Classifying Illegal Activities on Tor Network Based on Web Textual Contents

Mhd Wesam Al Nabki[1,2], Eduardo Fidalgo[1,2], Enrique Alegre[1,2], and Ivan de Paz[1,2]

[1]Department of Electrical, Systems and Automation, University of León, Spain
[2] Researcher at INCIBE (Spanish National Cybersecurity Institute), León, Spain
{mnab, eduardo.fidalgo, ealeg, ivan.paz.centeno}@unileon.es

Abstract

The freedom of the Deep Web offers a safe place where people can express themselves anonymously but they also can conduct illegal activities. In this paper, we present and make publicly available[1] a new dataset for Darknet active domains, which we call it "**D**arknet **U**sage **T**ext **A**ddresses" (DUTA). We built DUTA by sampling the Tor network during two months and manually labeled each address into 26 classes. Using DUTA, we conducted a comparison between two well-known text representation techniques crossed by three different supervised classifiers to categorize the Tor hidden services. We also fixed the pipeline elements and identified the aspects that have a critical influence on the classification results. We found that the combination of TF-IDF words representation with Logistic Regression classifier achieves 96.6% of 10 folds cross-validation accuracy and a macro F1 score of 93.7% when classifying a subset of illegal activities from DUTA. The good performance of the classifier might support potential tools to help the authorities in the detection of these activities.

1 Introduction

If we think about the web as an ocean of data, the Surface Web is no more than the slight waves that float on the top. While in the depth, there is a lot of sunken information that is not reached by the traditional search engines. The web can be divided into Surface Web and Deep Web. The Surface Web is the portion of the web that can be crawled and indexed by the standard search engines, such as Google or Bing. However, despite their existence, there is still an enormous part of the web remained without indexing due to its vast size and the lack of hyperlinks, i.e. not referenced by the other web pages. This part, that can not be found using a search engine, is known as Deep Web (Noor et al., 2011; Boswell, 2016). Additionally, the content might be locked and requires human interaction to access e.g. to solve a CAPTCHA or to enter a log-in credential to access. This type of web pages is referred to as "database-driven" websites. Moreover, the traditional search engines do not examine the underneath layers of the web, and consequently, do not reach the Deep Web. The Darknet, which is also known as Dark Web, is a subset of the Deep Web. It is not only not indexed and isolated, but also requires a specific software or a dedicated proxy server to access it. The Darknet works over a virtual sub-network of the World Wide Web (WWW) that provides an additional layer of anonymity for the network users. The most popular ones are "The Onion Router"[2] also known as Tor network, "Invisible Internet Project" I2P[3], and Freenet[4]. The community of Tor refers to Darknet websites as "Hidden Services" (HS) which can be accessed via a special browser called Tor Browser[5].

A study by Bergman et al. (2001) has stated astonishing statistics about the Deep Web. For example, only on Deep Web there are more than 550 billion individual documents comparing to only 1 billion on Surface Web. Furthermore, in the study of Rudesill et al. (2015) they emphasized on the immensity of the Deep Web which was estimated to be 400 to 500 times wider than the Surface Web.

The concepts of Darknet and Deep Net have ex-

[1]The dataset is available upon request to the first author (email).

[2]www.torproject.org
[3]www.geti2p.net
[4]www.freenetproject.org
[5]www.torproject.org/projects/ torbrowser.html.en

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 35–43,
Valencia, Spain, April 3-7, 2017.

isted since the establishment of World Wide Web (WWW), but what make it very popular in the recent years is when the FBI had arrested Dread Pirate Roberts, the owner of Silk Road black market, in October 2013. The FBI has estimated the sales on Silk Road to be 1.2 Billion dollars by July 2013. The trading network covered among 150,000 anonymous customers and approximately 4,000 vendors (Rudesill et al., 2015). The cryptocurrency (Nakamoto, 2008) is a hot topic in the field of Darknet since it anonymizes the financial transactions and hides the trading parties identities (Ron and Shamir, 2014).

The Darknet is often associated with illegal activities. In a study carried out by Intelliagg group (2015) over 1K samples of hidden services, they claimed that 68% of Darknet contents would be illegal. Moore et at. (2016) showed, after analyzing 5K onion domains, that the most common usages for Tor HS are criminal and illegal activities, such as drugs, weapons and all kind of pornography.

It is worth to mention about dramatic increase in the proliferation of Darknet domains which doubled their size from 30K to 60K between August 2015 and 2016 (Figure 1). However, the publicly reachable domains are no more than 6K to 7K due to the ambiguity nature of the Darknet (Ciancaglini et al., 2016).

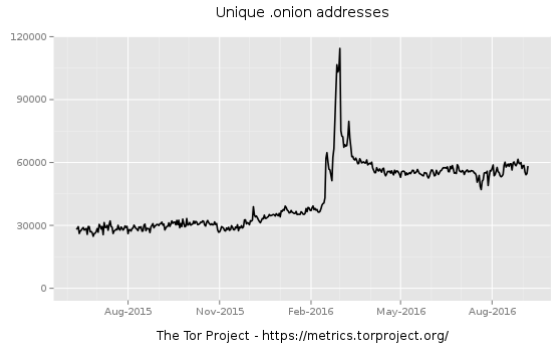

Figure 1: The number of unique *.onion addresses in Tor network between August 2015 to August 2016

Motivated by the critical buried contents on the Darknet and its high abuse, we focused our research in designing and building a system that classifies the illegitimate practices on Darknet. In this paper, we present the first publicly available dataset called "**D**arknet **U**sage **T**ext **A**ddresses" (DUTA) that is extracted from the Tor HS Darknet. DUTA contains 26 categories that cover all the legal and the illegal activities monitored on Darknet during our sampling period. Our objective is to create a precise categorization of the Darknet via classifying the textual content of the HS. In order to achieve our target, we designed and compared different combinations of some of the most well-known text classification techniques by identifying the key stages that have a high influence on the method performance. We set a baseline methodology by fixing the elements of text classification pipeline which allows the scientific community to compare their future research with this baseline under the defined pipeline. The fixed methodology we propose might represent a significant contribution into a tool for the authorities who monitor the Darknet abuse.

The rest of the paper is organized as follows: Section 2 presents the related work. Next, Section 3 explains the proposed dataset DUTA and its characteristics. After that, Section 4 describes the set of the designed classification pipelines. Then, in Section 5 we discuss the experiments performed and the results. In Section 6 we describe the technical implementation details and how we employed the successful classifier in an application. Finally, in Section 7 we present our conclusions with a pointing to our future work.

2 Related Work

In the recent years, many researchers have investigated the classification of the Surface Web (Dumais and Chen, 2000; Sun et al., 2002; Kan, 2004; Kan and Thi, 2005; Kaur, 2014), and the Deep Web (Su et al., 2006; Xu et al., 2007; Barbosa et al., 2007; Lin et al., 2008; Zhao et al., 2008; Xian et al., 2009; Khelghati, 2016). However, the Darknet classification literature is still in its early stages and specifically the classification of the illegal activities (Graczyk and Kinningham, 2015; Moore and Rid, 2016).

Kaur (2014) introduced an interesting survey covering several algorithms to classify web content, paying attention to its importance in the field of data mining. Furthermore, the survey included the pre-processing techniques that might help in features selection, like eliminating the HTML tags, punctuation marks and stemming. Kan et al. explored the use of Uniform Resource Locators (URL) in web classification by extracting the features through parsing and segmenting it (Kan,

2004; Kan and Thi, 2005). These techniques can not be applied to Tor HS since the onion addresses are constructed with 16 random characters. However, tools like Scallion[6] and Shallot[7] allow Tor users to create customized .onion addresses based on the brute-force technique e.g. Shallot needs 2.5 years to build only 9 customized characters out of 16. Sun et at. (2002) employed Support Vector Machine (SVM) to classify the web content by taking the advantage of the context features e.g. HTML tags and hyperlinks in addition to the textual features to build the feature set.

Regarding the Deep Web classification, Noor et al. (2011) discussed the common techniques that are used for the content extraction from the Deep Web data sources called "Query Probing", which is commonly used for supervised learning algorithms, and "Visible Form Features" (Xian et al., 2009). Su et al. (2006) have proposed a combination between SVM with query probing to classify the structured Deep Web hierarchically. Barbosa et al. (2007) proposed an unsupervised machine learning clustering pipeline, in which Term Frequency Inverse Document Frequency (TF-IDF) was used for the text representation, and the cosine similarity for distance measurement for the k-means.

With respect to the Darknet, Moore et. al. in (2016) have presented a new study based on Tor hidden services to analyze and classify the Darknet. Initially, they collected 5K samples of Tor onion pages and classified them into 12 classes using SVM classifier. Graczyk et al. (2015) proposed a pipeline to classify the products of a famous black market on Darknet, called Agora, into 12 classes with 79% of accuracy. Their pipeline architecture uses the TF-IDF for text features extraction, the PCA for features selection, and SVM for features classification.

Several attempts in literature have been proposed to detect illegal activities whether on the World Wide Web (WWW) network (Biryukov et al., 2014; Graczyk and Kinningham, 2015; Moore and Rid, 2016), peer-to-peer networks (P2P) (Latapy et al., 2013; Peersman et al., 2014) and in chatting messaging systems (Morris and Hirst, 2012). Latapy el at. (2013) investigated P2P systems, e.g. eDonkey, to quantify the paedophile activity by building a tool to detect child-pornography queries by performing a series of lexical text processing. They found that 0.25% of entered queries are related to pedophilia context, which means that 0.2% of eDonkey network users are entering such queries. However, this method is based on a predefined list of keywords which can not detect new or previously unknown words.

3 The Dataset

3.1 Dataset Building Procedure

To best of our knowledge, there is no labeled dataset that encompasses the activities on the Darknet web pages. Therefore, we have created the first publicly available Darknet dataset and we called it *Darknet Usage Text Addresses (DUTA)* dataset. Currently, DUTA contains only Tor hidden services (HS). We built a customized crawler that utilizes Tor socket to fetch onion web pages through port 80 only i.e. the HTTP protocol. The crawler has 70 worker threads in parallel to download the HTML code behind the HS. Each thread dives into the second level in depth for each HS in order to gather as much text as possible rather than just the index page as in others work (Biryukov et al., 2014). It searches for the HS links on several famous Darknet resources like onion.city [8] and ahmia.fi [9]. We reached more than 250K HS addresses, but only 7K were alive, and the others were down or not responding. After that, we concatenated the HTML pages of every HS into a single HTML file resulting a single HTML file for each single HS domain. We collected 7,931 hidden services by running the crawler for two months between May and July 2016. For the time being, we labeled 6,831 samples.

3.2 Dataset Characteristics

Darknet researchers have analyzed the HS contents and categorized them into a different number of categories. Biryukov et al. (2014) sampled 1,813 HS and detected 18 categories. Intelliagg group in (2015) analyzed 1K HS samples and classified them into 12 categories. Moore et al. (2016) studied 5,615 HS examples and categorized them into 12 classes. Based on our objective to build a multipurpose dataset and for the sake of completeness, we classified DUTA manually into 26 classes. To the best of our knowledge, this classification is the most extent and complete up to

[6] www.github.com/lachesis/scallion
[7] www.github.com/katmagic/Shallot
[8] www.onion.city
[9] www.ahmia.fi

date. The collected samples were divided among the four authors and each one labeled their designated part; if an author hesitated, it was openly discussed with the rest of the authors. Finally, to check the consistency of the manual labeling, the first author reviewed the final labeling by analyzing random samples of the categorization made by the others.

In addition to labeling the main classes, we dived into labeling the sub-classes of the HS. For example, the class *Counterfeit Personal Identification* has three sub-classes: *Identity Card*, *Driving License*, and *Passport*. Table 1 enumerates DUTA classes.

Main Class	Sub-Class	Count	Main Class	Count
Violence	Hate	4	**Art/ Music**	8
	Hitman	11	**Casino/ Gambling**	26
	Weapons	47	**Services**	285
Counterfeit Personal Identification	Driving-Licence	4	**Cryptocurrency**	586
	ID	7	**Down**	608
	Passport	37	**Empty**	1649
Hosting and Software	File-Sharing	111	**Forum**	104
	Folders	63	**Hacking**	90
	Search-Engine	38	**Wiki**	29
	Server	95	**Leaked-Data**	12
	Software	121	**Locked**	435
	Directory	142	**Personal**	405
Drugs	Illegal	230	**Politics**	8
	Legal	9	**Religion**	6
Marketplace	Black	63	**Library/Books**	27
	White	67	**Fraud**	4
Pornography	Child-pornography	914([10])	**Counterfeit Money**	55
	General-pornography	83	**Counterfeit Credit Cards**	240
Social-Network	Blog	71	**Human-Trafficking**	2
	Chat	47		
	Email	56		
	News	32	**The total count**	**6831**

Table 1: DUTA dataset classes

Counterfeit is a wide class so we split it into three main classes 1) *Counterfeit Personal Identification* which is related to government documents forging. 2) *Counterfeit Money* includes currencies forging and 3) *Counterfeit Credit Cards* covers cloning credit cards, hacked PayPal accounts and fake markets cards like Amazon and eBay. The class *Services* contains the legal services that are provided by individuals or organizations. The class *Down* contains the errors that were returned by the down web pages while crawling them e.g. an SQL error in a website database or a javascript error.

We assign class *Empty* to a web page when: 1) The text is very short i.e. less than 5 words, 2) It has only images with no text, 3) It contains unreadable text like special characters, numbers, or unreadable words, 4) The empty Cryptolockers pages (ransomware) (Ciancaglini et al., 2016). The class *Locked* contains the HS that require solving a CAPTCHA or a log-in credential. We noticed that some people love to present their works, projects, or even their personal information through an HS page so we labeled them into class *Personal*. The pages that fell under more than one category were labeled based on its main content. For example, we assign *Forum* label to the multi-topic forums unless the whole forum is related to a single topic. e.g. a hacking forum was assigned to *Hacking* class instead of *Forum*. The class *Marketplace* was divided into *Black* when it contained a group of illegal services like Drugs, Weapons, and Counterfeit services and *White* when the marketplace offered legal shops like mobile phones or clothes.

As we have labeled DUTA manually, we realized that some forums on HS contain numerous web pages and all of them are related to a single class i.e. we found a forum about child-pornography that has more than 800 pages of textual content, so we split it up into single samples representing one single forum page, and we added them to the dataset.

4 Methodology

Each classification pipeline is comprised of three main stages. First, text pre-processing, then, features extraction, and finally, classification. We used two famous text representation techniques across three different supervised classifiers resulting six different classification pipelines, and we examined every pipeline to figure out the best combination with the best parameters that can achieve high performance.

4.1 Text Pre-processing

Initially, we eliminated all the HTML tags, and when we detected an image tag, we preserved the image name and removed the extension. Furthermore, we filtered the training set for the non-English samples using Langdetect[11] python library and stemmed the text using Porter library from NLTK package[12]. Additionally, we re-

[10]This class includes 57 unique sample plus 857 samples that are extracted from a single forum (See Section 3.2)

[11]https://pypi.python.org/pypi/langdetect

[12]https://tartarus.org/martin/PorterStemmer/

moved special characters and stop words thanks to SMART stop list[13] (Salton, 1971). At this stage, we modified the stop words list by adding 100 words more in order to make it compatible with the work domain. Moreover, we mapped all emails, URLs, and currencies into a single common token for each.

4.2 Features Extraction

After pre-processing the text, we used two famous text representation techniques. A) Bag-of-Words (BOW) is a well-known model for text representation that extracts the features from the text corpus by counting the words frequency. Consequently, every document is represented as a sparse feature vector where every feature corresponds to a single word in the training corpus. B) Term Frequency Inverse Document Frequency model (TD-IDF) (Aizawa, 2003) is a statistical model that assign weights for the vocabularies where it emphasizes the words that occur frequently in a given document, while at the same time de-emphasizes words that occur frequently in many documents. However, even though the BOW and TF-IDF do not take into considerations the words order, they are simple, computationally efficient and compatible with medium dataset sizes.

4.3 Classifier Selection

For each features representation method, we examined three different supervised machine learning algorithms which are Support Vector Machine (SVM) (Suykens and Vandewalle, 1999), Logistic Regression (LR) (Hosmer Jr and Lemeshow, 2004), and Naive Bayes (NB) (McCallum et al., 1998).

5 Empirical Evaluation

5.1 Experimental Setting

Due to the purpose of this paper to classify the Darknet illegal activities, we selected a subset of our DUTA dataset by creating eight categories trying to cover the most representative illegal activities on the Darknet. Another condition that we imposed was that each class in the selected subset should be monotopic (i.e. related to a single category) and contain a sufficient amount of samples (i.e. 40 samples minimum). The rest of the classes are assigned to a 9th category which we called *Others*. Since we are working on classifying the illegal activities, we did not consider the class *Black-Market* in the training set because its contents are related to more than one class at a single time, and we wanted the classifier to learn from pure patterns. Moreover, when a sample contains relevant images but an irrelevant text or without any textual information, we excluded it from the dataset. Therefore, we had 5,635 samples distributed over nine classes i.e. the eight classes plus the *Others* one (Table 2). After the text pre-processing, we got 5,002 sample split it into a training set that contains 3,501 samples and a testing set of 1,501 samples.

Experiment Main Class	Count
Pornography	963
Cryptocurrency	578
Counterfeit Credit Cards	209
Drugs	169
Violence	60
Hacking	57
Counterfeit Money	46
Counterfeit Personal Identification (Driving-License, ID, Passport)	40
Others	3513

Table 2: Illegal activities dataset classes (A portion of DUTA dataset)

The dataset is highly unbalanced since the largest class has 3,513 samples while the smallest one has only 40 samples. We solved the skew in the dataset thanks to the *class-weight* parameter in Scikit-Learn library[14] which assigns a weight for each class proportional to the number of samples it has (Hauck, 2014). In addition to adjusting the weights of classes, we split up forums by the discussion page (See Section 3.2).

For the models tuning, we applied a grid search over different combinations of parameters with a cross-validation of 10 folds. The successful combination, which corresponds to the selected classification pipeline, is the one that can achieve the highest value of an averaged F1 score metric and an accuracy of 10 folds cross-validation.

We used Python3 with Scikit-Learn machine learning library for the pipelines implementation. We modified the parameters that have a critical influence on the performance of the models. For the BOW dictionary, we set it to 30,000 words with a minimum word frequency of 3, and we left the rest of the parameters to default. Regarding the TF-IDF, we set the maximum feature vectors length to

[13] http://jmlr.csail.mit.edu/papers/ volume5/lewis04a/a11-smart-stop-list/

[14] http://scikit-learn.org/

10,000 and the minimum to 3. With respect to the classifiers parameters, we kept the default setting for the NB. In contrast, for the LR, we modified only the value of the regularization parameter "C" by setting it to 10 with the balanced class-weight flag activated. For the SVM classifier, we set the decision function parameter to one-vs-rest "ovr", kernel to "RBF", "C" parameter to 10e5, balanced classes weights, and the rest were left to default.

5.2 Results and Discussion

Since we are working on an unbalanced multiclass problem, every class has a precision, a recall, and an F1 score. To combine these three values into a single value, we calculated the macro, micro and weighted average for each class as Table 3 shows. We can see that the pipeline of TF-IDF with LR achieves the highest value with a macro F1 score of 93.7% and the highest cross-validation accuracy of 96.6%. The state-of-the-art paper has achieved 94% accuracy on a different dataset that contains 1K samples (Intelliagg, 2015). Additionally, we plot the macro average precision-recall curve for four classifiers (Figure 2). The plot indicates that the pipeline of TF-IDF with LR achieves the highest precision-recall.

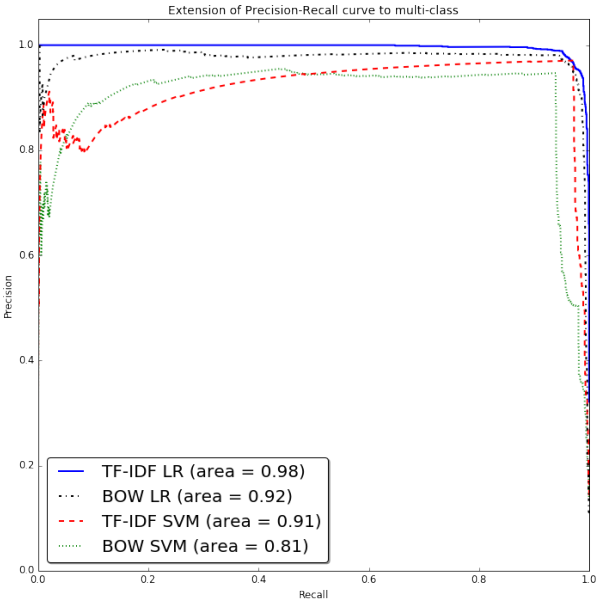

Figure 2: Macro averaging Precision-Recall curve over 4 pipelines, where the area value corresponds to the macro-average Precision-recall curve

Figure 3 shows F1 score comparison between the six classification pipelines over the nine classes. We can see that the classes *Counterfeit*

Metrics/ Methods		Average (macro)	Average (micro)	Average (weighted)	CV Accuracy
BOW LR	**P**	0,952	0,965	0,965	0,958 +/- 0,010
	R	0,889	0,965	0,965	
	F1	0,916	0,965	0,964	
TFIDF LR	**P**	0,982	0,974	0,975	**0,966 +/- 0,010**
	R	0,902	0,974	0,974	
	F1	**0,937**	**0,974**	**0,974**	
BOW SVM	**P**	0,877	0,941	0,942	0,932 +/- 0,013
	R	0,875	0,941	0,941	
	F1	0,874	0,941	0,941	
TFIDF SVM	**P**	0,983	0,971	0,972	0.960 +/- 0,011
	R	0,882	0,971	0,971	
	F1	0,924	0,971	0,970	
BOW NB	**P**	0,865	0,941	0,943	0,924 +/- 0,009
	R	0,790	0,941	0,941	
	F1	0,812	0,941	0,940	
TFIDF NB	**P**	0,530	0,885	0,855	0,863 +/- 0,012
	R	0,425	0,885	0,885	
	F1	0,460	0,885	0,860	

Table 3: A comparison between the classification pipelines with respect to 10 folds cross-validation accuracy (CV), precision (P), recall (R) and F1 score metrics for micro, macro and weighted averaging.

Credit Cards and *Hacking* have a low F1 score over all the pipelines, which is due to several reasons: firstly, the words interference between the classes. For example, the websites which offer counterfeiting credit cards services are most probably "Hack" the credit card system or "Attack" the PayPal accounts, the use sentences like "We hack credit card" or "Hacked Paypal account for sale". Moreover, those classes intersect with *Counterfeit Personal Identification* class due to their similarity from the perspective of forgery. Secondly, the number of samples that were used for training plays an important role during the learning phase, e.g. class *Violence* has 60 samples only.

Nevertheless, the learning curve for the TF-IDF LR pipeline in Figure 4 proves that the algorithm is learning correctly where the validation accuracy curve is raising up and classification accuracy is improving by increasing the number of the samples while the training accuracy curve is starting to decrease slightly. This high accuracy archived will help to build a solid model that will be able to detect illegal activity on Darknet.

6 Application and Implementation

The work presented in the previous sections has been included into an application that can be accessed and tested through a web browser. The implementation of the methods was developed in Python3 using Nltk library to stem the document text, Langdetect library to detect the language of

Figure 3: F1 score comparison for each class for 6 classification pipelines. When a bar is not shown, it means that its value is zero.

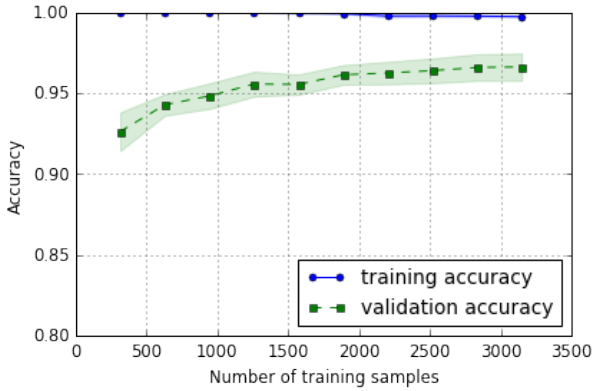

Figure 4: Learning Curve for TF-IDF with LR classifier

the documents and the Scikit-learn library to build the classifiers. The web application is made up of 3 views: one for algorithm selection, the second one for the selection of data to analyze and the third one for showing the results of the analysis (Figure 5).

The Docker image is not publicly available, neither the applications, but under email request, we will grant a temporal access to the web interface.

7 Conclusions and Future Work

In this paper, we have categorized illegal activities of Tor HS by using two text representation methods, TF-IDF and BOW, combined with three classifiers, SVM, LR, and NB. To support the classification pipelines, we built the dataset DUTA,

Figure 5: The application has three interfaces. (a) Pipeline selection. (b)The HS content preview. (c) The classification result.

containing 7K samples labeled manually into 26 categories. We picked out nine classes, including the *Others* class, that are related only to illegal activities e.g. drugs trading and child pornography and we used it for training our model. Furthermore, we distinguished the critical aspects that affect the classification pipeline results in term of text representation i.e. the dictionary size and the minimum word frequency influence the text representation techniques performance, and the regularization parameter on the LR and the SVM classifiers. We found that the combination of the TF-IDF text representation with the Logistic Regression classifier can achieve 96.6% accuracy over 10 folds of cross-validation and 93.7% macro F1 score. We noticed that our classifier suffers from overfitting due to the difficulty of reaching more samples of onion hidden services for some classes like counterfeiting personal identification or illegal drugs. However, our results are encouraging, and yet there is still a wide margin for future improvements. We are looking forward to enlarging the dataset by digging deeper into the Darknet by adding more HS sources, even from I2P and Freenet, and exploring ports other than the HTTP port. Moreover, we plan to get the benefit of the HTML tags and the hyperlinks by weighting some tags or parsing the hyperlinks text. Also, during the manual labeling of the dataset, we realized that a wide portion of the hidden services

advertise their illegal products graphically, i.e. the service owner uses the images instead of the text. Therefore, our aim is to build an image classifier to work in parallel with the text classification. The high accuracy we have obtained in this work might represent an opportunity to insert our research into a tool that supports the authorities in monitoring the Darknet.

8 ACKNOWLEDGEMENT

This research was funded by the frame agreement between the University of Len and INCIBE (Spanish National Cybersecurity Institute) under addendum 22. We also want to thanks Francisco J. Rodrguez (INCIBE) for providing us the .onion web pages used to create the dataset.

References

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Luciano Barbosa, Juliana Freire, and Altigran Silva. 2007. Organizing hidden-web databases by clustering visible web documents. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 326–335. IEEE.

Michael K. Bergman. 2001. White paper: the deep web: surfacing hidden value. *Journal of electronic publishing*, 7(1).

Alex Biryukov, Ivan Pustogarov, Fabrice Thill, and Ralf-Philipp Weinmann. 2014. Content and popularity analysis of tor hidden services. In *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 188–193. IEEE.

Wendy Boswell. 2016. How to mine the invisible web: The ultimate guide.

V. Ciancaglini, M. Balduzzi, R. McArdle, and M. Rösler. 2016. Below the surface: Exploring the deep web. *Trend Micro Incorporated. As of*, 12.

Susan Dumais and Hao Chen. 2000. Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263. ACM.

Michael Graczyk and Kevin Kinningham. 2015. Automatic product categorization for anonymous marketplaces.

Trent Hauck. 2014. *scikit-learn Cookbook*. Packt Publishing Ltd.

David W. Hosmer Jr. and Stanley Lemeshow. 2004. *Applied logistic regression*. John Wiley & Sons.

Intelliagg. 2015. Deep light shining a light on the dark web. *Magazine*.

Min-Yen Kan and Hoang Oanh Nguyen Thi. 2005. Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326. ACM.

Min-Yen Kan. 2004. Web page classification without the web page. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 262–263. ACM.

Prabhjot Kaur. 2014. Web content classification: A survey. *arXiv preprint arXiv:1405.0580*.

S Mohammadreza Khelghati. 2016. *Deep Web Content Monitoring*. Ph.D. thesis.

Matthieu Latapy, Clemence Magnien, and Raphael Fournier. 2013. Quantifying paedophile activity in a large p2p system. *Information Processing & Management*, 49(1):248–263.

Peiguang Lin, Yibing Du, Xiaohua Tan, and Chao Lv. 2008. Research on automatic classification for deep web query interfaces. In *Information Processing (ISIP), 2008 International Symposiums on*, pages 313–317. IEEE.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Daniel Moore and Thomas Rid. 2016. Cryptopolitik and the darknet. *Survival*, 58(1):7–38.

Colin Morris and Graeme Hirst. 2012. Identifying sexual predators by svm classification with lexical and behavioral features. In *CLEF (Online Working Notes/Labs/Workshop)*, volume 12, page 29.

Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system.

Umara Noor, Zahid Rashid, and Azhar Rauf. 2011. A survey of automatic deep web classification techniques. *International Journal of Computer Applications*, 19(6):43–50.

Claudia Peersman, Christian Schulze, Awais Rashid, Margaret Brennan, and Carl Fischer. 2014. icop: Automatically identifying new child abuse media in p2p networks. In *Security and Privacy Workshops (SPW), 2014 IEEE*, pages 124–131. IEEE.

Dorit Ron and Adi Shamir. 2014. How did dread pirate roberts acquire and protect his bitcoin wealth? In *International Conference on Financial Cryptography and Data Security*, pages 3–15. Springer.

Dakota S Rudesill, James Caverlee, and Daniel Sui. 2015. The deep web and the darknet: A look inside the internet's massive black box. *Woodrow Wilson International Center for Scholars, STIP*, 3.

Gerard Salton. 1971. The smart retrieval systemexperiments in automatic document processing.

Weifeng Su, Jiying Wang, and Frederick Lochovsky. 2006. Automatic hierarchical classification of structured deep web databases. In *International Conference on Web Information Systems Engineering*, pages 210–221. Springer.

Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. 2002. Web classification using support vector machine. In *Proceedings of the 4th international workshop on Web information and data management*, pages 96–99. ACM.

Johan A.K. Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

Xuefeng Xian, Pengpeng Zhao, Wei Fang, Jie Xin, and Zhiming Cui. 2009. Automatic classification of deep web databases with simple query interface. In *Industrial Mechatronics and Automation, 2009. ICIMA 2009. International Conference on*, pages 85–88. IEEE.

He-Xiang Xu, Xiu-Lan Hao, Shu-Yun Wang, and Yun-Fa Hu. 2007. A method of deep web classification. In *2007 International Conference on Machine Learning and Cybernetics*, volume 7, pages 4009–4014. IEEE.

Pengpeng Zhao, Li Huang, Wei Fang, and Zhiming Cui. 2008. Organizing structured deep web by clustering query interfaces link graph. In *International Conference on Advanced Data Mining and Applications*, pages 683–690. Springer.