ATOL: A Framework for Automated Analysis and Categorization of the Darkweb Ecosystem

Shalini Ghosh Phillip Porras Vinod Yegneswaran Ken Nitz Ariyam Das
CSL, SRI International, Menlo Park
{shalini, porras, vinod, nitz}@csl.sri.com ariyam@cs.ucla.edu

Abstract

We present a framework for automated analysis and categorization of `.onion` websites in the *darkweb* to facilitate analyst situational awareness of new content that emerges from this dynamic landscape. Over the last two years, our team has developed a large-scale darkweb crawling infrastructure called `OnionCrawler` that acquires new onion domains on a daily basis, and crawls and indexes millions of pages from these new and previously known `.onion` sites. It stores this data into a research repository designed to help better understand Tor's hidden service ecosystem. The analysis component of our framework is called Automated Tool for Onion Labeling (`ATOL`), which introduces a two-stage thematic labeling strategy: (1) it learns descriptive and discriminative keywords for different categories, and (2) uses these terms to map onion site content to a set of thematic labels. We also present empirical results of `ATOL` and our ongoing experimentation with it, as we have gained experience applying it to the entirety of our darkweb repository, now over 70 million indexed pages. We find that `ATOL` can perform site-level thematic label assignment more accurately than keyword-based schemes developed by domain experts — we expand the analyst-provided keywords using an automatic keyword discovery algorithm, and get 12% gain in accuracy by using a machine learning classification model. We also show how `ATOL` can discover categories on previously unlabeled onions and discuss applications of `ATOL` in supporting various analyses and investigations of the darkweb.

1 Introduction

There is growing public awareness of Internet accessible *darkweb* sites, such as Silkroad or Wikileaks, where the illicit sales of drugs or dissemination of national secrets are hosted in an anonymous manner that makes it difficult for law enforcement to shut them down. These sites are hosted using the *Tor Hidden Service* (HS) protocol. Tor facilitates free anonymous online communication using an overlay network routing scheme called *onion routing*, which sends traffic through multiple relays to obfuscate the connection initiator's IP address. The HS-protocol extends Tor's IP-address obfuscation to enable network servers to protect their IP addresses as well. These hidden anonymous Internet services are referred to as *Onion sites* or simply *onions*.

The analysis presented in this paper arises from an ongoing research project, called LIGHTS, that seeks to provide a complete indexing of publicly known Tor Onion sites. Our ElasticSearch-based darkweb repository, grows daily, and currently indexes over 70 million pages from 32 thousand unique Tor Hidden Services reached since the commencement of this project. We have developed `OnionCrawler`, a fully automated crawling infrastructure to acquire new Tor onion domains (see Figure 1). Onion crawling, content indexing, and meta data generation are also fully automated, with the acknowledgment that substantial effort has been applied to derive critical meta-data to thematically label the content discovered within each harvested onion site. These labels are critical for navigating content, facilitating searches and content filtering, and for broadly understanding darkweb user communities and the ecosystem that is captured within the ocean of darkweb pages. It is this basic need to discover and assign thematically-descriptive labels and functional categorizations to the newly discovered onion sites that has motivated the development of the Automated Tool for Onion Labeling (`ATOL`) framework.

Another important area of application of thematic labels is bitcoin transaction analysis on the darkweb. Anonymous digital currencies such as Bitcoin are at the center of the darkweb economy. Bitcoin in particular is the current de facto digital currency used throughout the thriving dark markets (Soska and Christin 2015a), where illicit goods and services are sold. Bitcoin enables easy large-scale currency transactions, moving funds raised for entire movements or organized crime. It is a popular payment mechanism used by the hacker community to sell malicious tools, attack services, steal user data, and to extort payment (or ransom) from compromised victims. Indeed, in the last two years our darkweb crawling team has mined nearly 1.5 million unique Bitcoin addresses across approximately 34 thousand .onion sites and 70 million pages that we have crawled to date. The thematic labeling provided by `ATOL` will enable one to, e.g., distinguish a "drug transaction" from a "weapon transaction", and to provide categorical labels that enable investigators to quickly navigate the onion sites based on the contextual use of the digital currencies within the darkweb. If a bitcoin address occurs more often in drug-related onion pages than weapon-related ones, then through our statistical analysis tools we will determine and label the address as

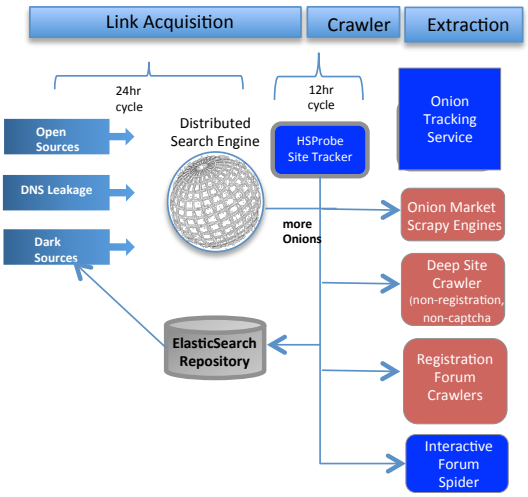

Figure 1: Overview of the onion acquisition and crawling infrastructure.

more likely to be drug-related rather than weapon-related.

Section 4 discusses the results of experiments we did with different variants of the ATOL algorithm. We find that ATOL can perform site-level thematic label assignment more accurately than keyword-based schemes developed by domain experts — we expand the analyst-provided keywords using an automatic keyword discovery algorithm, and get 12% gain in accuracy by using a machine learning classification model. We also discuss how a classifier trained using ATOL was able to outperform existing classifier significantly in a deployed system. Section 5 discusses example situational aware applications that are facilitated by our research.

2 Onion Crawler

We first provide a brief overview of an acquisition infrastructure that was constructed to discover new onion websites, crawl their content, and integrate them into our index repository. We refer collectively to these services as the LIGHTS Onion Crawler. We employ this system continually, twice per day to address diurnal patterns in onion site availability. Our sources of seed data include various published onion datasets((Branwen et al. 2016), (Biryukov, Pustogarov, and Weinmann 2013), (J. Nurmi 2016a), (HERMES Center for Transparency and Digital Human Rights 2016)), .onion references from a large collection of recursive DNS resolvers (Farsight Security, Inc. 2016), and an open repository of (non-onion) Web crawling data, called Common Crawl (Common Crawl Foundation 2016). Using these data sources as starting points, we developed tools to acquire additional onion addresses both from the onion Web and the open Web.

Specifically, we developed two tools, HSProbe (Tor Hidden Service Prober) and OnionCrawler, to check the operational status of onion sites and to crawl onion sites that are alive. HSProbe uses Tor's stem API (Tor Project 2016) for accessing onion sites over the Tor protocol and interpret a broad range of Tor Hidden Service-protocol status messages to determine how to proceed, as it encounters errors and unresponsive interactions with target hidden services. In addition, we developed the OnionCrawler tool that crawls onions. OnionCrawler also employs a Web search engine to find web pages whose contents contain onion addresses. After extracting onion addresses from the content of pages returned by the open Web search engine and from the onion sites, OnionCrawler iteratively queries the search engine using these onion addresses as search terms to learn new onion addresses. Finally, the collected data was parsed and indexed into ElasticSearch (Elastic 2016).

HSProbe is equipped with a port discovery function that can identify the virtual port used by hidden services that do not use the default TCP/80 port. Specifically, HSProbe is equipped with a configurable list of commonly used virtual ports used by non-botnet hidden services. Moreover, when Tor error codes suggest that a hidden server exists, but is not responding for the default port, HSProbe attempts to connect to these ports in turn until it successfully establishes a connection to a hidden service or all the ports are exhausted. This enables us to identify onion-deployed web services services operating configured on various non-standard ports.

Once an onion web service is detected, it is inserted into an onion site status tracking system that is utilized by our website crawling and indexing services, which we refer to as the *onionElasticBot* (J. Nurmi 2016b). onionElasticBot has modules that are specialized to do deep-crawling, marketplace crawling, as well as capability to monitor interactive forums. This service parses the webpages, perform data extraction, (e.g., titles, headers, and mail and Bitcoin addresses), and indexes the data using ElasticSearch (Elastic 2016). ElasticSearch provides a query API for the indexed data, that we used to generate some of the analysis describing in the forthcoming sections. Due to the complex legal and ethical considerations involved in crawling the Dark Web, our measurement study and resulting analysis was approved by our Institutional Review Board (IRB).

3 ATOL Analyzer

The onion network has some distinct characteristics for which we have developed a custom analysis platform called ATOL. ATOL can process the crawled onion sites in ElasticSearch and their underlying graph structure to do different types of analysis. In this paper, we consider one instantiation of such analysis — one of the key tasks that we perform using ATOL is categorizing onion sites in different ways, say according to their thematic labels (e.g., category), functional roles (e.g., hosted by seller), content type (e.g., blog). To this end, we can use both classification or clustering approaches — the former being more appropriate when we have a known set of categories, and the latter being relevant for new category discovery. For these approaches, it's important to characterize the categories — one way to do that is to represent a category using a set of relevant keywords. We first discuss a keyword-based classification approach in ATOL for thematic labeling.

One strong motivation for analyzing onion sites and label-

ing them thematically is to identify malicious onions. There are many malicious onion sites that are malicious clones of legitimates sites, operated by attackers with the presumed intent of executing malicious attacks on users, e.g., phishing login credentials, stealing bitcoin payments by rewriting bitcoin address — these sites pose a large threat to onion commerce sites and also undermine the trust in any onion services. There are other malicious activities on onion sites, e.g., illegal trade in weapons, drugs — it is also important to detect such sites to be able to track potentially criminal activity. To solve the latter problem, i.e., the problem of detecting crawled onion sites with illicit content, we first focus on reliably detecting onion categories. We then further analyze onion sites associated with sensitive categories (e.g., Weapons, Drugs, Hacker), to see whether the sites actually contain illicit content — related experiments and analyses are described in detail in Section 4.

3.1 ATOL: Thematic Labeling

We have built a prototype of ATOL over OnionCrawler, using which we did experiments on thematic categorization of onions. The thematic categorization of onions in ATOL uses a 2-stage algorithm:

1. Expanding a given list of keywords associated with a category, and
2. Using the category keywords to train a classifier for categorizing onions into thematic labels.

Algorithms 1 and 2 give an outline of the overall 2-stage algorithm for thematic labeling using ATOL. The following 2 sections, Section 3.2 and 3.3, outline the details of these 2 stages of Thematic Labeling using ATOL.

3.2 ATOL: TFICF Weighting

For each onion category/theme, domain experts (analysts) initially provided a manually-curated set of keywords, e.g., the "Weapons" category has keywords like gun, glock, silencer, caliber, etc. The goal of ATOL in this case is to automatically discover relevant keywords, using data from multiple sources, e.g., title and content words from onion text as well as the existing manually-tuned keywords.

Note that such list of keywords can also be automatically extracted from onions whose content and category label are known, by doing natural language processing (NLP). We show how both the approaches — using manually-curated keywords or completely automated keywords — can be outperformed by the ATOL approach that combines these two techniques. In ATOL, we start with a seed list of keywords per category and use a bootstrapping mechanism to augment the seed list with other relevant keywords for those categories. Our experiments and analysis in Section 4 shows how our bootstrapping approach gives the best empirical result. Algorithm 1 shows the TFICF-based keyword-discovery algorithm used in ATOL for that purpose. ATOL finds the keywords with the highest Term Frequency Inverse Class Frequency (TFICF) weights for a given category, where TFICF is defined as the product of TF and ICF scores. We define TFICF as follows:

Algorithm 1: TFICF-based Keyword Weighting

function `tficfFeature` $(K_{expert}, C_{train}, L_{train}, \lambda)$

Input : $K_{expert} \leftarrow$ Seed list of keywords from domain expert; $C_{train} \leftarrow$ content of onion sites where each onion d is represented as a n-dimensional *bag-of-words* vector $X \in \mathbb{R}^n$, n being the vocabulary size; $L_{train} \leftarrow$ set of class labels assigned to the corpus based on rater labelings; $\lambda \leftarrow$ weight multiplier on title words.

Output: $M \leftarrow$ Matrix where each row is a weighted vector of keywords per category, with weight = TFICF score of keyword in category.

```
// Populate category x keyword matrix M
M = []
for k ∈ K_expert do
    if existing keyword k is in category c then
        M[c, k] += categoryCount(k, c) // count of keyword in category
    end
    if onion d has labels c_1 ... c_m in training data then
        for category c ∈ {c_1 ... c_m} do
            for word w ∈ d do
                M[c, w] += onionCount(w, d)/m // count of word in onion, where m is the number of labels for onion d
            end
            for word t ∈ title of d do
                M[c, t] += λ × titleCount(t, d)/m // count of word in onion title
            end
        end
    end
end
// Compute TFICF scores per category, using M
for each word w in col(M) do
    ICF_w = 0
    for each category c in row(M) do
        if M[c, w] > 0 then
            ICF_w += 1
        end
    end
    if ICF_w > 0 then
        ICF_w = log(|C| / ICF_w) // |C| = number of categories
    end
    for each category c in row(M) do
        M[w, c] = M[w, c] × ICF_w // compute TF x ICF
    end
end
return M
```

$$
\begin{aligned}
tficf(w, c, C) &= tf(w, c) \times icf(w, C), \text{where} \\
tf(w, c) &= freq(w, c), \text{and} \\
icf(w, C) &= \log \frac{|C|}{|c \in C : w \in c|}
\end{aligned}
$$

where w is a keyword, c is a category, C is the set of all categories, and $freq(w, c)$ counts the number of times w occurs across all onions assigned to category c.

Intuitively, for a given keyword and category, TF (Term Frequency) measures the popularity of the keyword in that category, while ICF (Inverse Class Frequency) estimates the rarity of they keyword across all categories — so, the product TFICF gives a high weight to keywords that are common within a category, but not common in other categories. This helps to identify keywords that are more unique to a category, and hence better representatives of the category. Note that the TFICF score is a variant of the TFIDF score that is used extensively in information retrieval (Manning and Schütze 1999), where we have defined (and used) the ICF score to compute the popularity of a word across categories, instead of using the IDF score used in TFIDF to compute the popularity of a word across documents.

One of the key aspects of the TFICF score is how the TF score is computed using data from multiple sources — Algorithm 1 outlines that in the steps that populate the matrix M, which is used to compute the final TFICF score.

3.3 ATOL: Classifier

Using the keywords inferred by the TFICF algorithm, ATOL trains a classifier to predict the category of an onion. Different classifiers can be used in this Stage (2) of the ATOL framework — in our experiments we trained SVM, Naive Bayes, and Logistic Regression classifiers, using different kinds of feature weighting schemes (e.g., BOW, TFIDF, TFICF) to represent the training/test data points.

Algorithm 2 gives an outline of the classification stage of the ATOL algorithm. We compared how the performance of the thematic category prediction stage of ATOL changed with different classifiers, as well as different keyword weighting schemes, i.e., whether using TFICF weights gave improvements over the keywords manually curated by the analysts — details of our ablation experiments are outlined in Section 4.

3.4 Semi-supervised Classifier

As discussed in the previous sections, automated supervised thematic classification achieves high accuracy. However, getting labeled training data becomes acute in the context of darkweb ecosystem, since even manual labelling of onion sites would require deep domain expertise. On the other hand, crawling onion sites is mostly inexpensive and does not require much human intervention — so, we can easily collect significant amount of unlabeled data. This motivates us to explore certain semi-supervised approaches where we attempt to learn from labelled as well as unlabeled examples. Semi-supervised learning methods can be divided into many categories. In this work, we mainly examined the graph-

Algorithm 2: ATOL Classifier

function `atolClassify` $(C_{train}, C_{test}, C_{unlabeled}, T)$

Input : $C_{train}, C_{test} \leftarrow$ Corpus of training and test documents, where each document d is represented by a n-dimensional *bag-of-words* vector $X \in \mathbb{R}^n$, n being the vocabulary size and l is the class label assigned to d; $C_{unlabeled} \leftarrow$ Corpus of unlabeled documents, for which ATOL will try to discover categories; $T \leftarrow$ threshold for category discovery.

Output: $ML \leftarrow$ ML classifier trained using C_{train}; $L_{test} \leftarrow$ Labels assigned by ML for every onion $d \in C_{test}$; *accuracy* $\leftarrow$ onion classification accuracy in C_{test}; $L_{discover} \leftarrow$ Labels discovered by ML on subset of onions in $C_{unlabeled}$.

Train: $D_{train} \leftarrow \emptyset$
for $d \in C_{train}$ **do**
  $X := \mathbb{R}^n$ *bag-of-words* vector for d
  $l :=$ class label for d
  $D_{train} \leftarrow D_{train} \bigcup$ `tficfFeature`$(X, D, l, 1)$
  `// change` X `from bow to tficf`
end
Fit a classifier *ML* on D_{train}.

Evaluate: $L_{test} \leftarrow \emptyset$
correct := 0
for $d \in C_{test}$ **do**
  $X := \mathbb{R}^n$ *bag-of-words* vector for d
  $l :=$ class label for d
  $l' :=$ `predict`(*ML*, X) `// predict label for onion`
  $L_{test} \leftarrow L_{test} \bigcup l'$
  if l *matches* l' **then**
    correct := *correct*+1 `// correctly classified`
  end
end
accuracy $\leftarrow \frac{correct}{|C_{test}|}$ `// compute accuracy`

Discover: $L_{discover} \leftarrow \emptyset$
for $d \in C_{unlabeled}$ **do**
  $X := \mathbb{R}^n$ *bag-of-words* vector for d
  $l :=$ `predict`(*ML*, X) `// predict label for onion`
  prob := `prob`(*ML*, X, l) `// probability of onion label`
  if $prob > T$ **then**
    $L_{discover} \leftarrow L_{discover} \bigcup l$
  end
end
return $(ML, L_{test}, accuracy, L_{discover})$

based approaches and in particular used the label propagation algorithm (Zhu and Ghahramani 2002) for our experiments.

4 Experimental Results

We ran experiments on the `OnionCrawler` output, to evaluate the effectiveness of `ATOL` Thematic Labeling and variations. This section describes the different experiments and related analyses.

4.1 Methodology

For the experiments, we considered a dataset sampled from the `OnionCrawler` snapshot of February 19th, 2016. Analysts annotated a sample of 529 onion *sites* with 3 labels — Weapons, Drugs or Hacker. Table 1 shows the details of the labeled dataset. Note that the labels were provided at the site-level — each site had multiple associated pages, and the label was provided for the dominant category related to the content of those pages. We ran experiments with 5-fold cross-validation and stratified sampling, such that each fold approximately has 1/5-th of the labeled dataset, and also approximately 1/5-th of each category label.

Dominant Category	Number of examples
Drugs	178
Hacker	268
Weapons	83

Table 1: 529 onion sites marked with categories.

We use 5-fold cross validation for evaluating the accuracy of the experiments. In each cross-validation run, we use $fold_i$ as test set and the remaining folds as training set (for i = 1 to 5) — this enables us to get 95% confidence intervals of our accuracy results in Section 4.2.

For the other results, e.g., keyword discovery in Section 4.3 or reduction of analysis burden in Section 4.4, we use the training/test split outlined in Table 2. This split was provided by the analysts, based on their annotation of the labeled data. Note that this split has a skew in distribution of categories between train and test datasets, so it was not used for accuracy calculation — however, it's ok to use for other tasks, e.g., keyword discovery or estimation of reduction of analysis burden.

Dominant Category	# Train data	# Test data
Drugs	158	20
Hacker	248	20
Weapons	45	38

Table 2: Train/test split with dominant categories.

4.2 Accuracy

The performance of the algorithms on the test sets using 5-fold cross validation, with different feature weighting schemes, are shown in Table 3.

As outlined in Section 3, we have 2 phases of the `ATOL` Thematic Labeling algorithm — the keyword generation, and the classifier. As shown in the table, we compare 4 methods of keyword generation:

Features	Classifier	5-fold Accuracy
BOW	Multinomial Naive Bayes Classification	0.802 ± 0.038
	Linear SVM (Stochastic Gradient Descent)	0.822 ± 0.069
	Logistic Regression	0.771 ± 0.099
TFIDF	Multinomial Naive Bayes	0.857 ± 0.072
	Linear SVM (Stochastic Gradient Descent)	0.853 ± 0.083
	Logistic Regression	0.819 ± 0.077
TFICF (ATOL)	Multinomial Naive Bayes	$\mathbf{0.964 \pm 0.029}$
	Linear SVM (Stochastic Gradient Descent)	0.942 ± 0.060
	Logistic Regression	0.918 ± 0.074
	CosineSim + Softmax	0.884 ± 0.047
Baseline (Analyst)	CosineSim + Softmax	0.858 ± 0.044

Table 3: Test-set performance of algorithms on dominant category prediction.

1. Baseline: The list of keywords provided by the analyst, based on their domain expertise.
2. BOW: Keywords obtained by simple tokenization of the onion documents related to a category label, and then considering the bag-of-words vector of the words in a category as the relevant keywords.
3. TFIDF: Considers BOW representation, but additionally applies the TFIDF algorithm (Manning and Schütze 1999) to give feature weights to the words.
4. TFICF: Applies the feature weighting scheme outlined in Algorithm 1 to the BOW representation, to get a set of keywords with associated weights.

Let us analyze the results of Table 3 in more detail. Using the analyst-provided keywords in `Baseline`, we compute the cosine similarity of a new onion with the keyword vector for a category, followed by softmax transform, to estimate the probability of the onion belonging to the category probabilities — we use this to predict the most probable category for an onion. This gives an accuracy of 0.858 ± 0.044.

We use different classifiers from SciKit-Learn [1]:

1. Multinomial Naive Bayes Classification: Naive Bayes Classifier (NBC) that uses the multinomial distribution on discrete features.
2. Linear SVM (Stochastic Gradient Descent): Linear SVM classifier that is trained using Stochastic Gradient Descent (SGD) learning, using hinge loss and L2 regularizer.
3. Logistic Regression: Logistic regression classifier that uses L2 regularizer, using a Stochastic Average Gradient (SAG) descent solver.

When we trained these classifiers on the Bag of Words (BOW) and TFIDF weighted keywords, the results were comparable to the Baseline performance — in some cases we got better results on average accuracy, but the confidence intervals overlapped. However, when we used these classifiers along with the TFICF weighting, the Multinomial NBC classifier gave an accuracy of 0.964 ± 0.029, which was

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

Word	Explanation
scam	Strong indicator for hacker topic
mitgliedjoined	German for "member joined"
patternjuggled	github.com/pjstorm – hosts crypto software
phpcredlocker	Secure repository for credentials
dekryptering	Swedish for encryption
moneymail	Money maker website
altergold	Online payment gateway
cryptostormteam	Team of cryptostorm
cryptohavennet	pure.cryptohaven.net - security darknet team
darkwebscience	Strong indicator for hacker topic

Table 4: Top 10 keywords discovered by ATOL in "Hacker" category.

Word	Explanation
smoketime	Tobacco shop
clenotabs	Tabs of Clenbuterol
pharmachem	Strong indicator of drugs
sustamed	Oil-based testosterone
oxandrolonecentrino	OXANDROLONE from centrinolab
testosterone	Steroid
lsdxtal	Lsd from xtal
neurogroove	Polish drug website
cocaine	Drug
testobolin	Hormone

Table 5: Top 10 keywords discovered by ATOL in "Drugs" category.

a statistically significant improvement over Baseline (non-overlapping confidence intervals). Comparing the average accuracy values, we get a 12% improvement with MultinomialNBC + TFICF compared to Softmax + Baseline, showing the efficacy of the ATOL algorithm in giving us high-accuracy thematic labeling classifiers.

4.3 Keyword Discovery

Table 4, 5 and 6 shows the top 10 keywords (sorted by TFICF) that were found by the keyword discovery algorithm in ATOL for the Hacker, Drugs and Weapons categories respectively, when run on the analyst-provided train/test split — the tables also have explanations of why the discovered keywords are relevant for the corresponding categories.

Word	Explanation
waltherppk180x180jpg	Walther ppk 180 gun
stungun180x180jpg	Stun gun
Guns	Indicator of weapons
police	Indicator of weapons
selfcocking	Type of gun
sigsauer180x180jpg	Sigsauer gun
stratgietactique	Tactical strategy
pt99af	Taurus pt99af gun
darkdontay	Writes guide to making ninja weapons
flashball lbd	French ball launcher defense weapon

Table 6: Top 10 keywords discovered by ATOL in "Weapons" category.

4.4 Reduce Analysis Burden

We demonstrate how ATOL can reduce the burden on the human analyst. By selecting onions where ATOL predicts a category with probability > 0.9, the candidate list of onions that an analyst needs to analyze per category was pruned substantially. Table 7 shows the reduction of the Baseline and ATOL algorithms on a dataset with 26072 onion sites for the "Drugs" category — the reduction is 94%. So, if an analyst wants to look at onions marked as "Drugs" with > 0.9 probability, the analyst has to look at only 6% of the original set of 889 candidates. Similarly, for the "Hacker" category, ATOL reduces the set by 97% and for "Weapons" by 92% (with probability > 0.9). So, ATOL is able to reduce the analyst's burden substantially, since it helps to focus the analyst's attention on the cases where the algorithm is most certain of the categorization.

Algorithm	Labeled as Drugs
Baseline	889
ATOL	61

Table 7: Reduction of analysis burden on dataset with 26072 onion sites.

4.5 Theme Discovery

The goal in theme discovery is to find unlabeled onions that are most likely to belong to a particular theme or category. For the theme discovery experiments, we considered a data snapshot from OnionCrawler from March 3rd, 2016 that had 19,342 onions. We focused on onions marked as "Weapons" by ATOL but not by the Baseline algorithm — considering ATOL probability of "Weapons" to be > 0.5 gave us 32 onions (of which analysis revealed that 11 were actually weapons-related), while probability > 0.7 gave us 15 onions (of which 6 were actually weapons-related). Table 8 gives examples of the 11 onion sites that were discovered by ATOL to be weapons-related at probability threshold of 0.5. So, ATOL was able to discover new onion sites in existing categories, which did not have any category labels in the original dataset.

4.6 Semi-supervised Classifier Results

We performed some initial experiments using the semi-supervised classification results. In some of the classifiers, e.g., Logistic Regression, we get gains in the semi-supervised classification w.r.t. supervised classification. We want to tune the parameters of the semi-supervised algorithms in future work, to be able to get gains on all of the supervised algorithms considered in our experiments.

5 Applications of ATOL

There are several applications of ATOL that arise in the realm of large-scale darkweb index management and the automated contextual analysis of content or persona attributes that appear in the darkweb.

1) Automated portal generation: Automated categorization of sites that are dynamically discovered through our crawling infrastructure is a core challenge addressed by ATOL.

Onion	Sample Words
nethack3dzllmbmo	weapons, dagger, armor, fighting, killing
armoryx7kvdq3jds	pistols, revolvers, hunting, rifles, shotguns, assault, defense
c6q2m57ts2crvtiz	gunsite, pistol, clubs, shooter, gamesmanship, range, shoot, colt
6xbcodgrkz3tffpv	armory, inventory, barrel, military, trigger, mag, ruger
sqmbat5xti4jhzx4	military, ammunition, bombs, missiles, atomic
armor64oojvty6ob	armory, pistols, compact, subcompact, revolvers, shotguns, weapons
pyro5l7wciwhv2sa	mines, bombs, shells, rockets, mortars, explosives, nitroglycerin
rsci7rl3rmpcsvyf	explosives, weapons, nitric, acid, demolition, firearm, gunsmithing
f2onmpf722dtn7hs	rockets, shells, canister, mines, bombs, explosives, nitrocellulose
truth77k52rbo3ov	warfare, army, firing, shot, nuclear, weapons
tp7qimqtpdxl44gq	submachine, gunpowder, boltaction, ammunition, weapons, gunsmithing

Table 8: Sample of new onions (previously unlabeled) discovered as weapons-related by running ATOL inference.

Site-level thematic categorization facilitate topic-driven investigations, site filtering, and the ability to track changes in specific topic domains. We are continuing to extend our automated portal generation with new classifiers in addition to thematic classifiers: (a) functional classifiers, which classify onion sites by function (e.g., hosted by a seller or not), (b) content type classifier, which classify whether the site is a blog, wiki, forum, etc., (c) sentiment classifier, which classifies whether a site has positive or negative sentiment about the topic it discusses.

2) Topic-driven extractions of various search terms and persona attributes: A second use case for thematic categorization is the common investigation challenge of identifying the set of attributes that all fall within the same topic model. For example, identifying persona attributes, such as email or instant message handles that are associated with *weapon sales* is a highly useful applications. So too is the search for posted bitcoin addresses that are involved in drug sales or other illicit topics.

3) Contextual meta-data generation for target persona attributes: An inverse application to the prior bullet is the use of ATOL to drive the generation of contextual metadata that can be associated with a given persona attribute, such as an email or bitcoin address, a PGP key, or a persona name. The persona attribute may appear on many pages across multiple onion sites. We are exploring a scoring function that enables one to identify the thematic labels that dominate the pages on which the persona attribute appears. For example, given a bitcoin address, the scoring could identify the most dominant thematic labels (e.g., drug sales) that capture the context regarding where this bitcoin address predominantly appears.

6 Related Work

There have been a few prior measurement studies of content present in the onion ecosystem. These include measurement studies and analysis of the dynamics of onion drug marketplaces (Christin 2013; Soska and Christin 2015b), as well as studies that have exploited flaws in Tor's hidden service design of onion domains (Biryukov, Pustogarov, and Weinmann 2013; Biryukov et al. 2014; Owen and Savage 2016), to reveal private .onion domains, including botnet C&Cs. Systems like DeepDive (Niu et al. 2012), which have been used to analyze content of the darkweb, need the crawled content to be available before doing any analysis. Christin et al., conducted a comprehensive analysis of the sellers of SilkRoad marketplace (Christin 2013). Soska et al., follow up by conducting a longer term measurement study of vendor activity across marketplaces. Biryuokov et al., exploited flaws in Tor's hidden service protocol to measure the popularity of onion services and deanonymize them. They follow up with an analysis of hidden service content (Biryukov et al. 2014) from 3050 HTTP services, finding that the most popular services are from botnets.

Unlike these prior efforts, we do not rely on HSDir harvesting (i.e., setting up HSDir relay nodes for the purpose of harvesting onion addresses). Instead we rely on strategies, such as open as dark web crawling as well as DNS traces to acquire onion addresses, that conform to Tor's ethical research guidleines (Tor Project 2015). Hence our results on popular content are also different. Furthermore, to the best of our knowledge, our approach of combining OnionCrawler and ATOL is the first attempt to develop a principled framework for crawling onion sites and classifying their extracted content with relevant labels.

Machine learning (ML) reseach in cyber-security has focused on different applications of ML to the openweb, e.g., modeling threat propagation for detecting malicious activities (Carter, Idika, and Streilein 2013), adaptive trust modeling for cyber security (Robertson and Laddaga 2012), game-theoretic modeling of cyber security threats like information leakage (Xu et al. 2015), adaptive attacker strategy evolution (Winterrose et al. 2014), privacy-preserving data analysis (Foulds et al. 2016), attacks on ML classifiers (Burago and Lowd 2015), or detecting user authenticity and spammy names in social networks (Freeman et al. 2016; Xiao, Freeman, and Hwa 2015; Freeman 2013). However, not a lot of work has been done on analyzing the darkweb. Sabbah et al. (Sabbah et al. 2016) have proposed a keyword-weighting and classification scheme for dark web classification — however they focus on combining different feature weighting schemes for a binary classification task of detecting terrorist pages. In contrast, the TFICF-based keyword weighting scheme can use prior keyword distributions effectively.

7 Conclusion

Automated site and page-level classification labels provide interesting meta-data for indexing and understand communities of anonymous personas. For example, consider the integration of thematic and site-function attributes as features

to associate with persona-specific attributes such as persona names, PGP keys, anonymous email addresses, bitcoins, and references to social media handles. The appearance of these persona attributes on site with a consistent thematic label could provide a useful insight for census tracking the size and growth rates of the user community associated with that label. Thematic meta-data could also provide novel search indexing services for tracking user communities or in understanding usage patterns from a specific anonymous persona. For example, to date we have isolated over 1.5M unique bitcoin addresses that have appeared on Darkweb pages. Thematic and functional site labels could offer search indexing that allows one to isolate, for example, all bitcoins involved in various illicit marketplaces (e.g., drug dealing, weapons sales, counterfeiting, identity theft, or hacking services).

This paper presented an automated system for crawling (`OnionCrawler`) and analyzing (`ATOL`) content in the public Tor hidden service ecosystem. During the last two years, our system has automatically crawled and classified millions of pages in the darkweb. We have developed a data analysis framework to derive thematic labels to analyze the content of onions crawled within our repository. Our preliminary results indicates that `ATOL` outperforms a keyword-based baseline algorithm used by analysts by 12%, using a novel keyword weighting scheme called TFICF used in conjunction with supervised machine learning classification algorithms. We also develop a novel semi-supervised learning framework, which shows promising initial results. Finally, we outline a series of important problems in this space, related to both `OnionCrawler` and `ATOL`, such as keyword enrichment, multi-classifier analysis, theme learning, graph analysis and thematic census mining, which would be useful to explore in the future.

Acknowledgments

The authors would like to thank Dr. Patrick Lincoln for his valuable feedback regarding this work, and Dr. Steven Cheung for his help developing the onion crawler. This project was partially funded by the National Science Foundation (NSF) under Award Number CNS-1314956 and the Defense Advanced Research Projects Agency (DARPA) under Grant No. FA8750-14-C-0237. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF or DARPA.

References

[Biryukov et al. 2014] Biryukov, A.; Pustogarov, I.; Thill, F.; and Weinmann, R. P. 2014. Content and popularity analysis of tor hidden services. In *ICDCSW*.

[Biryukov, Pustogarov, and Weinmann 2013] Biryukov, A.; Pustogarov, I.; and Weinmann, R.-P. 2013. Trawling for tor hidden services: Detection, measurement, deanonymization. In *IEEE-SP*.

[Branwen et al. 2016] Branwen, G.; Christin, N.; Décary-Hétu, D.; Andersen, R. M.; StExo; Presidente, E.; Anonymous; Lau, D.; Sohhlz; Kratunov, D.; Cakic, V.; and Buskirk, V. 2016. Dark Net Market Archives 2011-2015. `www.gwern.net/Black-market%20archives`. Last accessed: May 10, 2016.

[Burago and Lowd 2015] Burago, I., and Lowd, D. 2015. Automated attacks on compression-based classifiers. In *AISec*.

[Carter, Idika, and Streilein 2013] Carter, K. M.; Idika, N. C.; and Streilein, W. W. 2013. Probabilistic threat propagation for malicious activity detection. In *ICASSP*.

[Christin 2013] Christin, N. 2013. Traveling the silk road: A measurement analysis of a large anonymous online marketplace. In *WWW*.

[Common Crawl Foundation 2016] Common Crawl Foundation. 2016. Common Crawl. `http://commoncrawl.org`.

[Dalvi et al. 2004] Dalvi, N.; Domingos, P.; Mausam; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *KDD*.

[Elastic 2016] Elastic. 2016. Elasticsearch. `https://www.elastic.co/`.

[Farsight Security, Inc. 2016] Farsight Security, Inc. 2016. SIE: The Security Information Exchange. `https://www.farsightsecurity.com/SIE/`.

[Foulds et al. 2016] Foulds, J. R.; Geumlek, J.; Welling, M.; and Chaudhuri, K. 2016. On the theory and practice of privacy-preserving bayesian data analysis. *CoRR* abs/1603.07294.

[Freeman et al. 2016] Freeman, D.; Jain, S.; Dürmuth, M.; Biggio, B.; and Giacinto, G. 2016. Who are you? A statistical approach to measuring user authenticity. In *NDSS*.

[Freeman 2013] Freeman, D. M. 2013. Using naive bayes to detect spammy names in social networks. In *AISec*.

[HERMES Center for Transparency and Digital Human Rights 2016] HERMES Center for Transparency and Digital Human Rights. 2016. Tor2web: Browse the Tor Onion Services. `https://tor2web.org/`.

[J. Nurmi 2016a] J. Nurmi. 2016a. Ahmia Search Engine. `https://ahmia.fi/`.

[J. Nurmi 2016b] J. Nurmi. 2016b. onionElasticBot - Crawl .onion and .i2p websites from the Tor network. `https://github.com/ahmia/search/tree/ahmia-redesign/onionElasticBot`.

[Lowd and Meek 2005] Lowd, D., and Meek, C. 2005. Adversarial learning. In *KDD*.

[Manning and Schütze 1999] Manning, C. D., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

[Niu et al. 2012] Niu, F.; Zhang, C.; Re, C.; and Shavlik, J. W. 2012. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. In *VLDS*.

[Owen and Savage 2016] Owen, G., and Savage, N. 2016. Empirical analysis of tor hidden services. *IET Info. Sec.* 10.

[Robertson and Laddaga 2012] Robertson, P., and Laddaga, R. 2012. Adaptive security and trust. In *SASOW*.

[Sabbah et al. 2016] Sabbah, T.; Selamat, A.; Selamat, M. H.; Ibrahim, R.; and Fujita, H. 2016. Hybridized term-weighting method for dark web classification. *Neurocomputing* 173(3).

[Soska and Christin 2015a] Soska, K., and Christin, N. 2015a. Measuring the longitudinal evolution of the online anonymous marketplace. In *USENIX*.

[Soska and Christin 2015b] Soska, K., and Christin, N. 2015b. Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In *USENIX*.

[Tor Project 2015] Tor Project. 2015. Ethical Tor Research: Guidelines. `https://blog.torproject.org/blog/ethical-tor-research-guidelines`.

[Tor Project 2016] Tor Project. 2016. Stem. `https://stem.torproject.org/`.

[Winterrose et al. 2014] Winterrose, M. L.; Carter, K. M.; Wagner, N.; and Streilein, W. W. 2014. Adaptive attacker strategy development against moving target cyber defenses. *CoRR* abs/1407.8540.

[Xiao, Freeman, and Hwa 2015] Xiao, C.; Freeman, D. M.; and Hwa, T. 2015. Detecting clusters of fake accounts in online social networks. In *AISec*.

[Xu et al. 2015] Xu, H.; Jiang, A. X.; Sinha, A.; Rabinovich, Z.; Dughmi, S.; and Tambe, M. 2015. Security games with information leakage: Modeling and computation. In *IJCAI*.

[Zhu and Ghahramani 2002] Zhu, X., and Ghahramani, Z. 2002. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, CMU.