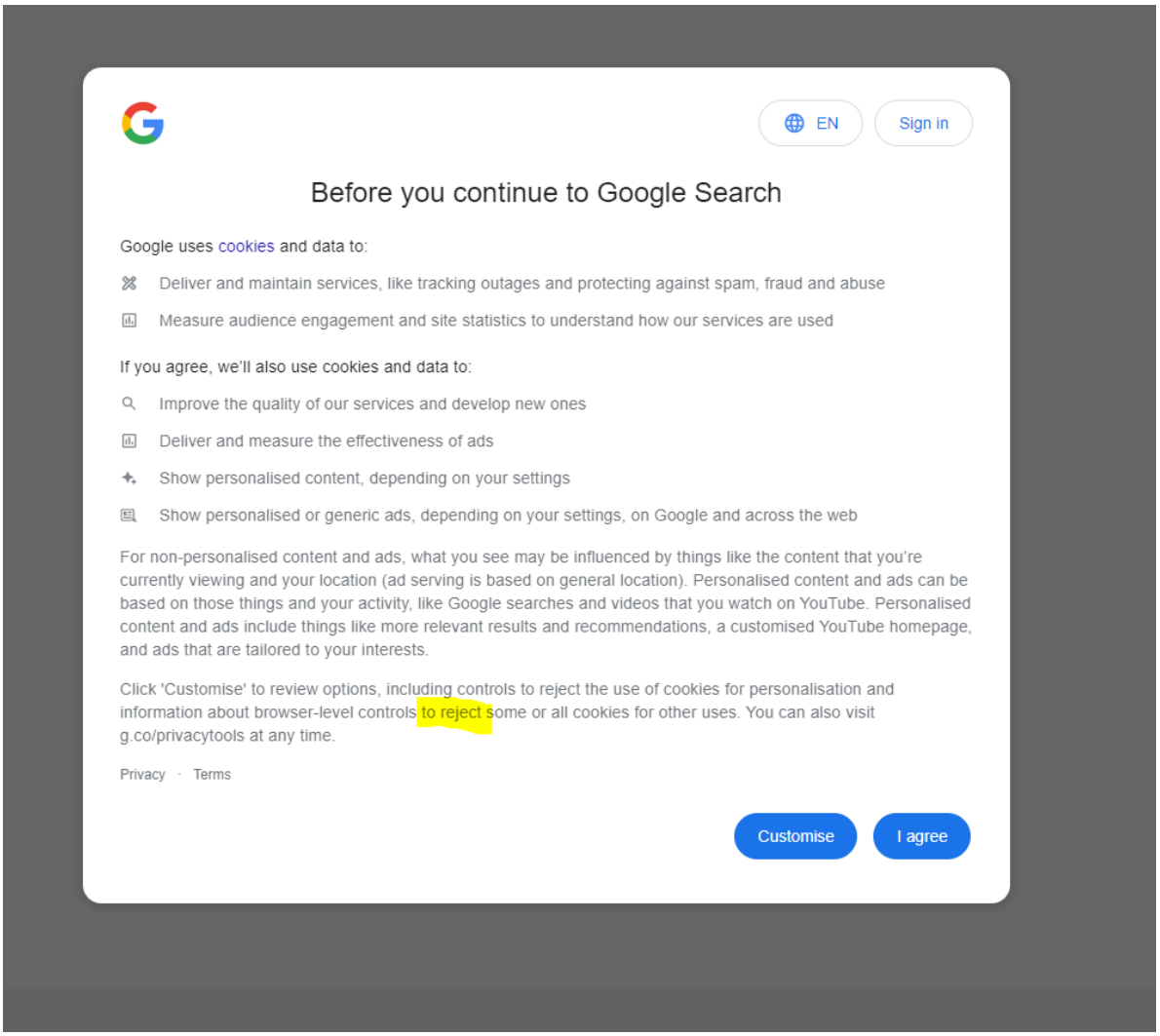

Figure 5.8: Example from Google.com

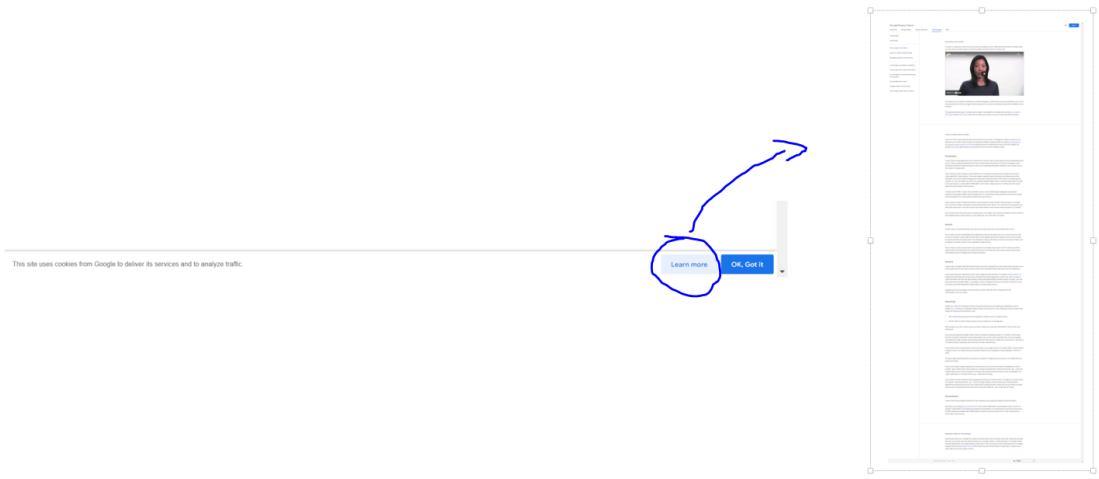

Figure 5.9: Example from https://ads.google.com/ If you click the Learn more button you are sent to a long page about cookies with no option to decline

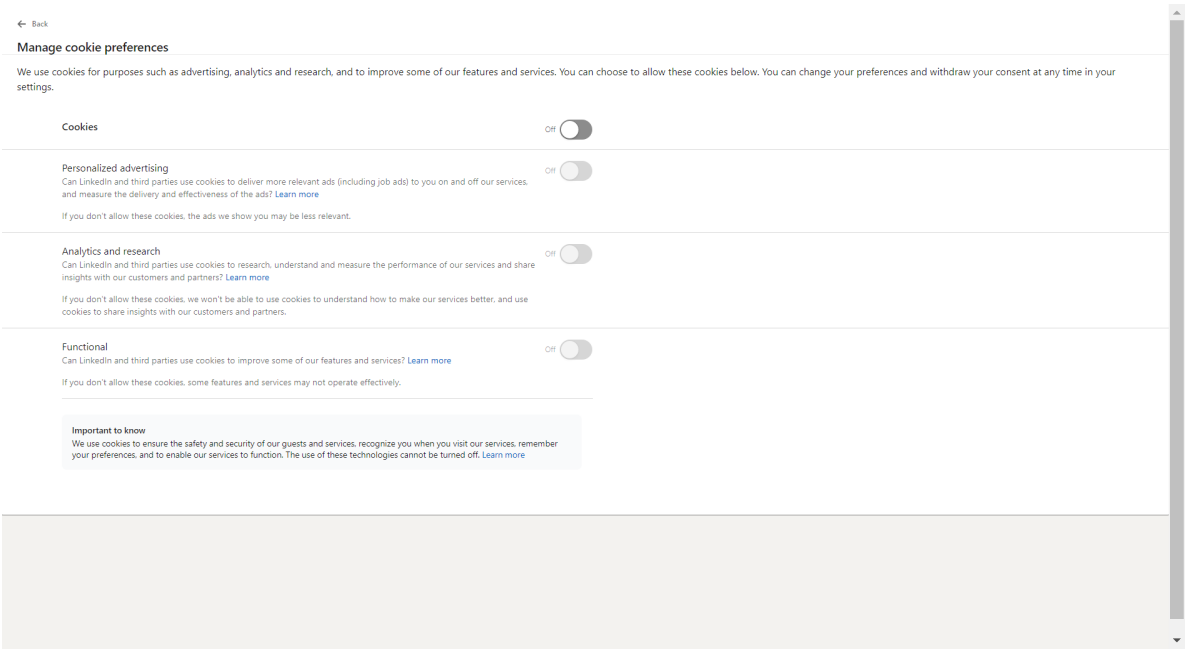

Figure 5.10: Example from Linkedin.com

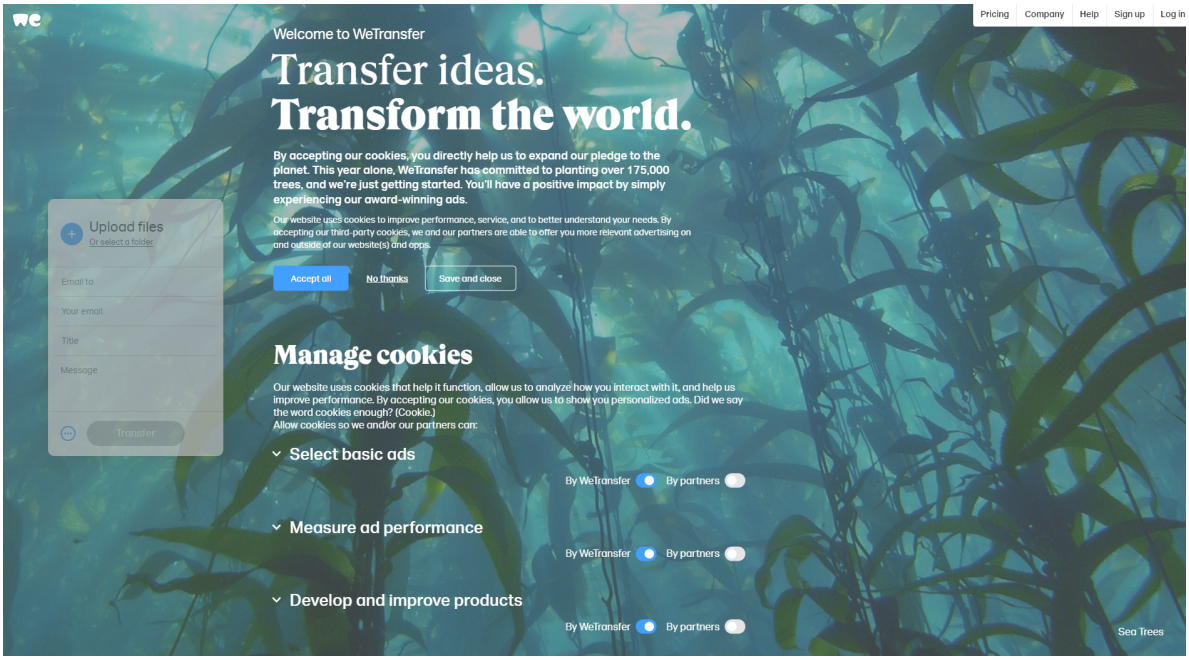

Figure 5.11: Example from https://wetransfer.com of pre-checked checkboxes

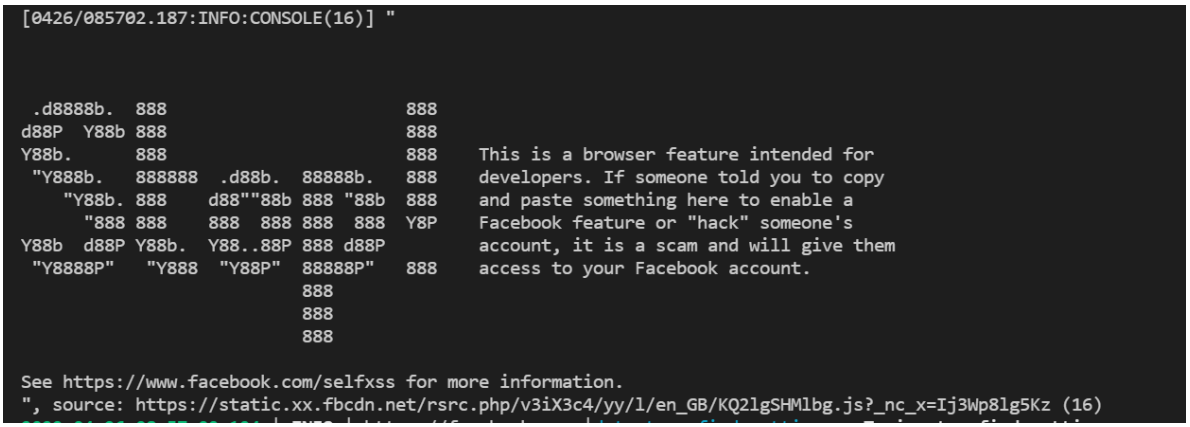

Figure 5.12: Example from the Scrapers terminal when it attempts to crawl on facebook.com

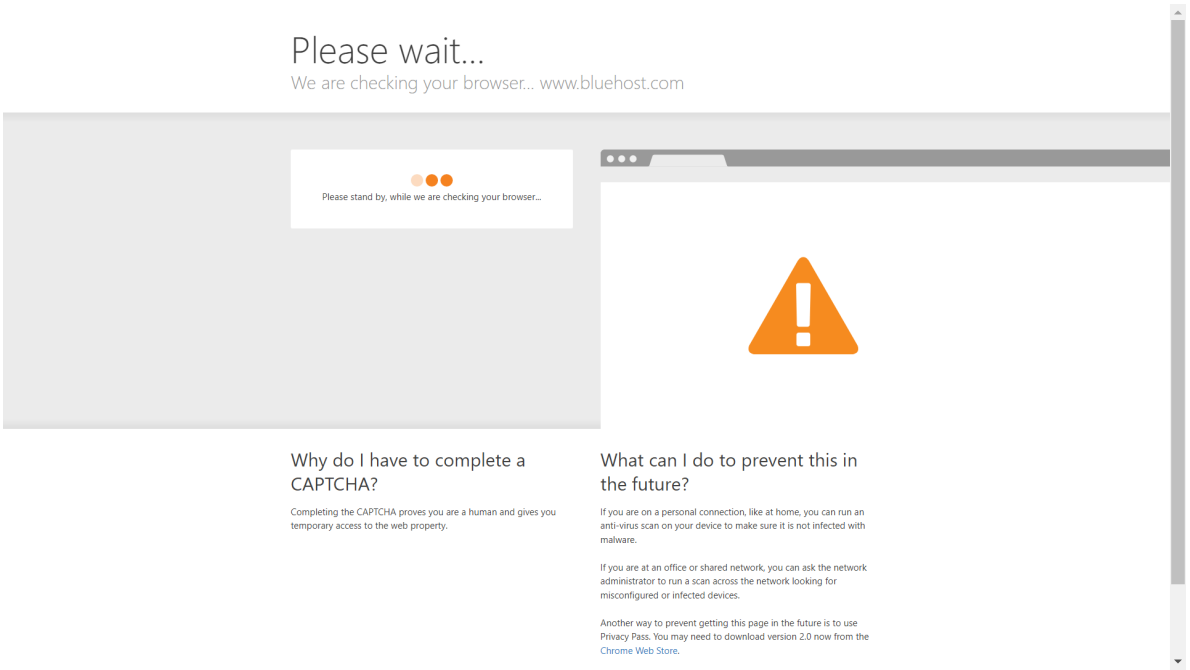

Figure 5.13: Example of bluehost.com blocking the scraper from scraping their website by giving the bot a human check.

Chapter 6

Related work

Utz et, al 2019 [30] conducted a field study on a German e-commerce website with a focus on how the CMP was interpreted by the visitors to the website. In the study they logged how the CMP was interacted with by over 80 000 unique users. Of those 80 000 they got 110 users to respond to their survey. They wanted to know if the position of the CMP, the number of choices, how a privacy policy link on the CMP and how the language of the CMP influenced users. Some other interesting insight their study found was that with the default choices, that should be opt-in according to the GDPR. It would result in less than 0.1% of users choosing to actively opt-in for allowing third-party cookies. Which is an interesting dilemma and may help explain why dark patterns in the CMPs are this prevalent as they are.

Nouwens et, al 2020 [21] considered how many CMPs followed three features which are implicitly given by GDPR. Those being: Is consent explicit, is the accept and reject option on the same widget or hierarchy (you need the same number of clicks) and lastly if there are checkboxes are they pre-ticked. They started scraping from a list of 10 000 websites and got fully scraped CMPs from 680 pages. From these 680 they found that the minority of them upheld all three features (80/680). They also did a study where they found that information that is not on the first available layer/widget is for the most part ignored by the users.

Matte et, al 2020 [19] focused their research on CMPS which complied with Europe's Transparency and Consent Framework (TCF). This narrowed their data set significantly but allowed them to better connect them to the GDPR. They then looked on what the website stored on the user. They ended up finding 141 websites register positive consent even if the user has not made their choice; 236 websites nudge the users towards accepting consent by pre-selecting options; and 27 websites store a positive consent even if the user has explicitly opted out. [19]

Hausner and Gertz 2021 [13] conducted a study as part of the larger dark pattern detection project between Heidelberg university and German Research Institute for Public Administration in Speyer[1]. They also focused on dark patterns in CMPs. Their aim is to be able to automatic detect the malicious patterns to protect the user from these. They have not per 25.03.2022 presented their final results, but in their paper[13] they present how far they are currently. Their scraper can extract 2800 CMPs from 4000 websites which gives them a higher than 50% return rate (only works for German websites). Then by extracting textual features from elements on the CMP which is clickable they use those features to perform unsupervised learning (clustering). Then after relabeling some critical items, they train a Support Vector Classifier to distinguish the different button options. Their end results are a classifier which automatically detects cases of aesthetic manipulation between the users choices.

Tahaei et, al 2021 [29] analyze code snippets from official ad-networks for mobile, and show how if they are used as they come, it will introduce dark patterns for the user of the app. These code snippets are among other things consent pop-ups that does not provide the user with a reject option also called a "no choice" dark pattern. Others are consent forms which will continuously annoy the user until they give their consent. This is relevant as it shows how these big providers (Google and Amazon) very openly implements dark patterns in their products. This can be of help if someone decides to take a deep look into the code the big CMP providers provide.

Mathur et, al 2019[18] focused on the five patterns from Gray et, al 2018[12] and crawled through around 11 000 shopping websites and discovered 1 818 dark pattern instances. The observant reader may notice they described 15 types of dark patterns from Harry Bringnull's work. This is because in a time period between their article in 2019 and Gray's article they added three extra types before removing them again as they were a type of sub elements of some of the types already there.

Bollinger et, al 2022 [6] take a more machine learning classification approach, when classifying CMPs and GDPR violation. First, they make important decision in their scraper with only targeting CMPs from distinct CMP providers such that they can better focus their scraper and achieve clean data. They scrape through 5.94M websites and find 37 587 websites with their type of CMPs. From these 37 587 they crawl a second time this time also entering subpages extracting cookies after they have consented to everything and found 2.2M cookies from these websites. From these they extracted

[1] https://dapde.de/en/

which cookies was used and of what type they were. (necessary, functional, analytics, advertising) They were then used to create a classifier using XGBoost which can classify what type of cookie a unknown cookie is. This model scored 87.2% accuracy in predicting what kind of cookie it is. They later show with this classifier that 69.7% of websites will set cookies that are not necessary if the user have not interacted with the CMP. (Implicit consent) They also found that 21.3% will still use at least one third party cookie which is originally rejected when denied consent.

Soe et, al 2020 [26] considered 300 CMPs and discovered that 297 of them utilized dark patterns to entice users to consent. When it comes to CMPs, the authors discovered that dark patterns are quite common. This thesis is based on their classification of various dark pattern types related to CMPs. In their research they manually labeled features of the 300 CMPs and in a later work Soe et, al 2022 [27] they trained a classifier using that dataset. Their classifier did use the dark patterns from [12] and had accuracies ranging from 0.72 on the pattern nagging to 0.50 for the pattern obstruction.

Liljedahl and Nyquist 2021 [17] aim was to create a scraper that could fully automate the analysis of CMPs against a set of parameters. In the author's thesis, they developed a scraper that combines four distinct approaches for identifying a CMP. Their analysis of the CMPs used features such as size, button size, pre-checked checkboxes, readability, how long the site store the cookies and if the site redirect the user that denies the cookies. Their thesis produced the scraper that besides some small modification is what is used in this research.

Gray et, al 2018 [12] developed and refined the emerging phenomenon "dark patterns". They noted that there was a difference between "everyday commercial UX design" and UX ethics related articles from the HCI community. They analyzed data from both the "hall of shame"[2] and exemplars collected from UX design practitioners and journalists. With this wanted to find what the overall design motivations and categories for why they were the way they were. From this work they created five categories of dark pattern, which are described above in this thesis. In a hierarchical sense these five categories are above the original twelve and one can derive those from the corresponding category.

Curley et, al 2021 [9] created a framework from work previously done on dark patterns and attempted to classify which of the original twelve dark pattern types can be

[2]https://www.deceptive.design/hall-of-shame/all

classified automatically (fully or partially), manually (fully or partially) or not possible. From the twelve they only classify the "Roach Motel" as possible to do fully automatically.

Papadogiannakis et, al 2021 [22] performed an in depth check on whether websites respect the users privacy when they reject or don't interact with a CMP by using advanced post cookie tracking mechanisms. The mechanisms they checked was browser fingerprinting, ID leaking, and ID synchronization. They found that over 75% of websites ignore whether the user deny the use of cookies. This is even higher in certain countries, such as in Czech Republic where almost 100% of the websites will use third-party cookies even if you reject all. They did find that it was better to ignore the CMP than interact with it in certain countries. [22] searched 15 354 websites and found that before the user had any chance of even interacting with the website 14 238 websites had started first party ID leaking.

Chapter 7

Conclusion

The research question of this thesis is: "is it possible to automatically detect dark patterns in CMPs from the web code?" To answer this, we need to solve three problems which we identified as sub goals in this thesis. They are:

- find a way to collect data from CMP's of websites,
- find a way to analyze the data to classify the different dark pattern types,
- evaluate how well the dark pattern automatic identification works.

I address them by first researching what dark patterns there are and what features they rely on in a CMP context. Thereafter an investigation on how to extract features that could be used to classify the different dark patterns was done with a comparison of three different scrapers. With the best selected scraper and some small modifications, it successfully managed to extract the wanted features from 560 CMPs out of 2000 websites.

I made a program that were to check which dark patterns that were present on each CMP. This program scored with an accuracy on each pattern from 0.76 for "choice cascade" to 0.99 for "widget inequality". However, the F1-scores is from 0.58 for "No choice" to 0.95 for "no antonyms". With the evaluation we can conclude that it is possible to build an automatic classifier for dark patterns regarding CMPs. The classifier presented in this thesis can be improved, but the groundwork for further research and a perfect classification method is paved.

An organization such as NOYB[1] may have use for a classifier as the one created in this research. NOYB currently uses software that can only detect CMPs from ONETRUST. With a program like the one created in this thesis it could help them

[1] https://noyb.eu/en/more-cookie-banners-go-second-wave-complaints-underway

detect more CMPs that are not only from one CMP provider. The program could help the organization more as it also tells the user which dark pattern a CMP contain.

7.1 Future work

For the future it would be recommended to optimize the scraper for more consistent extraction of the CMPs. One way this could be done is using visual machine learning algorithms to learn to recognize a CMP. This may be a better method as it does not rely on how the website is coded but rather more how the CMP is visually represented. From the manual investigation there seem to only be a handful of different types of CMPs, those with a large square right in the middle of the screen, those with full banners at the bottom and top, and those with a small pop up in one of the corners. I suspect the main issue with this method would be to acquire a labelled data set of screenshots so the researcher would have to create one themselves. When the data set is created the researchers could then uses for example a Convolutional Neural network (CNN) to recognize the different patterns for the CMPs from screenshots of websites. If the training of such a CNN got a high accuracy on recognizing the CMPs, one could use it to either mark for a scraper where the CMP was to extract text from the CMP that way or use the screenshot and apply different algorithms like OCR to extract text and maybe other features.

The method could also optimize the dark pattern type recognition. One method could be to try to train a machine learning algorithm to recognize the buttons, accept, decline and settings. Then if this algorithm has a high accuracy on recognizing the buttons and what their text is, they will increase the accuracy for defining "no choice", "choice cascade" and "no antonyms" as there would be no cases where the program is tricked by the text of the CMP writing about rejection of the CMP.

I have made a diagram of how such a model could look in figure 7.1. On the figure I have included the pattern multiple choice panels as that could be implemented by having the CNN return CMPs on a website until it cant find one. The red squares on figure 7.1 around the buttons on the CMP is to highlight that the potential machine learning model could learn to extract these buttons.

For the future the program could also be more optimized by adapting to how the CMPs would look. This will however fit areas under GDPR or being treated as such better than those areas outside the GDPR region. The areas under GDPR have regulation on what features a CMP should have and where. Therefore, it is here the CMPs will look most like each other especially now that they can be strictly fined like Google

and facebook[2], and risk being reported by organizations like NOYB, who reports up to 500 websites at a time for unlawful CMPs.[3]

[2] https://www.cnil.fr/en/cookies-cnil-fines-google-total-150-million-euros-and-facebook-60-million-euros-non-compliance

[3] https://noyb.eu/en/more-cookie-banners-go-second-wave-complaints-underway

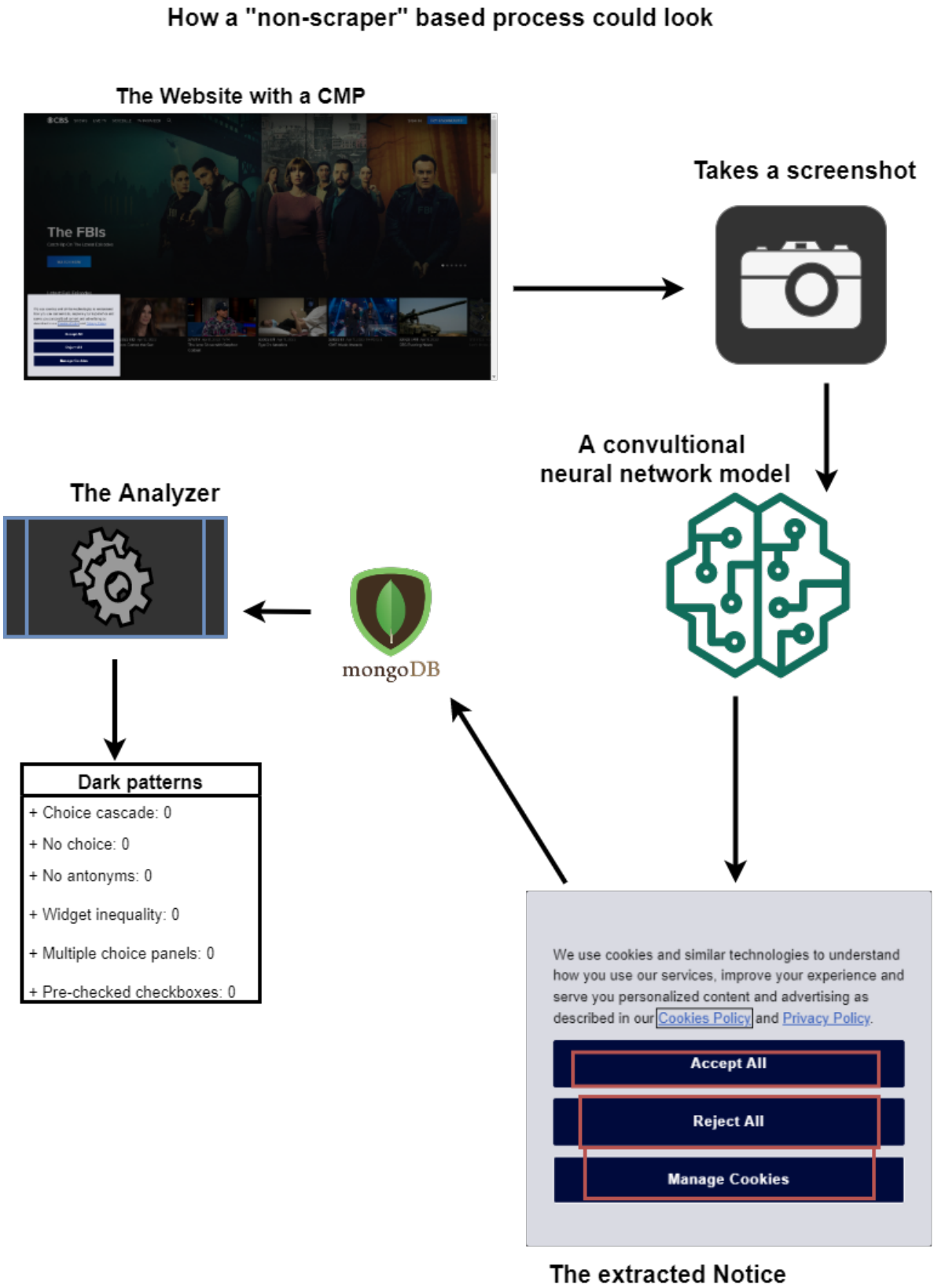

Figure 7.1: An example on how a Machine learning method could look like

Chapter 8

Appendix

The github with the datasets from the investigation and the code for the programs can be found here: https://github.com/MAP-12/Automatic_detection_of_Dark_patterns. The program that classifies each of the dark patterns is called Analyzer.ipynb. The investigation dataset of CMPs that was returned by the scraper is called cookienotices.csv and can be found in the folder called Data investigation. In this folder you will also find a file called "nonotices" which is the investigation of 101 random sample from the websites the scraper did not find any CMPs. For the scraper you can find it and all the related files in the folder called Scraper, along with a license of free to use from the bachelor thesis that first created the scraper.

Bibliography

[1] The california consumer privacy act of 2018. URL: https://oag.ca.gov/privacy/ccpa. 1

[2] General data protection regulation. URL: https://gdpr-info.eu/. 1

[3] Nosql databses list by hosting data. https://hostingdata.co.uk/nosql-database/. Accessed: 2022-05-24. 4.1

[4] CFPIM Amir M. Hormozi Ph.D. Cookies and privacy. *Information Systems Security*, 13(6):51–59, 2005. arXiv:https://doi.org/10.1201/1086/44954.13.6.20050101/86221.8, doi:10.1201/1086/44954.13.6.20050101/86221.8. 1

[5] Jan M. Bauer, Regitze Bergstrøm, and Rune Foss-Madsen. Are you sure, you want a cookie? the effects of choice architecture on users' decisions about sharing private online data. *Computers in Human Behavior*, 120:106729, 2021. URL: https://www.sciencedirect.com/science/article/pii/S0747563221000510, doi:https://doi.org/10.1016/j.chb.2021.106729. 1

[6] Dino ; Bollinger, Karel ; Kubicek, Carlos ; Cotrini, David Basin, Dino Bollinger, Eth Zurich, Karel Kubicek, and Carlos Cotrini. Automating Cookie Consent and GDPR Violation Detection. 2022. URL: https://doi.org/10.3929/ethz-b-000525815. 2.1.1, 2.3.1, 6

[7] Facundo Bre, Juan Gimenez, and Víctor Fachinotti. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, 11 2017. doi:10.1016/j.enbuild.2017.11.045. (document), 3.2

[8] Ignacio N. Cofone. The way the cookie crumbles: online tracking meets behavioural economics. *International Journal of Law and Information Technology*, 25(1):38–62, 10 2016. arXiv:https://academic.oup.com/ijlit/article-pdf/25/1/38/10297056/eaw013.pdf, doi:10.1093/ijlit/eaw013. 1

[9] Andrea Curley, Dympna O'Sullivan, Damian Gordon, Brendan Tierney, and Ioannis Stavrakakis. The design of a framework for the detection of web-based dark patterns. *ICDS 2021: The 15th International Conference on Digital Society*, 07 2021. URL: https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1002&context=ascnetcon. 6

[10] Lee Raymond Dice. Measures of the amount of ecologic association between species. *Ecology*, 26:297–302, 1945. 3.2.1

[11] Daniel Glez-Peña, Anália Lourenço, Hugo López-Fernández, Miguel Reboiro-Jato, and Florentino Fdez-Riverola. Web scraping technologies in an API world. *Briefings in Bioinformatics*, 15(5):788–797, 04 2013. arXiv:https://academic.oup.com/bib/article-pdf/15/5/788/17488715/bbt026.pdf, doi:10.1093/bib/bbt026. 3.1

[12] Colin M. Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L. Toombs. The dark (patterns) side of ux design. page 114, 2018. URL: https://doi.org/10.1145/3173574.3174108. 1, 2.1, 2.1, 2.2, 2.1, 2.1.1, 6, 6, 6

[13] Philip Hausner and Michael Gertz. Dark Patterns in the Interaction with Cookie Banners. *Position Paper at the Workshop "What Can CHI Do About Dark Patterns?" at the CHI Conference on Human Factors in Computing Systems, May 8-13, 2021, Yokohama, Japan*, 1(1):1–5, 2021. URL: http://arxiv.org/abs/2103.14956, arXiv:2103.14956. 1, 4.1, 4.1, 6

[14] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6:107–116, 04 1998. doi:10.1142/S0218488598000094. 3.2

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi:10.1162/neco.1997.9.8.1735. 3.2

[16] David Liedle. A brief history of optical character recognition. 11 2018. URL: https://blog.filestack.com/thoughts-and-knowledge/history-of-ocr/#:~:text=The20first20OCR20tools20in20modern20history20were,by20interpreting20Morse20Code20to20read20text20aloud. 3.2

[17] Teodor Liljedahl Hildebrand and Filip Nyquist. Cookies, gdpr and dark patterns: Effect on consumer privacy (dissertation), 2021. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:bth-21726. 1, 3.1, 4.1, 4.1, 6

[18] Arunesh Mathur, Gunes Acar, Michael J. Friedman, Elena Lucherini, Jonathan Mayer, Marshini Chetty, and Arvind Narayanan. Dark patterns at scale: Findings from a crawl of 11K shopping websites. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW), 2019. arXiv:1907.07032, doi:10.1145/3359183. 1, 6

[19] Celestin Matte, Nataliia Bielova, and Cristiana Santos. Do cookie banners respect my choice?: Measuring legal compliance of banners from IAB europe's transparency and consent framework. *Proceedings - IEEE Symposium on Security and Privacy*, 2020-May:791–809, 2020. URL: https://arxiv.org/pdf/1911.09964.pdf, arXiv:1911.09964, doi:10.1109/SP40000.2020.00076. 1, 2.1.1, 2.3.1, 6

[20] Norwegian-Consumer-Council. You can log out, but you can never leave. 01 2021. URL: https://fil.forbrukerradet.no/wp-content/uploads/2021/01/2021-01-14-you-can-log-out-but-you-can-never-leave-final.pdf. 2.1

[21] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–13, 2020. arXiv:2001.02479, doi:10.1145/3313831.3376321. 1, 4.1, 6

[22] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. User tracking in the post-cookie era: How websites bypass GDPR consent to track users. In *Proceedings of the Web Conference 2021*. ACM, apr 2021. doi:10.1145/3442381.3450056. 2.1.1, 2.3.1, 6

[23] J.S. Park and R. Sandhu. Secure cookies on the web. *IEEE Internet Computing*, 4(4):36–44, 2000. doi:10.1109/4236.865085. 1

[24] David Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Mach. Learn. Technol.*, 2, 01 2008. 3.2.1

[25] Haim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, 2014. URL: https://arxiv.org/abs/1402.1128, doi:10.48550/ARXIV.1402.1128. 10

[26] Than Htut Soe, Oda Elise Nordberg, Frode Guribye, and Marija Slavkovik. Circumvention by design - Dark patterns in cookie consent for online news outlets.

ACM International Conference Proceeding Series, 2020. arXiv:2006.13985, doi:10.1145/3419249.3420132. 1, 1, 2.1.1, 2.2.2, 2.3, 6

[27] Than Htut Soe, Cristiana Teixeira Santos, and Marija Slavkovik. Automated detection of dark patterns in cookie banners: how to do it poorly and why it is hard to do it any other way, 2022. URL: https://arxiv.org/abs/2204.11836, doi:10.48550/ARXIV.2204.11836. 1, 6

[28] T. Sørenson. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*. Biologiske skrifter. I kommission hos E. Munksgaard, 1948. URL: https://books.google.no/books?id=rpS8GAAACAAJ. 3.2.1

[29] Mohammad Tahaei and Kami Vaniea. Code-Level Dark Patterns : Exploring Ad Networks ' Misleading Code Samples. pages 1–5, 2021. URL: https://groups.inf.ed.ac.uk/tulips/papers/tahaei2021-darkpatterns.pdf. 6

[30] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. (un)informed consent: Studying gdpr consent notices in the field. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 973990, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3319535.3354212. 1, 6

[31] Luis von Ahn, Benjamin Maurer, Colin McMillen, David J. Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321:1465 – 1468, 2008. 5.4

[32] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi:10.25080/Majora-92bf1922-00a. 4.2.3

[33] Carrie Williams. Research methods. *Journal of Business & Economic Research March*, 5, 01 2007. doi:10.19030/jber.v5i3.2532. 13

[34] Jun Yan, Ning Liu, Gang Wang, Wen Zhang, Yun Jiang, and Zheng Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 261270, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1526709.1526745. 1

[35] W. J. Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950. doi:https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3. 3.2.1