Content-based Classification of Fraudulent Webshops

Mick Cox
University of Amsterdam
mick.cox@os3.nl

Sjors Haanen
University of Amsterdam
sjors.haanen@os3.nl

Supervisors:
Marco Davids (SIDN)
Maarten Wullink (SIDN)

***Abstract*—In this research, we evaluate the possibility of classifying fraudulent webshops on the basis of website content. Fraudulent webshops form a problem on the Internet and the Dutch Consumers Association (Consumentenbond) reports that at least one out of five webshops in the .nl top level domain (TLD) acts in a fraudulent manner. Content-based classification can be part of the detection and mitigation strategy. This method demonstrates the viability of this approach, and follows multiple steps. First, we identified prerequisites necessary for running a successful fraudulent webshop. Second, based on these prerequisites, features were engineered to classify these fraudulent webshops. Third, a classification model was trained, resulting in an F_1-score of 99.15% using 10-fold cross validation. Fourth, this trained model classified an unlabelled dataset of 4.9M domains. Although results include many false positives, we estimate the precision to be 30.1%. This would imply that out of 32,815 positively classified domains, 9,877 are correctly identified fraudulent webshops. Furthermore, we also estimate that results in general will improve if more optimisations are conducted. Ultimately, our contribution is constituted on the demonstration that a content-based classification method is viable and the presentation and analysis of characteristics and features used. The software resulting from these experiments online[1].**

I. Introduction

As an early adopter of the Internet, the Netherlands has a strong online presence. As an indication: it's capital city is hosting AMS-IX, one of the largest internet exchanges in the world; the average mobile internet speeds globally ranks in the top 10, and broadband speeds in top 15; and lastly, with 5.8 million registered domains, the .nl top-level domain (TLD) is the 10th largest TLD (Q4 2017) [1]. These developments foster a healthy platform for trade, which is supported by the global ranking on (B2C) e-commerce by the United Nations Conference on Trade and Development, in which the Netherlands is ranked 4th [2]. Although these indications seem positive, a well-developed e-commerce presence also attracts actors with malicious intent. Abuse is present on the web, most of which is about privacy, identity theft and fraud. Examples of which are phishing, the spreading of malware and typosquatting. One of the more recent phenomena is fraudulent webshops, which is the subject of this study.

Fraudulent webshops lure Dutch consumers into buying products of popular brands. However, after payment, the customers receive either counterfeit products or nothing at all. In the process, the customer often has to pay with a credit card and is forced to share personally identifiable information, opening the door for identity and credit card theft. This phenomenon is increasingly becoming a problem and lately gaining attention from the Dutch Consumers Association (Consumentenbond). In one of their recent publications, they estimate that at least one in five of the 90,000 webshops targeting Dutch customers is fraudulent. [3]. Fraudulent webshops are a specialised case of Internet fraud, to which many people fall prey to. The Dutch hot-line for Internet fraud (Landelijk Meldpunt Internetoplichting) reported that 38,343 Dutch citizens filed complaints of Internet fraud in 2017 alone, while these account only for 47% of the total victims [4]. Furthermore, the Dutch police reports a sharp increase in the number of fraudulent webshops in that year, which indicates that operating these webshops is a lucrative business.

A. Motivation

Stichting Internet Domeinregistratie Nederland (SIDN) is a nonprofit foundation and the registry operator responsible for the .nl ccTLD. As one of the authoritative organisations involved in Dutch internet infrastructure, SIDN contributes to the general safety of Dutch internet usage. Having recently identified the problem of fraudulent webshops, SIDN has set out to understand the problem better. It is in the interest of the ccTLD operator, Dutch domain name registrants, as well as the end users, to maintain the high level of security and stability of the .nl ccTLD. This requires effective prevention and detection measures. Since the number of fraudulent webshops is growing, there exists a need for a detection mechanism specifically for this kind of abuse. Therefore, SIDN has provided a dataset containing the index pages of all Dutch domains for this research project. This research is conducted within the context of dissertation of the Master studies System and Network Engineering, at the University of Amsterdam.

B. Problem Definition

The problem of fraudulent webshops is still unclear from many perspectives. First, one cannot, with complete certainty and on first sight, state whether a webshops is fraudulent or not; as this ultimately necessitates a purchase. Second, it is unclear who exactly is operating these webshops. Third, no single organisation by itself is entirely responsible for combating the problem. Furthermore, preventive actions against suspected fraudulent webshops is often not possible, and can

[1] http://www.github.com/mjrc/web_or_nep/

entail serious liability issues. For instance, simply removing suspicious webshops from the DNS zone file is restricted by the SIDN abuse policies. SIDN depends on the domain name registrars to take action in case of suspicious activity. Not all registrars are able or willing to address the abuse. In other words, combating this problem is far beyond self-evident, and a more in-depth understanding of the nature of the problem is required before formal steps can be undertaken.

Related work, as will be discussed in section II, is approaching the problem from a "meta perspective", and studies WHOIS information, network and service provider information, such as geolocation, IP ranges and more. To aid this effort, this research aims to study webshops on the basis of web content. This research answers whether it is possible to perform recognition and identification of webshops in an automated fashion. Therefore, we pose the following question:

> *Is it possible to reliably classify fraudulent webshops in the .nl TLD based on web content?*

In this research question, web content refers to the parts of the website which are directly presented to the end user. This includes the natural text occurring within the pages, images, HTML structure and the URL.

Furthermore, classification entails the separation of *non-fraudulent webshops* from *fraudulent webshops*, which calls for an approach involving classification by machine learning. Non-fraudulent webshops refer to all other legitimate websites, other than fraudulent webshops. Although it seems more intuitive to distinguish fraudulent webshop from legitimate webshops, this would not work in the context of SIDN, in which the input could constitute any possible website.

Following from this, a machine learning approach entails the use of a classification algorithm and a trustworthy labeled dataset. Given some set of fraudulent webshops, one can speculate that any website meeting the same pattern must be fraudulent as well. This introduces the following two problems: First, one cannot judge a webshop to be fraudulent without resorting to a purchase, as previously mentioned. Second, by following this pattern or structure, whilst extending the training and testing dataset and during feature engineering, a bias can arise which could impact the effectiveness of the classification. In other words, in building the classifier, we have to find a balance between certainty and effectiveness. The approach we follow in order to achieve this is detailed in section III.

C. Paper outline

After having introduced the problem and topic at hand in section I, we continue by outlining prior and related work in section II. In order to correct for a possible bias, we identify prerequisites to the business model of fraudulent webshops, as part of our approach. This approach is detailed in section III, which is followed by a dedicated section to these prerequisites in section IV. From these prerequisites, we construct features, which we describe in section V: Feature Engineering. Afterwards, we describe the results from our experiments in section VI and cover the implications in section VII. Finally, we conclude the research and acknowledge key parties in section VIII and section IX, respectively.

II. Related Work

In order to perform classification, a selection of features that represent the characteristics of fraudulent webshops must be chosen. Sahoo, Liu, and Hoi recently released a survey which categorises and reviews contributions in detecting malicious domains using machine learning [5]. In this survey, features found in other studies are categorised as follows: Blacklist features, URL-based lexical features, host-based features, content-based features, and others (context, popularity, etc.). As for content-based features, Hou, Chang, Chen, *et al.* found features in the HTML documents, including the length of the document, (distinct) word count, and the use of string concatenation [6]. Additionally, they used features in the DHTML client side scripts like encoding functions and the use of the "eval" and "exec" function. These features were used to compare different classification algorithms, ultimately to classify websites which try to compromise the victims' computer systems. Since fraudulent webshops have other goals than websites attempting to compromise end systems, their website characteristics are different. This means that other features may be needed in order for algorithms to be able to classify domain names as fraudulent webshops.

Lexical features are another category that Sahoo, Liu, and Hoi named in their survey. Features used by the mentioned studies are specifically focused on the URL string itself. Most of the approaches used the bag-of-words model, which is a Natural Language Processing (NLP) technique in which a text (in this case the URL string) is seen as a collection of words. This model uses the frequency of words to classify on, though valuable information like grammar and the order of words are not taken into account. Since web content includes natural language, we aim to investigate whether NLP can help in classifying domains.

Giovane C. M. Moura, Moritz Muller, Maarten Wullink, and Cristian Hesselman from SIDN Labs developed a system which can detect several types of abuse within a DNS zone, including fraudulent webshops. This is made possible by analyzing both domain registration and global DNS lookup patterns of a TLD, selecting useful features and classifying domains by employing the k-means clustering algorithm. Since SIDN runs the .nl authoritative servers, SIDN Labs could implement this system by monitoring the DNS traffic of the .nl zone. As future work, the researchers suggested including more features including HTML and content analysis.

The Consumentenbond helps consumers identify fraudulent webshops by summarising several points to look out for [3]. Fraudulent webshops often seem to use odd domain names, or sometimes use product or brand names in the domain name; fraudulent webshops tend not to use HTTPS, have a generic shop logo, focus on selling luxury items, offer high discounts, show mistakes in language and grammar, do not provide

any contact details, utilise mandatory account creation, force payment by credit card and lack certification marks. Especially strange sentences and mistakes in language, indicate that the operators are non-Dutch.

The winners of the *Dutch Open Hackaton 2018* [8] developed a proof of concept named CrimeBusterBot [9]. This tool could identify fraudulent webshops by using several external sources like WHOIS information and data from the Dutch police, though they only did some basic checks on the web content like looking for the string "shopping_cart".

The prior work mentioned in this section identified fraudulent webshops by using external sources and measuring DNS traffic. Most of the studies using web content for classification attempt to find domains which distribute malware, whereas this research project focuses on finding fraudulent webshops.

III. Method

A. Overview

As previously mentioned, classification of fraudulent webshops necessitates a thorough approach. In this section, we evaluate the datasets used and constructed, evaluate its limitations and identify a countermeasure used during feature engineering. Lastly, we subsequently outline the experiments conducted in this project.

B. Datasets

In order to perform the research of this project, two labeled sets are composed: A set of known fraudulent webshops and a set of general websites which are *not* fraudulent webshops. In this project, these sets are called the *nep* and `web` sets respectively. All websites have a .nl domain name, or are at least targeted to Dutch visitors (in some cases HTTP requests to Dutch domain names are redirected to domain names in other TLDs). Also, each website is checked manually to make sure it is in the right set. The set of fraudulent webshops is derived from several sources. The first one is a list of 2,000 websites composed by the Consumentenbond [10]. Despite the fact that a large part of the websites in the list have already been taken down by the respective registrants, we found that the remaining ones are still online and usable for our set. Another source originates from the output of the CrimeBusterBot, which is discussed in related work, section II. With these webshops as starting point, we were able to extend our combined list of fraudulent webshops by searching the Web for specific strings consistently found in these webshops, such as "Nieuwe artikelen voor June". With these sources combined, we were able to construct a list of 3,369 alleged fraudulent webshops.

The other set is composed by using a dataset provided by SIDN. This dataset is the output of SIDN's web crawler [11], which is a complete HTTP-crawl of all the .nl domain names (5.7 million) as of 1 July 2018. The dataset consists of the index pages of the corresponding domains including HTML content and scripts, along with TLS information and metadata about each web server. It also performs several checks, like recognising types of websites (e.g. webshops, CMS or forums) and checking whether a privacy policy page is present. From this dataset we randomly selected a subset of 3,600 domain names, of which 3,557 remained after manual checking and sanitising the dataset from false positives. We labeled this subset as the set of general websites.

Some domain names belong to a chain of redirects to other domains. In these cases, we only included the last domain of the chain, which is the URL that the end user eventually sees in the address bar. This is because some of our features estimate the semantic relationship between domain name and the page content.

Aside from the labeled sets, we also define an unlabelled set called the *zone* set: From the 5.7 million domains in the SIDN dataset, we extracted the domains containing an index page, and excluded the domains which forward to the same domain. The resulting *zone* set contains 4,294,557 domains.

C. Main approach

As a limitation to the datasets available, we recognise that the *nep* dataset is likely biased. Not only is it unknown how the Consumentenbond and the CrimeBusterBot exactly constructed their lists, we also manually extended the set by searching for commonly reoccurring strings. Furthermore, we noticed that many of the domains in their lists likely share the same operator(s), since in many cases only the themes and logos differ while the website structure and texts are the same. Hence, we cannot be sure whether the combined set is fully representative for all types of fraudulent webshops operating in the .nl TLD, and therefore the set is likely not perfectly suitable for classification. When engineering features to this end, purely matching on specific technical implementations (e.g. the usage of the popular Zencart e-commerce software), or commonly found strings or patterns, is not a reliable and flexible classification mechanism. Not only do we risk missing a significant subset of fraudulent webshops due to initial dataset bias; also, operators could just slightly adjust their code in order to circumvent detection. Therefore, we take a slightly different approach in this research project. First, we identify essential prerequisites in order to successfully deploy fraudulent webshops. Then, we model features for classification based on those prerequisites. As such, we attempt to create a model which corrects for the possible bias in our initial dataset. Furthermore, ancillary to this approach is the implication that any effort by an operator to remain undetected goes at the expense of the business model or likelihood of success. The effectiveness of the features will be tested by two experiments, in which we classify the domains in our datasets.

D. Experiments

In order to test our approach, we setup two experiments, denoted *Experiment I* and *Experiment II*.

In Experiment I, we train and test a classification model on our labeled *web* and *nep* sets using K-fold cross validation. This technique implies that we subdivide our dataset in K equal parts, and perform K tests for each of the parts, such that

we test on only one subset and train on the remaining $K-1$ subsets. K-fold cross validation is a common technique for evaluating estimator performance and builds further on train-test splitting, such that the accuracy of a model can reliably be tested and the likelihood of overfitting is reduced.

In Experiment II, we classify the *zone* set using the classification model trained in Experiment I. With this, we again evaluate the effectiveness of our classifiers, but this time on an unlabelled set. Afterwards, we manually validate these results by checking a sample of 1,000 classifications.

E. Motivation classification algorithm

During feature engineering and earlier iterations of our experiments, we tried seven classification algorithms, as earlier iterations of did not warrant a focus on the algorithm yet. We combined the classification results of multiple algorithms into a confidence ratio, and as such were able to classify new examples during Experiment II on a majority vote based system. Although this method seems interesting from the perspective of identifying fraudulent webshops, we experienced several downsides to this approach. In the end, we chose to base our results on the AdaBoost Classification algorithm (Adaptive Boosting). Not only because it was the best performing classifier in our experiments when using the default parameters; it also is an ensemble function which involves a weighted combination of classifiers internally, and thus it mimics the voting behaviour earlier constructed. Boosting entails the recursive training of classifiers such that later classifiers focus on examples which previous classifiers did not get right. As such, many weak learners can be combined into a strong classifier.

During the AdaBoost training process, the algorithm reduces the amount of dimensions of the dataset by selecting only features which improve the predictive power of the model. During the following section, table I shows used features and their weights. This table shows some features weights being zero, which depicts this functionality.

F. Metrics

The metrics we use for evaluating the effectiveness of our model are accuracy, recall, precision and F_1 score. Accuracy is the ratio of correctly classified samples to falsely classified samples. Recall is the ratio of true positives to false negative, i.e. how many of the positive (*nep*) class where we able to recall. Precision is the rate of true positives to false positives, i.e. how many of the predicted positives were in fact true. At last, we calculate the F_1 score, which is the harmonic average of the precision and recall defined as follows:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{1}$$

As an example: let's say there are 100 fraudulent webshops within a dataset of 10,000 samples. Imagine classifying every sample as positive (*nep*). Since we extracted all 100 fraudulent webshops, we would achieve 100% recall, but we would only be precise 1% of the time. In this example, the F_1 score would be 0.04.

IV. Fraudulent webshop prerequisites

As mentioned, a classification method based on technical or syntactical features can be a brittle approach. Furthermore, it can enhance the classification bias already existing in our dataset. Therefore, we attempt to construct features around the prerequisites necessary for a fraudulent webshop business model to work. Before covering these prerequisites, a brief description of the website outline and structure is necessary. The examples offered in the following paragraphs are based on the *nep* and *web* datasets described before. Figure 1 depicts a picture of a product featured on a fraudulent webshop.

Most webshops encountered are set up with a minimum set of pages. Aside from product and category shopping pages, these include the index page, contact page, privacy policy page, shipping & returns page and a sitemap. Furthermore, most webshops include functionality for creating and logging in to a user account. The index page most often displays an overview of currently available products, mostly dedicated to the season. Translation errors can often be found here including such as "Nieuwe artikelen voor June", in which the month "June" is not translated to Dutch. Widget areas are often display either category information or featured products. The footer generally holds links to fore-mentioned pages and, depending on the webshop framework, a list of products or categories. Images of products are generally resized and compressed by webshop framework.

We assume that the fraudulent webshop operators' goal is to make money by either not sending the ordered product or sending a counterfeit product. Furthermore, operators could also commit identity or credit card theft from the information gathered. Based on this assumption, we can identify three main prerequisites which are needed to successfully run fraudulent webshops. These prerequisites include: First, *attracting customers* into making a purchases. Second, being found on search engines, by means of a high *search engine score*. Third, leveraging high *scalability* to generate many webshops and increase the probability of success.

A. Customer attraction

As for all online stores, fraudulent webshops need to look attractive in order to tempt visitors into making a purchase. This achieved in a number of ways.

First, popular brands are offered with high discounts, as depicted in figure 1. Second, the products offered generally appeal to a large audience because a multitude of reasons, namely: products are always in stock, offered products generally need frequently replaced, or can easily be purchased online. Commonly offered products include shoes and sneakers, hats and caps, sunglasses, bags, backpacks and football jerseys. Third, fraudulent webshop operators attempt to develop trust by showing a webshop logo, images of common brand products, logos social media networks, logos of credit card companies and parcel companies. Remarkably, Dutch

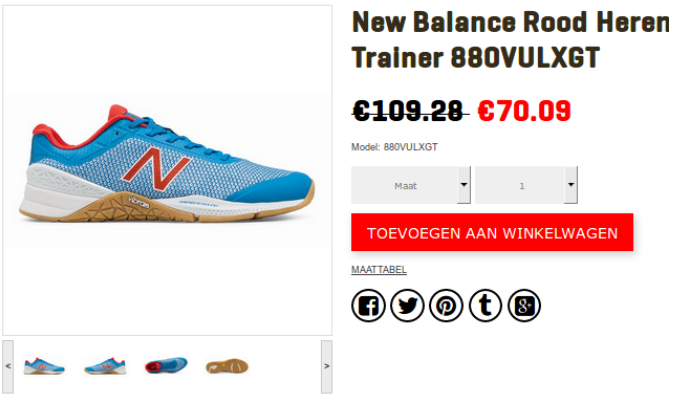

Figure 1. An attractive offer on a fraudulent webshop.

Figure 2. Logos of fraudulent webshop are often generic or directly generated from the domain name.

alternatives, such as the iDeal payment service and the PostNL mail and parcel company, are rarely mentioned. In essence, the offers on fraudulent webshops just looks too good to be true, though many people seem to fall for it, as shown by the number of filed complaints as mentioned.

B. Search engine score

When reviewing the publicised lists of suspected fraudulent webshops, it becomes self-evident that in general, domain names have little resemblance with webshops or its contents. For example, the domain *www.autorijschoolmathieu.nl*, meaning "driving school Mathieu", offers shoes instead of driving lessons. We expect nobody will consciously visit this domain directly in order to buy shoes. Therefore, we expect all website traffic originates from search engines after a user consciously searches for keywords relevant to the webshop content.

The most recent theory suggests that fraudulent webshop operators utilise register recently expired domains [3]. Considering these domains have been active and maintained earlier, search engines have already verified the validity of the domain and given it an index. Although the web contents change, the index of the previous domain owner still exists, and the webshop is easier and more often found.

One known method of registering soon-to-expire domains names, is by the services of dropcatchers. Dropcatchers are organisations that try to register (catch) a domain after being release from quarantine. Quarantine is the time a registry holds on to a domain name after being released by the previous customer. Companies offering such services include *www.domainorder.nl* and *www.dropcatcher.co.uk*.

The search engine score mainly depends on backlinks, i.e. links from other domains to the domain. Hence, if an expired domain is registered, backlinks are still intact. Fraudulent webshop operators eagerly take advantage of the reputation of these links. This reputation is also known as *link juice*.

Metrics from backlinks and search engine optimisation at large are sold by companies such as *www.majestic.com*. Dropcatchers use these related metrics to promote the value of domains.

Besides using recently expired domain names, we have found that fraudulent webshop operators use HTTP meta description and meta keyword tags to increase findability in search engines. These keywords include names of brands and words relating to discounts.

We have seen other known methods of search engine optimisation, though only in a few cases. Examples are web enhancement for mobile devices and a HTTPS version of webshops.

C. Scalability

We reason that the business model of fraudulent webshops is only feasible when operators can leverage high scalability. By this, we mean the following: Given the choice between deploying (fraudulent webshops) more and deploying more often, versus optimising just a few, we expect fraudulent webshop operators will choose the former.

Following from this, every fraudulent webshop risks a change of being taken down. This risk increases as the webshop becomes better known. Hence, any time spent in manually personalising a website, in order to increase its efficiency, is time which cannot easily be recycled.

This results in a generic appearance, of which multiple examples exists. First, we notice the same webshop software is commonly used, namely Zencard. However, other frameworks such as Prestashop and Woocommerce installations also occur. Other examples of similar software can be found in Javascript libraries used for ancillary features such as the webshop product image lightbox or image carousels. Second, product images are reused and on multiple occasions, product images contain a watermark of another .nl domain name, indicating the product image was used before. Third, exactly the same or very similar text on websites is reused. Fourth, a generic webshop logo is used, and likely generated using the same software. Examples of these are shown in figure 2.

In a sense, they treat webshops as kettle, not as pets

Just as a fraudulent webshop, a legitimate webshop operator is interested in a high search engine score. However, in contrast to the fraudulent webshop, the operator is generally willing

to invest manual work in optimising and personalising. An example is integration between the webshop and external software services to extend the functionality, such as Google Analytics, e-mail marketing such as Chipmunk or external payment service providers. These external software services often require valid personal information and their account registration procedures are often not automatable. The time spent on such setup tasks eventually adds up which adds pressure on the feasibility of the current business of fraudulent webshops.

Furthermore, after having manually registered user accounts on several fraudulent webshops, we noticed the automatic user account confirmation e-mail originate Gmail accounts. After sending an e-mail to this address, we received an e-mail back, indicating the inbox to be maintained. Considering many webshops use the same Zencard installation, we have written a script for the Selenium browser automation framework in order to register users accounts *en masse*. After registering many user accounts on many webshops, we have found that several fraudulent webshop operators use the same e-mail address for many domains. In total, we have registered with, and received e-mails from 2,761 webshops. Of the e-mails received, only 61 e-mail addresses are unique, though many show reoccurring patterns such as sharing a similar prefix or enumerating over numbers. Figure 3 depicts the 12 most common e-mail addresses. A complete list can be found on our project code repository.

Please note that these e-mail addresses only originate Zencart installations using the exact same registration form. By no means is this representative of all fraudulent webshops.

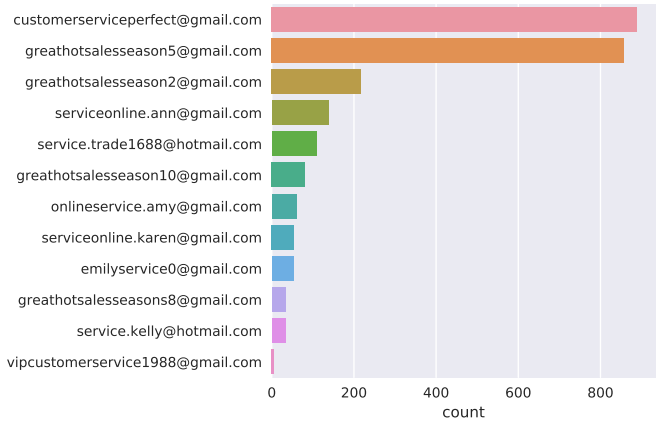

Figure 3. The twelve most frequent e-mail addresses found upon registering a user account.

V. Feature Engineering

A. Overview

In this section we cover feature engineering, one of the main contributions of this research. As described previously, these features follow from prerequisites described earlier: *customer attraction*, *search engine score*, and *scalability*. For each feature, we describe the reasoning behind the feature, the way we measure it and evaluate the performance. A list of all features used is shown in table I.

Figure 4 depicts a violin plot of the non-boolean features. Each feature depicts two distributions, one on each side of the axis. The left side depicts the *nep* set and the right side depicts the *web* set. Note that within this plot, each feature is rescaled to a common scale in order to make the distributions visible. The reason why some distributions are much smaller than others and start at the bottom, is because it represents a count, starting at 0, and has high outliers.

Figure 5 depicts a barplot of the boolean features. The y-axis depicts the relative amount of samples within the *nep* and *web* datasets, for which the feature is true.

Finally, for a complete statistical overview of all non-boolean features, see table V in Appendix A.

Table I
Feature Overview

No.	Name	Datatype	Weight
01	analytics	Boolean	0.1208
02	currency	Boolean	0.0000
03	currency_count	Integer	0.1048
04	distance_edit	Float	0.0987
05	distance_jaccard	Float	0.0419
06	image_count	Integer	0.0289
07	lexical_count	Integer	0.0161
08	lexical_diversity	Float	0.0305
09	lexical_unique	Integer	0.0421
10	links_external	Integer	0.0615
11	links_hash	Integer	0.0538
12	links_intent	Integer	0.0000
13	links_internal	Integer	0.0407
14	links_mailto	Integer	0.0401
15	links_map	Integer	0.0000
16	dom_title_dist_sonar	Float	0.0210
17	dom_title_dist_wiki	Float	0.0338
18	dom_title_sim_sonar	Float	0.0230
19	dom_title_sim_wiki	Float	0.0421
20	metadesc	Boolean	0.0000
21	metadesc_count	Integer	0.0071
22	metakeyword	Boolean	0.0198
23	metakeyword_count	Integer	0.0125
24	metaog	Boolean	0.0100
25	phone	Boolean	0.0000
26	place	Boolean	0.0000
27	scripts	Boolean	0.0000
28	scripts_count	Integer	0.0600
29	sm_deep_link	Boolean	0.0779
30	sm_link	Boolean	0.0000
31	styles	Boolean	0.0000
32	styles_count	Integer	0.0129

B. Syntactical domain-title similarity

As illustrated before, in many cases domain names have pre-existed and are re-registered to use for the application of a webshop, there is often a mismatch between the domain and the content of the website. In general, it is common pattern to find the domain name, in some way, be a representation of the content the website. For instance, if the domain name forms the name of an organisation, then one can expect that name to be included in the content of the website, such as in the HTML title. This is usually not the case for fraudulent

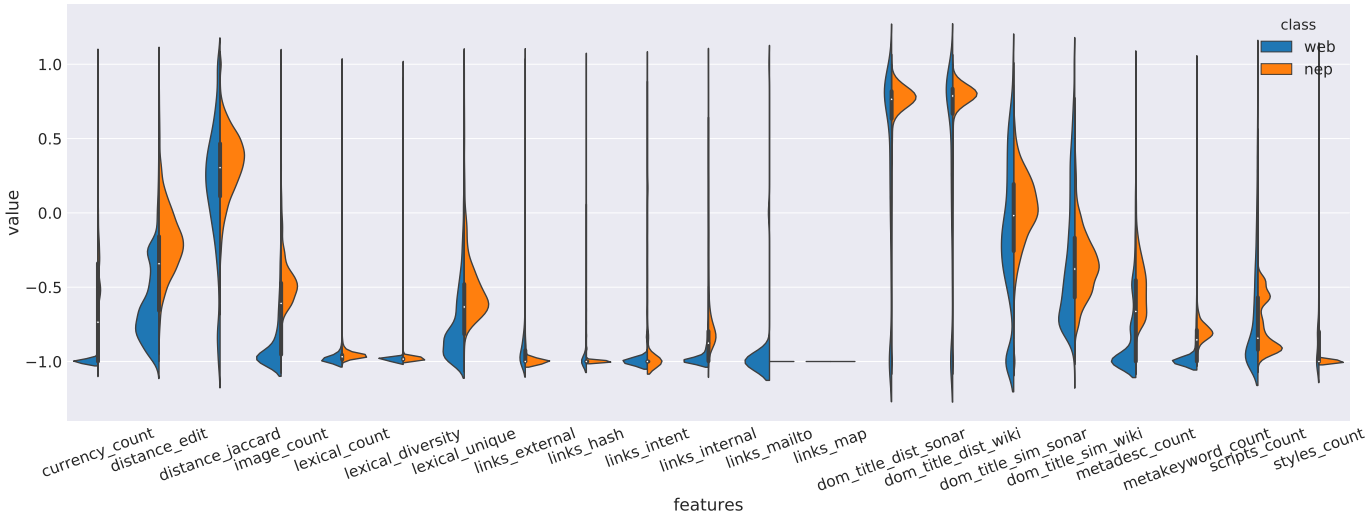

Figure 4. This violin plot depicts the web and nep distribution of each non-boolean feature. Note that these distributions have been rescaled in order to make them visible within the same plot.

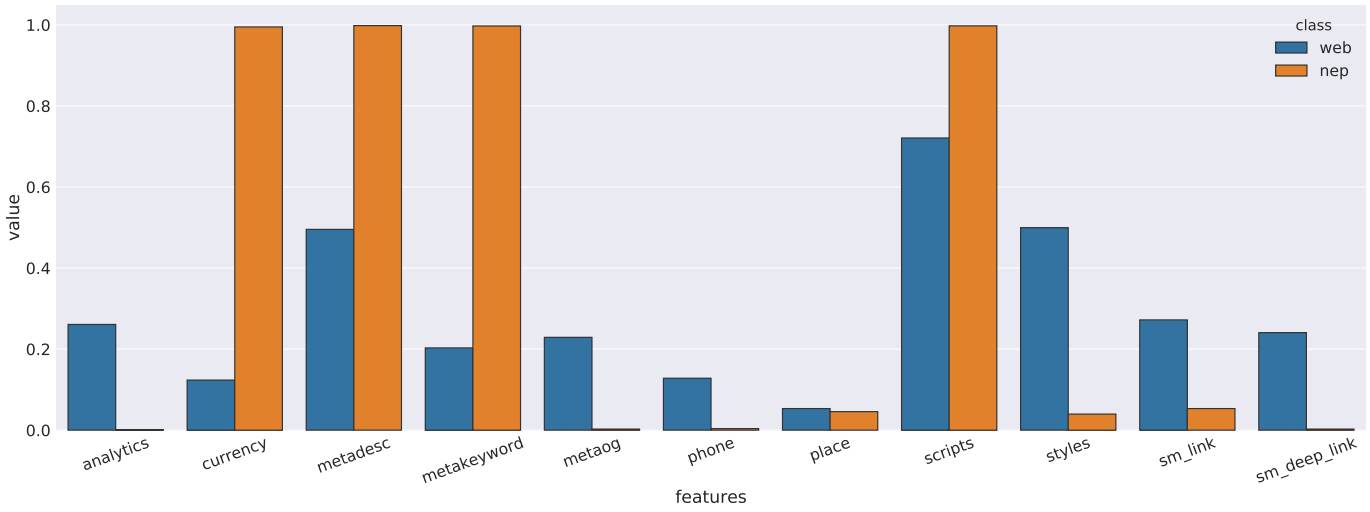

Figure 5. This barplot depicts the web and nep distribution of each boolean feature. Note that the relative amount of features is depicted for which the value is true.

webshops, which often carry HTML titles that do not resemble the domain name at all. In order to target this dissimilarity, one can calculate a string distance between the domain name label and the HTML title. Doing so over a large enough dataset results in a clear distinction between the distributions of the *nep* set and *web* set, as is visible in figure 6.

C. Semantic domain-title similarity

String distance measurements can measure the syntactic difference between words. However, they cannot measure the semantic distance or similarity between two words. Two words may be similar in writing, their meaning can vary wildly. For measuring semantic relationships, we used word2vec [12]. Using word2vec, or comparable word embeddings such as fastText and GloVe, one can represent words in a high dimensional vector space. This is done by training a model on the co-occurence of words within each sentence, for many documents in a large corpus of text documents. Pretrained models are widely available for many languages. We used a model which was pretrained on the SoNaR500 corpus [13] and on the Dutch Wikipedia [14].

We use gensim [15] for calculating the *cosine similarity* and *Word Mover's Distance* [16] between a tokenised and sanitised version of the domain name label and the HTML title, for both the Wikipedia and SoNaR500 corpus. These features are either prefixed with *dom_title_dist* for the W.M. distance, and *dom_title_sim* for the cosine similarity.

In order to arrive at a tokenised and sanitised version of the domain name, we follow the following method. First, we generate all possible substrings from the main domain label. Second, we filter these substrings to a dictionary in order to return a set of valid words. Third, we remove all common

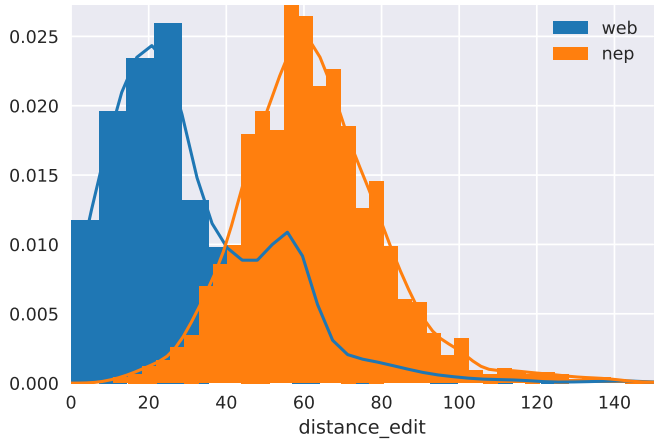

Figure 6. Distribution of the edit distance between the domain name and HTML title, for the nep and web sets.

stopwords, which are words not contributing to the meaning, such as determiners and prepositions. Fourth, we are left with a set of valid words but among which are still many “false positives”, i.e. words that are indeed valid but are not relevant to the domain name itself, mostly by being a subword of an actual word. We can filter these by taking the longest word inside our set, and filtering out any other words which are a subword of the longest word. If we continue doing this, until there are no more words inside the set for which there also exists a superword inside the set, then we are done.

For instance, *autorijschoolmathieu* will be broken down into many substrings, among which are strings which are valid words, but also many bogus strings. After filtering against a dictionary and against stop words, we end up with a set of valid words. These include words such as ’ijs’, ’auto’, ’mathieu’, ’autorijschool’, ’rij’, ’ij’, ’school’ and ’rijschool’. Notice that most words are false positives, such as ’ijs’ and ’auto’, in the sense they are subwords of ’autorijschool’. The last step is to take the longest word, ’autorijschool’ in this case, and filter every word from the set which is a subword of ’autorijschool’. By doing this, we can remove all words except ’autorijschool’ and ’mathieu’, which are our final words.

Next, we roughly apply the same method to the HTML title. First, we tokenize the title. We don’t need to generate substrings as we can assume the title is already in sentence form. Second, we filter against our dictionary and list of stop words, such that we end up with a valid list of words.

For instance, the HTML title of autorijschoolmathieu.nl is “Damesschoenen van aQa COGNAC (A3433-Z23A25) / Van Mierlo Schoenen”. If we tokenise and sanitise these, we end up with four different words: "Mierlo", "Damesschoenen", and "COGNAC".

Finally, we can calculate a similarity score between the two set of words ("autorijschool", "mathieu") and ("mierlo", "damesschoenen", "cognac"). This would result in 0.301 for the SoNaR500 based model and 0.211 for the Wikipedia based model.

Note that words such as "Mierlo" and "Mathieu" are not filtered by our dictionary, although being natural names. In our case, we used the dictionary which is part of the vector space and since these corpora are very large, natural names are included as well.

At last, when evaluating the performance of the semantic similarity between the *nep* and *web* sets, we can see that the distributions are indeed distinct. The quality of this feature is largely bound by the quality and amount of preprocessing done on the input data.

D. HTML meta tags

HTML meta tags are an important feature for different applications, such as the search engine score and the integration with third party services. By means of defining description and keyword meta tags, search engine operators, such as Google, are better able to index the website. As explained earlier, in section IV-B, the search engine score is an important requirement for webshops in general. Considering that defining meta tags is a low effort task, and that search engine score is especially important for fraudulent webshops, we expected each fraudulent webshop to have these enabled, while this is not often the case for websites in general. III-B. This indeed shows for the metadesc and metakeyword features in the barplot. Here, we can clearly see that nearly all domains in the nep set utilise both kinds of metatags, but only half or even less of the domains belonging to the web set.

Furthermore, we also identify the existence of *meta:og* tags. These tags indicate adaption to the Facebook platform and enable rich bodies. Existence of these meta tags indicates the website operator has optimised the website content to a possible appearance on Facebook. This optimisation is generally implemented on websites of which content is expected to be shared on Facebook, i.e. popular websites. Considering the extra effort in setting up these tags, we did not expect fraudulent webshops to use these tags. This also show in the barplot. We can clearly see that a little more than 20% of domains in the web set use Open Graph tags, whilst nearly 0% of the *nep* set do.

E. Pattern matching

When operating a legitimate webshop, it is often common practice to provide contact and business information on the website. As a matter of fact, payment providers in The Netherlands are required to verify if the business bank account, chamber of commerce identification number, VAT number, and contact details are clearly visible on the website. Hence, pattern matching for these values could indicate the legitimacy of a webshop.

We utilised regular expressions for matching Dutch telephone numbers, postal codes, address notations, IBAN bank numbers, Dutch VAT (BTW) numbers and regular strings indicating this number. Unfortunately, we discovered that regular expressions for addresses are too expensive to be utilised efficiently and postal codes yield too many false positives.

Hence, we eventually abandoned the use of these two patterns while keeping the others.

We initially expected this to be a potent feature in classifying fraudulent webshops, but as the barplot shows, the differences between the *nep* and web classes is not all that much. Some fraudulent webshops indeed contain valid Dutch address and place names. These addresses were often also featured on online real estate market places, such as *www.funda.nl*. We suspect these are simply copied such websites in order to generate more trust.

F. Currency symbols

We count the number of currency symbols such as Euro (€) and Dollar ($) on the index page. The idea behind this is twofold. Firstly, the currency symbol can represent a displayed product. This representation is interesting as it can be used to reason about web content. Namely, when visiting a website, a visitor is expecting a certain amount of web content; a deviation from this expectation might queue a suspicion towards the webshop, which we assume the operator intends to avoid. For instance, having a large amount of products on display is generally uncommon. On the contrary, although having only few products on display is more common, legitimate webshop operators tend to highlight displayed products and offer other web content as a compensation in order to engage customers. Creating such content does not scale since it is time-consuming and case dependent. Hence, we assume that fraudulent webshops generally do not contain this content. Therefore, the amount or products (and thus currency symbols) displayed on the index page is expected to not vary a lot.

Secondly, to attract and convince customers into purchasing products, discounts are offered. Generally, discounts are communicated by displaying a product price twice: the original price, and a discounted one. Hence, the currency symbol is displayed twice.

Although a rather crude metric, the currency count seems to be an effective measure in setting apart a website from a webshop, and, to a certain degree, in setting apart a legitimate from fraudulent webshop.

G. Image count

Similar to currency symbols, images are also used to represent products on sale. An image count may be used as a crude approximation for the amount of products on sale. Furthermore, as described previously, we expect the amount of products displayed between webshops to vary little.

Web pages contain images for many other reasons than solely to represent products on sale. However, for webshops in general, a strong correlation between the amount of currency symbols and the amount of pictures, gives a better representation for the amount of products on sale. Furthermore, for the same reasons as for currency symbols, we expected the amount of images between fraudulent webshops to vary little. The relationship between the amount of currency symbols and the amount of image is shown in figure 7. As can be seen, the *nep* dataset increases linearly as the amount of products displayed increases.

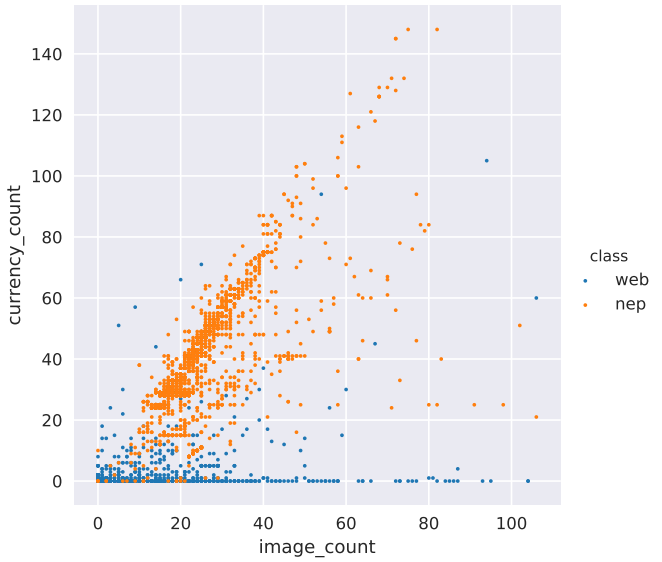

Figure 7. Estimating products by correlating images and currency symbols.

H. HTML Anchor tags

Links, or HTML anchor tags, are a fundamental feature of HTML and the Web. By traversing these anchor tags, one can execute HTTP requests for different web content, thus visiting different pages and sites. By looking at the amount and direction of links, we can approximate the size of the website. For instance, we reason that smaller sites like personal blogs and local community websites contain fewer links on their index page than bigger sites like wikipedia.org. Furthermore, the direction or kind of link is also relevant. We subdivide links into internal, external, hash, mailto, intent and map links.

Firstly, internal links point to resources within the same domain, either relative or absolute. Secondly, external links point to resources outside of the current domain. Thirdly, mailto links are prefixed with *mailto:* and are used to open the e-mail client from inside the browser and start a new draft e-mail. Fourth, intents are used on smart phones to start actions in smart phone applications and are prefixed with *intent://*.

Internal links mainly indicate the size of the website. External links have influence on the search engine score. Known as backlinks, these links help search engine operators estimate relevance and popularity of a certain group of pages. Although mainly of interest to the external party linked to, these links also help the search index of the linking party, if properly used and not abused.

We expect nearly all pages to predominantly have internal links and few external links. Furthermore, we expect many other websites to have a few external links. Apart from occasional links to the index pages of social media, we expect fraudulent webshops to not have any external links, mailto links, or intents due to the manual effort required for this.

I. CSS & JavaScript includes

JavaScript and CSS can be used for many wildly varying purposes. CSS is mostly used for stylistic purposes, such as default styling or applying hover effects on product images or buttons. Although JavaScript can also be utilised for stylistic purposes, such as transition effects, it is often utilised in a more functional way, such as form validation or integration with third party services, such as analytics or payment providers.

Moreover, most composable website or webshop frameworks extend core functionality by themes and plugins. Generally, these are offered in a standalone fashion and as such are not integrated with other source code files. Even if this option is available by means of optimisation, rendering problems can occur when dependencies are not met. Therefore, JavaScript and CSS source files are commonly not combined (or packed) when using webshop frameworks. This implies that the number of CSS and JavaScript source files increase for each theme or plugin used.

In the case of the fraudulent webshops examined, although we notice most webshops to utilise a composable open source framework, we noticed these implementations to be generic and only include a bare minimum of functionality, such as JavaScript files for offering image slider and lightbox functionality. Lightbox functionality implies the enlargement (zooming) of a picture and darkening of the background. This functionality is common in webshops to highlight products. To conclude, we expect fraudulent webshops to use a bare minimum of CSS and JavaScript, as some functionality are a necessity for webshops. Furthermore, we reason that a higher number of CSS and JavaScript implies that more customisation and manual labor was invested. This is less common at fraudulent webshops than general websites.

J. Lexical diversity

Lexical diversity is the amount of words (lemmas) used in web content divided by the total amount of unique words (lemmas). Considering fraudulent webshops offer little other content than a selection of products on display, and furthermore offer similar brands, names and sizes, we expected the lexical diversity to be lower than the average of regular websites.

K. Web analytics

Data on how websites are used, becomes increasingly important for research, business and market research. The process of collecting and interpreting this data is called web analytics. According to W3Techs, among Alexa's top 10 million websites, at least 64.9% use software to perform web analytics [17]. Wappalyzer, a utility that identifies known web technologies on the Internet, reports that at least 110,000 websites in the .nl TLD perform web analytics [18]. While these tools can give valuable insights in customer behaviour, it takes some manual effort in enabling these on a website. While analysing the known list of fraudulent webshops, we found that most of these webshops do not use any web analytics software. We presume that this is because setting up the software cannot easily be done automatically. On the other hand, the majority of the genuine webshops are developed with much more effort, and those operators see much more value in web analytics data. This makes it more likely that such a website has web analytics tools enabled, so the use of web analytics software could help distinct fraudulent webshops from genuine ones. We added this distinction as feature by using regular expressions derived from the Wappalyzer source code [19]. We chose to identify the top 20 marked leaders in web analytics revealed by Wappalyzer [18].

L. Deep links to social media

Many businesses use social media to promote products and to get in contact with customers. Most genuine webshops have buttons which link to their own page on various popular social media websites. We found that most fraudulent webshops have no real presence on social media websites. Some of them have social media buttons on their index page, though these link only to the default index page of the social media websites, or do not link to anything at all. Social media websites put great effort in validating registration and creation of business pages in their attempt to fight fake accounts and spam. For operators of fraudulent webshops, manually creating dedicated social media pages per webshop likely costs them too much effort. This would also give complaining customers more opportunity to spread the word that the webshop is fraudulent, which could result in faster domain revocations by the domain registrar. From this difference in social media links, we created two features for the classification. First, all the hyperlinks (*HTML* <*a*> tags) on the index page are extracted. From the hyperlinks which contain the word "facebook", "linkedin", "pinterest", "twitter", "vimeo" or "youtube", the *href* attribute is checked to see whether it is a deep link. This is a hyperlink which links to a specific webpage of a website, rather than the index page of that website. For example, a deep link of on the website "www.facebook.com" would be "www.facebook.com/nike/". The resulting features include whether a link *seems* to be linking to social media, and whether such link is a deep link.

VI. Results

A. Experiment I

This experiment embodies a 10-fold cross validation of the AdaBoost model trained on 3,300 *web* observations and 3,300 *nep* observations, totalling 6,600 observations. The *web* observations have been (pseudo) randomly sampled from all domains belonging to .nl. The *nep* observations constitute a combined set as explained in section III-B.

Statistics non-boolean features of all sets are shown in table V, Appendix A. Resulting metrics of this experiment are shown in table II. The model scores high on all metrics evaluated. Furthermore, further inspection and training other classification algorithms results in slightly lower but similar metric results. Therefore it is likely that these high metrics are accurate.

The weights of features for the classification algorithm is listed in table I. Interestingly, the analytics feature is the most

informative. Indeed, although many regular websites lack an integration with analytics providers, we have found this feature to accurately distinguish fraudulent webshops from legitimate webshops.

Also, the count of currency symbols is to be expected as a crude but accurate feature to distinguish webshops and non-webshops. Surprisingly, the edit-distance also seems to have a significant role in distinguishing the two classes.

Table II
EXPERIMENT I: AVERAGE METRICS

Average	AdaBoost
Accurancy	0.9934
Recall	0.9909
Precision	0.9941
F_1 score	0.9915

B. Experiment II

This experiment embodies classifying all available web pages from the .nl TLD. To this end, we have used the AdaBoost algorithm which we trained on the entire dataset (6,600 observations). The number of available web pages is around 4.9 million websites. This number is the result after filtering all domains without available page source. Of these, many domains forward to the same domain. After filtering these, we classified all remaining domains and pages, totalling around 4.3 million.

Of all 4,294,557 samples classified, the model estimates 32,815 samples to be positive, or *nep*; and 4,261,742 samples negative, or *web*. These numbers are shown in table III.

Table III
EXPERIMENT II: CLASSIFIED SAMPLES

Metric	Amount	Percentage
Total	4,294,557	100%
Negative	4,261,742	99.24%
Positive	32,815	0.76%

These numbers show a high inequality amongst predicted classes for different classifiers, which is to be expected. Considering the total set of non-fraudulent webshops is reasonably much larger than the total set of fraudulent webshops, the model seems to behave in an expected manner, despite being trained on an even number of class samples. However, these numbers alone don't convey any meaning about the performance of the model, namely the accuracy, recall and precision metrics. Since the set is unlabelled, manual evaluation of all classified samples would be necessary in order to measure the model's performance.

In order to approximate the precision, we have taken a (pseudo) random sample of 1,000 from the set of positively predicted domains of 32,815 observations. From these, we have found 301 true positives, which makes the precision 0.301.

Table IV shows the results when extrapolating this precision score to the total set of 32,815 positively predicted domains. It shows that on estimation a total of 9,877 fraudulent webshops are classified correctly.

Table IV
EXPERIMENT II: APPROXIMATED RESULTS BASED ON SAMPLES

Average	Approximation
True positive	9,877
False positive	22,937
True negative	unknown
False negative	unknown

VII. DISCUSSION

As demonstrated by the results of Experiment I and Experiment II, classification is a viable way in identifying fraudulent webshops. Experiment I achieved high performance metrics using 10-fold cross validation on the dataset and Experiment II is reasoned to have classified 9,877 fraudulent webshops. However, the precision metric between experiments decreased from 0.99 to 0.30. This indicates that the used *web* and *nep* sets during Experiment I are not fully representative of the *zone* set, or that insufficient features were used to appropriately distinguish both classes on a large scale. Furthermore, note that the *nep* and *web* sets are a subset of the *zone* set and were not excluded from the *zone* set during the training of the model in Experiment II.

It is well possible that the 3,300 fraudulent webshops on which AdaBoost was trained, are included in the 32,815 positively classified samples and in the 9,877 approximated true positives. However, this ultimately still implies that at least 6,000 new fraudulent webshops were found, thus still indicating the viability of the method.

Furthermore, one could object to the claim that the trained model classifies fraudulent webshops, but instead classifies webshops in general. Indeed, the set of *fraudulent webshops* is a subset of *all* the webshops, which subsequently is a subset of the .nl zone at large. Given this, one could argue that true positives found are a subset and a by-product of classifying webshops in general.

Let's consider the implications and assume a precision of 100% during Experiment II. This would imply that all 32,815 positively classified examples are webshops, and our sample approximates that 33%, or one in three, is a fraudulent webshop.

However, when inspecting the same sample from which the previous approximation follows, we also notice that around 33% of samples is not a webshop at all, but a general website. This would imply that the precision is not 100% but around 66%, from which follows that one in two webshops is fraudulent. When comparing this estimation to the report of the Consumentenbond, which estimates 90,000 total webshops and only one in five to be fraudulent; then it follows that the objection given, although reasonable, is likely false.

Furthermore, if the report from the Consumentenbond is true, then this implies that a total number of 18,000 fraudulent webshops exist. Given this, we can speculate and state that the recall achieved is around 50%.

On a final note, in deciding between webshop and non-webshop, we regard the distinctive factor to be the presence of a digital shopping cart and online payment capabilities. Websites advertising products without the possibility of immediate purchase are not recognised as webshops.

A. Future Work

Future work consists of two parts. First, we address several methods that can be utilised in order to improve the performance of the classification model. Afterwards, we provide general recommendations in order to address the problem of fraudulent webshops on the internet.

Many improvements to the current model can be made and features could be removed or extended. First, we suggest to experiment with training and testing using uneven classes. Although generally not recommended, in this case it may help improve precision, considering the ratio of fraudulent webshops to non-fraudulent webshops is very small. Second, we suggest building features to detect automatically translated text and text containing many grammatical mistakes. For instance, Aharoni, Koppel, and Goldberg showed machine translated text can be detected automatically [20]. Furthermore, mistakes may be identified by proofreading tools such as *LanguageTool* [21]. Third, we suggest evaluating which payment service provider is utilised. Payment service providers are generally well regulated and necessitate that their customers are real and legitimate companies. A common procedure amongst payment service providers is to manually verify a customer's website. Hence, this may be a good feature in separating fraudulent from legitimate webshops. Fourth, an in-depth study of different classification algorithms and configuration parameters could contribute to an even better performing algorithm. And at last, we suggest combining these features with other features based on the network, hosting or WHOIS data. The effectiveness of such a method is shown in nDEWS and in the research of Thijs Brands at SIDN labs. The combination of these could result in a reliable classification system. Furthermore, in terms of general recommendations, we would like to suggest an investigation into the options for resetting or degrading a domain's search engine score. For instance, returning a HTTP 404 is commonly regarded as a degrading measure towards some URI's search engine score. These measurements could be implemented by registries in order to discourage fraudulent webshop operators.

VIII. Conclusion

Operators of fraudulent webshop need to attract customers, maintain a high search engine score and achieve a high level of scalability in order to successfully exploit fraudulent webshops. Web content-based features can be modelled after these requirements. This research developed and tested 32 of such features. The three most informative features appear to be: The presence of web analytics software, the amount of currency symbols and the Levenshtein (edit) distance between the domain name and HTML title. This results in the identification of approximately 9,877 fraudulent webshops, of which at least 6,000 are new compared to the ones found in prior work. Although this method still shows false positives which can greatly be improved, fundamentally it shows the viability of content based classification of fraudulent webshops.

IX. Acknowledgements

This research was supported by SIDN. Their ongoing efforts into the monitoring and safekeeping of the .nl ccTLD have initiated and supported this research. Specifically, our gratitude goes out to Maarten Wullink and Marco Davids, who have put time and effort into supervising this work.

References

[1] VeriSign Inc., *The Domain Name Industry Brief*, https://www.verisign.com/assets/domain-name-report-Q42017.pdf, Visited on 2018-06-06.

[2] United Nations Conference on Trade and Development, *UNCTAD B2C E-COMMERCE INDEX*, http://unctad.org/en/PublicationsLibrary/tn_unctad_ict4d09_en.pdf, Visited on 2018-06-17.

[3] V. v. Amerongen, "De nepwebshop is overal", *DigitaalGids*, 2018.

[4] Nationale Politie, *Aangiften van internetoplichting gedaald*, https://www.politie.nl/nieuws/2018/februari/6/aangiften-van-internetoplichting-gedaald.html, Visited on 2018-07-07.

[5] D. Sahoo, C. Liu, and S. C. Hoi, "Malicious url detection using machine learning: A survey", *arXiv preprint arXiv:1701.07179*, 2017.

[6] Y.-T. Hou, Y. Chang, T. Chen, C.-S. Laih, and C.-M. Chen, "Malicious web content detection by machine learning", *Expert Systems with Applications*, vol. 37, no. 1, pp. 55–60, 2010.

[7] Giovane C. M. Moura, Moritz Muller, Maarten Wullink, and Cristian Hesselman, "nDEWS: a New Domains Early Warning System for TLDs", in *IEEE/IFIP International Workshop on Analytics for Network and Service Management (AnNet 2016), co-located with IEEE/IFIP Network Operations and Management Symposium (NOMS 2016)*, 2016.

[8] Dutch Open Hackathon, *Dutch Open Hackathon*, https://dutchopenhackathon.com/, Visited on 2018-06-30.

[9] Richard Garsthagen, *CrimeBusterBot*, https://github.com/AnykeyNL/CrimeBusterBot, Visited on 2018-06-30.

[10] De Consumentenbond, *Lijst onbetrouwbare webwinkels*, https://www.consumentenbond.nl/online-kopen/lijst-onbetrouwbare-webwinkels, Visited on 2018-06-20.

[11] Maarten Wullink, *Crawling .nl*, https://www.sidnlabs.nl/a/weblog/crawling-nl, Visited on 2018-06-06.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", *arXiv preprint arXiv:1301.3781*, 2013.

[13] N. Oostdijk, "Sonar-500 stevin nederlandstalig referentiecorpus (corpus geschreven nederlands, 500 miljoen woorden)", 2012.

[14] S. Tulkens, C. Emmery, and W. Daelemans, "Evaluating unsupervised dutch word embeddings as a linguistic resource", in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, N. C. C. Chair), K. Choukri, T. Declerck, M. Grobelnik, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds., Portorož, Slovenia: European Language Resources Association (ELRA), 2016, ISBN: 978-2-9517408-9-1.

[15] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora", English, in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, http://is.muni.cz/publication/884893/en, Valletta, Malta: ELRA, May 2010, pp. 45–50.

[16] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances", in *International Conference on Machine Learning*, 2015, pp. 957–966.

[17] Q-Success, *Usage of traffic analysis tools for websites*, https://w3techs.com/technologies/overview/traffic_analysis/all, Visited on 2018-06-27.

[18] wappalyzer.com, *Analytics*, https://www.wappalyzer.com/categories/analytics, Visited on 2018-06-27.

[19] Elbert Alias, *Wappalyzer*, https://github.com/AliasIO/Wappalyzer, Visited on 2018-06-27.

[20] R. Aharoni, M. Koppel, and Y. Goldberg, "Automatic detection of machine translated text and translation quality estimation", in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2014, pp. 289–295.

[21] LanguageTool, *languagetool-org/languagetool: Style and Grammar Checker for 25+ Languages*, https://github.com/languagetool-org/languagetool, Visited on 2018-07-10.

APPENDIX A
FEATURE STATISTICS

Table V
STATISTICS OF NON-BOOLEAN FEATURES

	NEP (3,300)				WEB (3,300)				ZONE (~4.29M)			
	mean	std	min	max	mean	std	min	max	mean	std	min	max
currency_count	39.73	18.75	0.00	217.00	1.75	14.93	0.00	692.00	1.34	13.87	0.00	4,784.00
distance_edit	62.5	18.8	11.0	194.0	31.23	23.79	0.00	356.00	32.27	689.31	0.00	402,506.00
distance_jaccard	0.67	0.11	0.20	1.00	0.59	0.21	0.00	1.00	0.62	0.23	0.00	1.00
image_count	24.89	11.98	0.00	191.00	8.47	19.53	0.00	360.00	7.15	24.55	0.00	4,962.00
lexical_count	624.07	1,129.92	14.00	31,848.00	1,764.14	7,497.32	0.00	210,175.00	1,745.86	8,215.22	0.00	1,191,077.00
lexical_diversity	2.55	6.70	1.12	227.49	18.21	78.95	0.00	824.01	19.16	88.75	0.00	13,334.00
lexical_unique	255.32	151.24	10.00	1,909.00	153.62	209.31	0.00	5,068.00	131.82	186.61	0.00	27,509.00
links_external	0.51	5.08	0.00	170.00	8.20	25.36	0.00	477.00	7.60	29.86	0.00	11,154.00
links_hash	0.07	0.52	0.00	12.00	1.40	8.07	0.00	219.00	0.99	8.69	0.00	2,391.00
links_intent	0.32	1.34	0.00	20.00	0.39	3.97	0.00	187.00	0.33	6.15	0.00	5,501.00
links_internal	154.47	263.86	0.00	4,443.00	21.76	59.26	0.00	1,153.00	15.47	58.71	0.00	9,767.00
links_mailto	0.00	0.00	0.00	0.00	0.17	0.75	0.00	11.00	0.11	0.89	0.00	532.00
links_map	0.00	0.00	0.00	0.00	0.00	0.04	0.00	2.00	0.00	0.01	0.00	4.00
dom_title_dist_sonar	1.18	0.15	0.00	1.35	0.93	0.44	0.00	1.36	0.85	0.49	0.00	1.45
dom_title_dist_wiki	1.28	0.16	0.00	1.45	1.01	0.48	0.00	1.46	0.93	0.54	0.00	1.55
dom_title_sim_sonar	0.52	0.12	0.00	0.96	0.43	0.25	0.00	1.00	0.40	0.27	-0.05	1.00
dom_title_sim_wiki	0.24	0.13	-0.10	0.82	0.27	0.25	-0.14	1.00	0.25	0.26	-0.24	1.00
metadesc_count	25.58	21.05	0.00	606.00	9.54	14.87	0.00	194.00	8.58	55.31	0.00	52,824
metakeyword_count	16.25	9.82	0.00	345.00	4.78	48.71	0.00	2,634.00	8.58	55.31	0.00	52,824
scripts_count	6.72	4.81	0.00	38.00	9.88	14.25	0.00	190	8.52	13.33	0.00	1,834.00
style_count	0.05	0.29	0.00	8.00	1.46	3.46	0.00	101.00	1.40	4.68	0.00	2,127.00