Fake News Websites

Diogo Pinheiro

***Abstract*—Fake News have been around us for a long time. With the sudden rise of social media and influx of information from various sources most people can not differentiate truth from a lie. In this thesis, we are going to discuss how we can help with this problem in Portugal. We try to find key elements that distinguish real news websites from fake websites. The system is also composed by a web crawler that searches the web pages from the chosen websites.**

In this thesis, we created a system capable of distinguish trustworthy websites from websites that spread fake, non verified news. The system starts by receiving an Uniform Resource Locator (URL). Then an http request is made and the system tries to find the sub-page with the web news site information. That page is processed and information is extracted. The system also finds other features, like the country localization and the security protocol. After gathering the features, the evaluator calculates the website score by giving a weight to each, marking the website as fake or trustworthy.

We have a data set with fake and trustworthy websites to test our system. We created a script that would give several weights values to the features and then run it against a training set. From that we extracted the four best results. We also created a Decision Tree to display an algorithm to mark fake websites.

***Index Terms*—Fake News, Web Scraper, Information Retrieval, Fake Websites**

I. Introduction

IT has become extremely difficult to filter every news post that we come by on our day to day life.

While we scroll in Facebook or Twitter news feeds, we are overwhelm with information and a great part of them are fake. A study [1] conducted in 2016 found that Facebook referred to untrustworthy websites over 15% of the time, in contrast it would only refer to trusty websites 6% of the time. Those news serve to catch the user attention and make him click the story, giving the story more and more visibility.

Identifying Fake news is not an exact science where we can always be right, sometimes news we call fake are exaggerating facts and making the information seem drastic. A news story has a lot of nuances and its very difficult to know for sure that the article we are reading is 100% right. Even the most trustworthy journal can use dubious sources or sources that are wrong. The best defense against Fake News is our judgement and doing research when we read a news story. There are several features and identifiers that we can use to know if the source where we are reading the news is worth our attention, specially for our use case in Portugal. A trustworthy source of news has some characteristics that can be used to distinguish from malicious and deceitful news sources. In Portugal we have an identifier that every news outlet has to have in order to publish news. Untrustworthy news websites will not have this identifier.

This identifier is not the only feature that can be used to determine which news source is deem of our attention. We can also use the location of their website, security protocol and provider.

This thesis as as an objective to research and implement a system that can, through a series of features, classify an website as fake or trustworthy.

In section II we discuss previous related work that could address our problems. In section III we talk about the system to identify fake websites. In section IV we explain how we manage our evaluation process. In section V we make our conclusions. Finally, in Section VI we discuss about the Future work.

II. Related Work

Our system is not the first system to try to find fake websites or fake news, many systems try to solve this issue using several methods, such as Linguistic, Network and Structural as mentioned in [2]. We can differentiate two types of systems, classifier and lookup systems. Classifier systems evaluate and analyze the webpage and make an assessment with the information gathered. Lookup systems have a blocklist with several websites that are previously marked as malicious.

SpoofGuard [3] is an example of a tool that identifies malicious websites through a set of features present on that website. This system checks for the URL, the links on the page, domain among others. Spoofguard is a classifier system since it classifies the websites by making an evaluation of the webpage being tested.

AZProtect is a tool mentioned in [4] is an hybrid system, using classifying and a lookup techniques. AZProtect uses attributes to clasify a webpage such as, inlinks, outlinks, https, language and hosting.

In the paper [5] Abbasi, A., Zahedi, F. “Mariam”, Kaza, propose an algorithm to identify fake medical web information. Their system, Recursive Trust Labeling (RTL), uses underlying content and graph-based classifiers designed to exploit the unique characteristics of fake medical Web sites, coupled with a recursive labeling mechanism.

Although these two systems have great results, they do not serve our purpose completely. The first system tries to identify spoofed websites, websites that try to copy genuine websites like facebook or amazon. The second system uses great attributes to classify the a website that will be useful to our system but is also focused in identifying spoof websites. And the third system objective is focused in the medical scene. Their system tries to identify fake websites with untrustworthy information and websites that sell counterfeit drugs.

III. System

The Fake Websites Catcher, is composed by five modules as showed in the figure 1. The entry point, an interface where

we post an website to be evaluated; The Web scraper where we scrape the pages of the web site for information; The Feature Extractor where we retrieve the websites attributes; The evaluator where we classify the website with its probability of being fake; Finally the Database where we store our data.

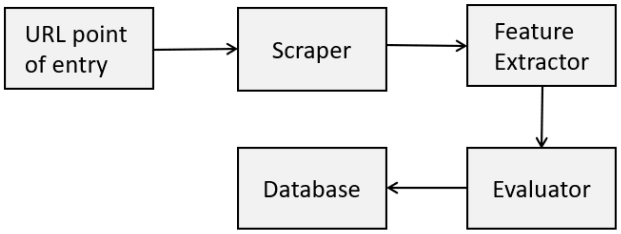

Fig. 1. Overall view of the system

A. Website Retrieval

We created an user interface that receives an url as an input. By accessing the base domain of the app we are prompted by a search text bar where we can insert an url. Clicking submit will start the web site evaluation and display the results on the next web page. We have a go back button that will return to previous page to evaluate another website or show all websites button, were we display all the websites that were evaluated.

In order to be able to evaluate websites we created a simple graphic interface using a web framework.

There was two choices of frameworks, Django and Flask.

Django is a framework with more features than Flask however is less explicit, an heavier framework and harder to start using. For that reason we decided to use Flask for this application, is a lighter and simpler framework with only the essential to build a web application. Although is a fairly simple framework it is extendable, in case we need to have to implement more features like authentication.

As is written in the article [6]:

> Based on the study, it is evident that Django can be best fit for large-scale projects with the cost of the learning curve. Flask is best fit for the prototyping and smallscale projects but not limited to it.

B. Web Scraper

A Web Crawler is responsible for navigating the web page and retrieving the information we need. We use a python package named *Beautiful Soup* to help us scraping the links and information necessary to evaluate the website.

Beautiful Soup [1] is a package commonly use to parse HTML or other markup languages. This is very useful because if a website does not give you a way to download the information you need, it is possible to obtain with *Beautiful Soup*. *Beautiful Soup* navigates through the markup text and remove those markups to return clean text. Not only can remove the markups but it can also search for a specific markup and return the text in that markup. In our project we use that feature to find any tag with <a >and return the hyperlink found. This feature is paramount in order to extract only the meaningful text from the web page. We also use a package named requests to be able to make a request to the web page we are going to evaluate.

We start the scrape of the web site by parsing the input. Because the url string needs to start with http in order to make the request we need to verify and append it to the string in case the url does not have it.

By using the package requests we create a *get request* that returns an object with the page information.

After that request we use *Beautiful Soup* to extract all the links in the page that have *Ficha* or *Ficha Técnica* and save all of the occurrences into a list. In case no page is found we try we a different approach. We use a XPATH parser with the following expression:

```
"//a[contains(@href, 'ficha')]/@href"
```

C. ERC and Editor

To be able to find the ERC and editor of the website news we take the urls found in the previous step and construct another http GET request. By using *Beautiful Soup* we try to find all the references of ERC in the page and retrieve the next set of words after that mention.

If it was not possible to find any ERC mentions, we will then proceed to verify the media group editor that the news paper belongs to.

To find the editor we use a similar approach. With the response from the request we use the *Beautiful Soup* to find all the references of "Editor" and take the next set of words and add them to a list. In the end we get a list of possible Editors for the news website.

After the ERC and editor phase, we advance for the next step. This step is where we confirm the existence and legitimacy of the numbers found. We can not take for granted that we got is a legitimate number or news group. We take the ERC numbers and start a search for them in two csv files, that we downloaded from the *Entidade Reguladora de Comunicação* website[2]. These files have ERC numbers, editor, location of the editor headquarters and headquarters postal code.

Finally if we do not find an ERC matching the one in the page we set the ERC value of the web site to 0 and we will try to check for the Editor.

For the media group Editor we have the same procedure. We take the words found in the page and search for them in the csv files. However, for the Editor, given it is a word we need to check for degrees of similarity, because there is a possibility that the name can be writing in several forms or with acronyms. To overcome this issue we use the Jaro-Winkler distance metric.

The Jaro-Winkler distance is a string measure between two sequences and a variation of the Jaro distance:

$$sim_j = \frac{1}{3}(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m})$$

[1]https://www.crummy.com/software/BeautifulSoup/

[2]https://www.erc.pt/

Where:

- $|s_i|$ is the length of the string s_i
- m is the number of matching characters
- t is the number of transpositions, that is the number of matching characters but in different order divided by 2.

The closer the value is to 1 the similar are the two words. The Jaro-Winkler similarity is:

$$sim_w = sim_j + lp(1 - sim_j)$$

Where:

- sim_j is the Jaro similarity
- l is the length of the common prefix
- p is a constant scaling factor that determines how much the l factor is valued. The standard value for p is 0.1

We set a Threshold of 0.9 and find any words in the files with a very high degree of similarity to the word found in the page. If we find more than one similar word, we check their Jaro value and return the word with the highest value. If we do not find any match we set the Editor value of the website to an empty string.

D. Database

The output of the website evaluation had to be saved in some place. To do that we created a table to save the results. For the database we decided to use SQL alchemy because is simpler and it has a plenty of support and a large community using it.

Our system uses one table with eight columns. The primary key, id, which is a database generated identifier for the entry, the erc which is an integer with default value 0, news group, domain, ip address, provider, country and color.

E. Websites Features

When we evaluate a website we look for some key features. These features will help us decide if the website can be trusted or not. The features are:

- ERC - This field tell us the charter number of the newspaper or the number of the group that they belong. Usually all the newspaper have this information in their information page. However we found a few big newspaper that did not have this information in their page.
- News Group - The news group to which the newspaper belongs. This information is important because it can tell us if the news group is registered and they are legitimate. But just has it happens, in the ERC, some website newspaper do not have the news group information in their page. And therefore we can not obtain this information all of the time.
- Top level Domain - The website top level domain specifies the entity the website is registered in. Usually, Portuguese websites have .pt has their top level domain.
- Communication Protocol - We check if the the website has the https protocol. As we referred in the previous chapter a website is likelier to be have dubious information if it does not use the secure communication protocol.
- Provider - If the provider is not a well know provider in Portugal, it can also be an indicative that the website is not credible.
- Country - the country where the website is located can also be an indicator that the website is not legitimate. Being located in a country that is not Portugal enables the website to not follow the Portuguese law.

These are the features that we weigh when considering the legitimacy of the website.

F. Evaluator

After we retrieve all the information we can from the website, we pass the features retrieved to an evaluator. We defined a weight value for each feature in an environment variables file.

Were we have the algorithm that scores a website to be fake:

We start by setting the score of being an illegitimate website to 0 and depending on the tests that fail we increase that value the set amount for that test. In these tests we compare the website location, where we check if it is in Portugal. We check if we found an ERC value. If it was found an editor for that news website. The website domain, where we see if the top level domain is .pt or .com. The website protocol to check if it uses the secure protocol and finally we check the provider that we verify if it is a Portuguese provider. If any of these tests fails we increase the value of the score of the website being fake.

After the tests we return a color matching the result we got from the evaluator. Green for a legitimate website, yellow the website has some characteristics of a fake website, orange we need to be careful with the website, red the website has all the key characteristics of a fake website.

IV. Evaluation

To evaluate our system we gather a list of fake news websites. This list was given to us by Prof. Bruno Martins, the list had forty six websites however by the time we started our evaluation seventeen of those websites were already deleted, putting the list at twenty nine websites. We also needed a list of trustworthy websites, to obtain that data we chose renown websites and check them manually against the list provider by the *Entidade Reguladora de Comunicação*. In the end we obtained twenty two trustworthy websites.

To perform the evaluation we started by choosing at random a training set containing 10 websites, five fake and five trustworthy. Those ten websites were given to the system and their features extracted. With the websites and their features in the database we gave the data to the evaluator. The evaluator change the six parameters weight between a threshold defined apriori (5-40) that would increase in increments of 5 and test the training set against all those combinations. Then we extract the four best results and then test all our data set with the weight of the parameters obtained. We also decided to use a Genetic Algorithm to help us find the best possible solution.

We will compare the different set of weights by their F-Score.

We can determine the F-Score of the results with the following formula:

$$F_\beta = (1+\beta^2)\frac{precision * recall}{(\beta^2 * precision) + recall} =$$

$$\frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + \beta^2 * FP + FN}$$

The β is a positive value, where β is chosen such that recall is considered β times more important than precision. For our case we will consider β equals to 1. We consider trustworthy websites, the ones that score below or equal to 0.5.

We are using the following attributes, Country, ERC Number, NewsGroup, Domain, Http Security Protocol and Provider.

We did four evaluations with the following weights:The four best results obtained are:

- 30, 05, 10, 20, 05, 30
- 25, 05, 10, 20, 05, 35
- 30, 10, 05, 15, 25, 15
- 35, 10, 05, 20, 05, 25

The results for the four evaluations are shown in the following graphs:

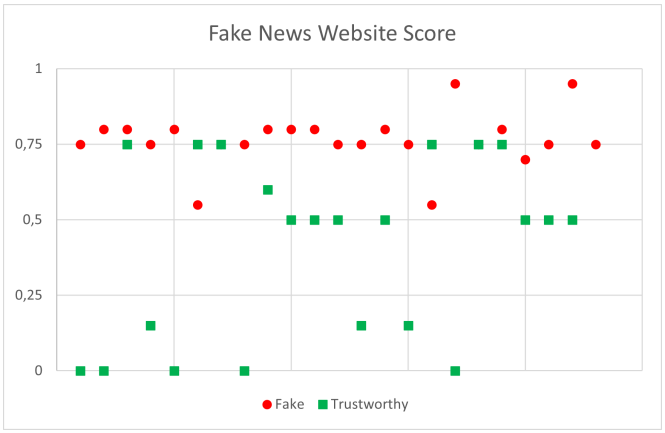

Fig. 2. Graph for the first test

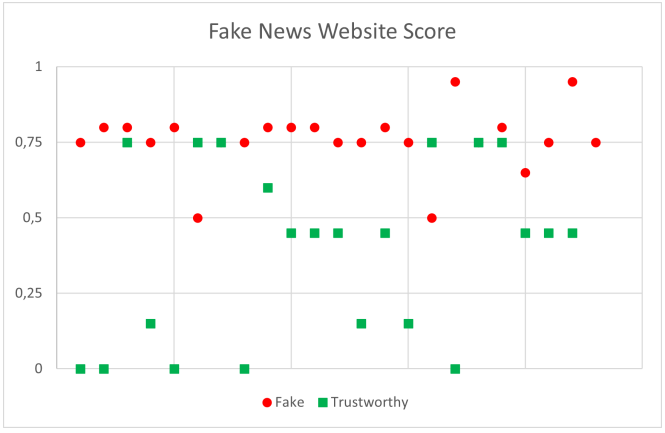

Fig. 3. Graph for the second test

For the first evaluation we have and F-Score of, $F_\beta = \frac{(1+\beta^2)*TP}{(1+\beta^2)*TP+\beta^2*FP+FN} = \frac{2*15}{2*15+2+7} = 0.77$

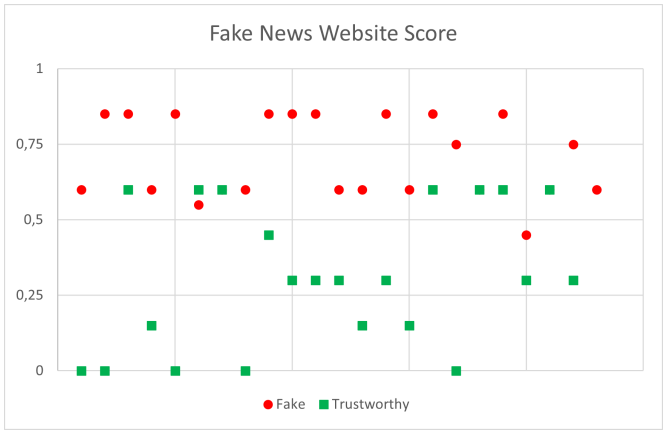

Fig. 4. Graph for the third test

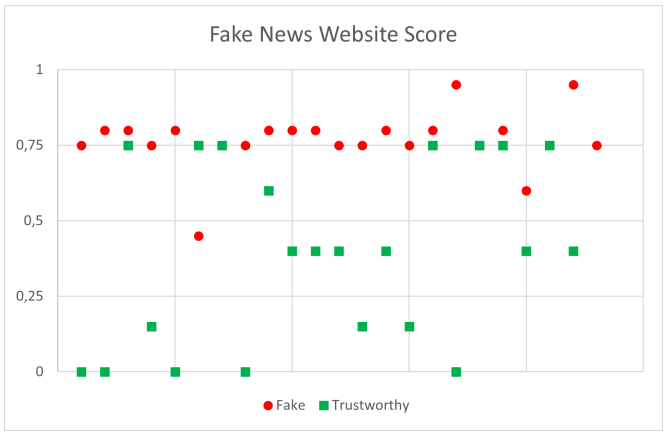

Fig. 5. Graph for the fourth test

In the second evaluation the F-Score was 0.82. In the third we got an F-Score of 0.79 and finally in the fourth evaluation we got a score of 0.76.

The best result that we got was in the second evaluation.

As shown in the 3, the evaluator gave a score greater or equal of 0.75 to all the fake websites except three. It gave 0 to Five websites, meaning that these websites are very likely to be trustworthy because it did not had any of the characteristics of a fake website. We have three trustworthy websites are lower than 0.25, meaning some characteristics of a fake website were found in these three websites. Finally, fourteen trustworthy websites had a score between 0.4 and 0.75. This can be explain by the way the web scrapper that finds the website features is built. Specifically the ERC Number and News Group feature. Every news websites have a different way to present and write the technical information of their business and it becomes really difficult for the parser to find the context of the ERC and News Group in every website. Although in this evaluation the Provider and the Country have a lot of weight if one of those features fail will affect greatly the result.

Since the higher values were the Country and the Provider when the parser does not find those features, the evaluator will give them a higher score and consider them as fake.

A. Genetic Algorithm

We decided to use a genetic algorithm to find the best set of values to the attributes and see how it compares to the scores in the previous section. To create the genetic algorithm we use a a python library called PyGAD [3].

We started by creating a fitness function that sees how close the solution provided by the genetic algorithm would be to the real results.

The fitness functions starts by checking if the sum of the solution is between 90 and 100. We created this condition because the sum of our solution needs to be 100, if we did not had this condition the algorithm would give higher numbers and the score of the websites would no be in the 0-100 range. We also gave the lower threshold the value 90 because we wanted to have some leeway for the algorithm to find some solutions. It would be almost impossible for the solution to be exactly 100. We run the algorithm some times and the sum of the solution would always converged to 100. Therefore, if the solution was not in the range of 90-100 we would give a very low fitness.

The next step would be to evaluate all the websites with the solution provided by the algorithm and saved them in an array with size of the numbers of websites we have to evaluate.

After we evaluate all the websites we find the F-Score for the iteration. We create two loops with two conditions each in order to fin the TP, TN, FP and FN values.

Finally we return the F-score and the highest value that we obtain will be the best generation that our algorithm could create.

We also had to give the genetic algorithm other settings, like number of generations, probability of mutations, low range and high range, mutation type and numbers of genes.

- Number of generations - How many iterations will the algorithm do : 50
- Probability of mutations - Probability of the the population change their parameters : 0.1
- Number of genes - How many attributes does it have : 6
- Low range - Lowest value of an attributes : 1
- High range - Highest value of an attributes : 50
- Mutation type - What attributes will be mutated : random
- Crossover point - When crossing to elements what type of crossover will be : Single point

When running the algorithm we got the following result:

```
Parameters of the best solution :
[33.81577413
3.84281601
10.01569059
31.38444731
6.01378099
6.00516052]

Fitness value of the best solution =
0.7894736842105263
```

With a F-Score value of 0.79, this solution results in the following graphic 6.

[3]https://pygad.readthedocs.io/en/latest/

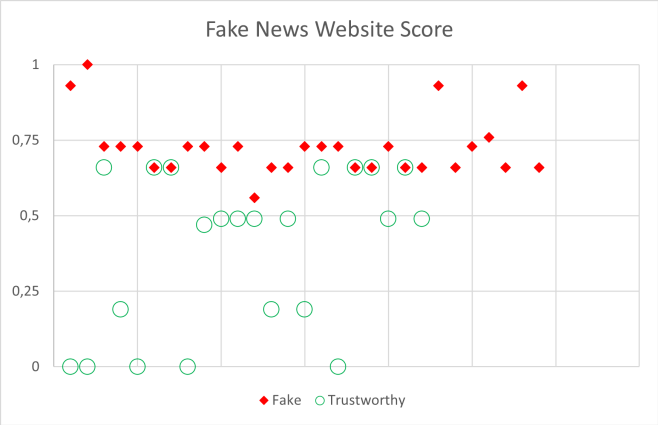

Fig. 6. Genetic Algorithm results

The result we got is equal to the second best result that we got in the previous evaluations, which is a good result. We could possible have better results by tweaking some settings of the genetic algorithm, like the numbers of generations.

Nevertheless the Genetic Algorithm provided a good solution. Classifying only two of the fakes websites in the wrong place, and miss classifying six trustworthy websites.

B. Decision tree

We also generated a decision tree. A decision tree is a flowchart where each node is a test on a feature and each line represents a binary outcome, true or false.

We decided to create the decision tree with the help of a package in python called Sci-kit learn to fit the data to the tree and we use a package named Pandas to arrange the data in a format the sci-kit learn can use it.

In order to use the data we had we converted the results in a binary form. If the Country was not Portugal it would get a 1 and if it was Portugal it would get a 0, that way we could create the decision tree.

We also chose to use a classifier tree because it is better suited for our problem given we have a binary classification, fake or not fake.

The tree we got is displayed in the Figure 7.

We start on the top of the tree. The first node has analyses the Country and divides the sample between less or equal than 0.5 and more than 0.5. If they are less than 0.5 the samples go to the branch on the left, if they are greater than 0.5 they go to the right. We also have the number of samples, 51, and the value of the samples, 29 fake websites and 22 trustworthy websites. In the first node we can see that 2 out of the 29 fake websites are not hosted in Portugal, and 7 out of the 22 trustworthy websites are not hosted in Portugal.

On the left branch we got the domain feature, where it divides between .com and .pt from the others domain, here we can see that only one fake websites does not have .com or .pt domain.

After that we find a node testing the http protocol where fifteen trustworthy websites have secure http protocol and one fake websites has not.

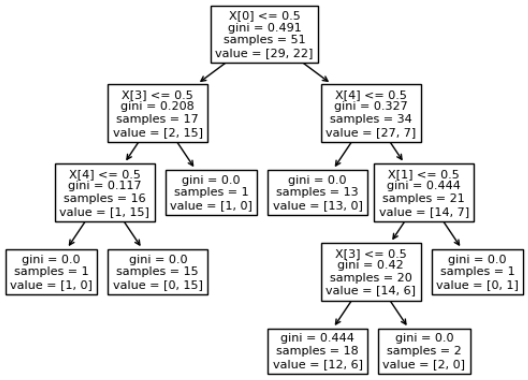

Fig. 7. Decision Tree

On the final classifier node we have the Domain where eighteen of the samples have the .com or .pt domain and two of them have other domains. This last node is the only leaf node that we have fake and trustworthy websites. That means that many fake websites shared the same feature as the trustworthy ones. We would need another feature to classify and separate these websites.

The tree also shows the Gini index, this index indicates the probability of a feature that is classified incorrectly when selected at random, the lower the Gini index the most accurate is the classification..

The Gini index can be calculated with the following formula:

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Where *C* is the amount of classes, in our case we have two classes and *p(i)* is the probability of selecting a class.

The News Group and the Provider parameter was not used by the classifier to create the tree.

With this tree we can take a website, extract their features and follow the tree path to obtain a result.

We can also observe that a website hosted in Portugal with a domain .pt or .com and using a secure http protocol can be considered trustworthy. It is also possible to identify that a website not hosted in Portugal and without Https protocol is considered Fake.

The other leaf nodes represent only one website and there is one that has 18 websites where those websites are hosted outside Portugal, use Https protocol, no ERC was found and their domain is .pt or .com we have higher Gini. That means that if we find a website with those characteristics, there is a higher possibility that we classified poorly the website.

C. Discussion

With this results we can infer some important points. The country is a good indicator of the trustworthiness of a website. The News Group and ERC Number are great metrics however they are complicated to extract automatically from the news websites pages. The second set of weights had the best results. The Genetic Algorithm solution had good results, equal to the second best evaluation. The decision tree as expected gave the bigger factor to ascertain if the website is fake or not, the location of the website. The decision tree did not use the news group feature and the ERC, because they are very difficult to obtain with a high degree of certainty. The decision tree also did not use the Provider feature which had the best results in our evaluation. This probably happened because in our training set, our samples had a good score with the provider feature. However testing with all the samples, the best weights are not the same as the training set.

V. Conclusion

We tried to create a system capable of distinguish between fake news websites and trustworthy websites. We created an web interface with an url as input, a website parser to read an web page and find useful information and a sub system that collects various information from the website in order to be able to successful be evaluated. We learned what are the key features that can show what website can be trusted being the most accurate for our case the location where the website is hosted.

One of the most difficult steps to implement was the retrieval of context from a web page, more specifically the retrieval of the ERC Number and News Group of the website. The parser is unsuccessful at identifying and retrieving those features for some news website. Every website is made differently and the way they expose their information is different from each other. Therefore there is no single rule that the system can use to retrieve that information with 100% accuracy. This problem impacts the evaluation, because some websites actually have the ERC Number and have a real News Group but the system can not find that information. Thus the trustworthy websites have an higher score than they were suppose to have.

VI. Future Work

The system showed some good results, and could detect with high accuracy the fake news websites, but some aspects can be improved. Although we previously stated that the retrieval of context was difficult. We believe that the parser can be improved to be more effective in retrieving their ERC Number and News Group. One way of improving this is adding more rules when searching for the ERC and Editor.

This system works in Portugal, because it uses an identifier given by a Portuguese authority. The system can be extended to work in other countries, if that country has an entity similar to what Portugal has. Having the files with those numbers, changing the value of the feature Country to the country that is being tested, and changing some key words that are exclusively to Portugal, the system would provided similar results for the country being tested.

REFERENCES

[1] J. R. Andrew M. Guess, Brendan Nyhan, "Exposure to untrustworthy websites in the 2016 us election," 2020.

[2] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[3] N. C. R. L. Y. Teraguchi and J. C. Mitchell, "Client-side defense against web-based identity theft," *Computer Science Department, Stanford University. Available: http://crypto. stanford. edu/SpoofGuard/webspoof. pdf*, 2004.

[4] A. Abbasi, F. Zahedi, S. Kaza, *et al.*, "Detecting fake medical web sites using recursive trust labeling," *ACM Transactions on Information Systems (TOIS)*, vol. 30, no. 4, p. 22, 2012.

[5] A. Abbasi and H. Chen, "A comparison of tools for detecting fake websites," *Computer*, vol. 42, no. 10, pp. 78–86, 2009.

[6] Ghimire, Devndra, "Comparative study on Python web frameworks: Flask and Django https://www.theseus.fi/handle/10024/339796," 2020.