Automated Categorization of Onion Sites for Analyzing the Darkweb Ecosystem

Shalini Ghosh
Computer Science Laboratory
SRI International
shalini@csl.sri.com

Ariyam Das
Department of Computer Science
University of California, Los Angeles
ariyam@cs.ucla.edu

Phil Porras
Computer Science Laboratory
SRI International
porras@csl.sri.com

Vinod Yegneswaran
Computer Science Laboratory
SRI International
vinod@csl.sri.com

Ashish Gehani
Computer Science Laboratory
SRI International
gehani@csl.sri.com

ABSTRACT

Onion sites on the *darkweb* operate using the Tor Hidden Service (HS) protocol to shield their locations on the Internet, which (among other features) enables these sites to host malicious and illegal content while being resistant to legal action and seizure. Identifying and monitoring such illicit sites in the darkweb is of high relevance to the Computer Security and Law Enforcement communities. We have developed an automated infrastructure that crawls and indexes content from onion sites into a large-scale data repository, called LIGHTS, with over 100M pages. In this paper we describe *Automated Tool for Onion Labeling* (`ATOL`), a novel scalable analysis service developed to conduct a thematic assessment of the content of onion sites in the LIGHTS repository. `ATOL` has three core components — (a) a novel keyword discovery mechanism (`ATOLKeyword`) which extends analyst-provided keywords for different categories by suggesting new descriptive and discriminative keywords that are relevant for the categories; (b) a classification framework (`ATOLClassify`) that uses the discovered keywords to map onion site content to a set of categories when sufficient labeled data is available; (c) a clustering framework (`ATOLCluster`) that can leverage information from multiple external heterogeneous knowledge sources, ranging from domain expertise to Bitcoin transaction data, to categorize onion content in the absence of sufficient supervised data. The paper presents empirical results of `ATOL` on onion datasets derived from the LIGHTS repository, and additionally benchmarks `ATOL`'s algorithms on the publicly available 20 Newsgroups dataset to demonstrate the reproducibility of its results. On the LIGHTS dataset, `ATOLClassify` gives a 12% performance gain over an analyst-provided baseline, while `ATOLCluster` gives a 7% improvement over state-of-the-art semi-supervised clustering algorithms. We also discuss how `ATOL` has been deployed and externally evaluated, as part of the LIGHTS system.

KEYWORDS

keyword discovery, classification, clustering, darkweb, onion sites

KDD'17, Halifax, Nova Scotia, Canada
© 2017 Copyright held by the owner/author(s). 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

ACM Reference format:
Shalini Ghosh, Ariyam Das, Phil Porras, Vinod Yegneswaran, and Ashish Gehani. 2017. Automated Categorization of Onion Sites for Analyzing the Darkweb Ecosystem. In *Proceedings of ACM SIGKDD conference, Halifax, Nova Scotia, Canada, August 2017 (KDD'17),* 9 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Tor's HS protocol [4] allows web services to remain hidden by obfuscating the IP addresses of the network servers, through multiple relays within Tor's overlay network, using a network routing scheme called *onion routing*. These hidden anonymous services use the `.onion` special top-level domain (TLD) and are, hence, often referred to as *onion sites* or simply *onions*. Some recent examples of these include drugs and weapons marketplaces (Silk Road, Armory) carrying out illegal trade, hacker forums (OnionWarez) publishing details of identity theft victims, terrorist forums [3] attracting bulk donations, whistleblower sites (WikiLeaks), and fraudulent financial sites (EasyCoin, OnionWallet) running monetary scams. These websites are hosted in an anonymous manner, making it difficult for law enforcement to shut them down. With the number of onion sites growing at an alarming rate (nearly doubled in the last year [31]), cyber and national security experts are increasingly investing in data mining tools that automatically identify suspicious activities on the darkweb [3].

Our ElasticSearch-based LIGHTS darkweb repository grows daily and currently indexes over 100 million pages from over 43 thousand unique Tor Hidden Services reached since the commencement of this project.[1] Onion crawling, content indexing, and metadata generation are also fully automated by our LIGHTS acquisition system, and substantial effort has been applied to derive critical metadata 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 within the ocean of darkweb pages. One strong motivation for analyzing onion sites and labeling them thematically is to identify malicious onions, e.g., sites doing illegal trade in weapons, drugs, etc. — it is important to detect such sites to be able to track potentially criminal activity. To solve this problem of detecting crawled onion sites with illicit content, we

[1] This paper analyzes a subset of this data from 23,585 onion sites

first focus on reliably characterizing onion categories by relevant keywords, which we use to annotate onions with sensitive category labels (e.g., weapons, drugs, hacker) and then try to identify sites actually containing illicit content. This need to characterize categories and assign the labels to onions motivated the development of the ATOL (Automated Tool for Onion Labeling) framework.

Another important area of application of thematic labels is Bitcoin transaction analysis on the darkweb. Anonymous digital currencies like Bitcoins are at the center of the darkweb economy, being used as the de-facto digital currency throughout thriving dark markets [37]. 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 from the LIGHTS repository. Onion thematic labeling can help us identify the nature of Bitcoins transactions. For example, if a Bitcoin address occurs more often in drug-related onion pages than weapon-related ones, then through our statistical analysis tools we label that Bitcoin address as a more likely indicator of drug-related transactions than weapon-related transactions. Another use of the Bitcoin transaction data is as a source of provenance data to help the thematic categorization. Bitcoin transactions are recorded publicly in block chains, allowing us to verify valid Bitcoin addresses used to make payments. Darkweb vendors can create different sites to obfuscate their operations, but if multiple onions share one or more common valid Bitcoin destination addresses, then it gives us useful evidence that these sites should be categorized together to have the same thematic label.

Contributions. We identify below the main challenges and contributions associated with our work:

1) Automated keyword discovery: ATOL has an automated keyword discovery algorithm (ATOLKeyword) that uses the available small-volume training data to automatically discover new category keywords. The scale at which ATOL operates and the fact that the size of the repository is growing everyday requires us to have this automated mechanism for keyword discovery. ATOLKeyword find keywords that are difficult for humans to detect, but are relevant nonetheless. For example, ATOL discovers "phpcredlocker" for the Hacker category (refers to a secure credential repository), "neurogroove" for the Drugs category (refers to a Polish drug website), and "flash-ball" for the Weapons category (refers to a French ball-launching weapon) — these are all valid and relevant keywords, but could be difficult for a human analyst to identify.

2) Classification with limited training data: Standard supervised classification frameworks [12, 26–28, 45], proposed in the context of surface web (i.e. for sites indexed by standard web search engines) are difficult to implement in the realm of darkweb mainly due to the lack of adequate training data. In fact, for this kind of problem, it is practically infeasible to create large volumes of human annotated data through crowdsourcing (e.g., using Amazon mTurk), as many onions may contain sensitive and illicit content which are legally prohibited from distribution. ATOL implements a new supervised categorization algorithm (ATOLClassify) that uses the keywords discovered by ATOLKeyword to get a novel TFICF-based feature weighting that gives substantial improvement in classifier performance, e.g., it gives a 12% improvement in F1-score over an analyst-provided baseline on the LIGHTS dataset (similar gains were also obtained on the 20 newsgroups dataset).

3) Clustering with external supervision: Unsupervised document clustering techniques [6, 15, 29, 44] often provide a good start to problems that suffer from a lack of labeled training data — a little supervision in the form of a small amount of labeled ground truth [1] or some prior known constraints [13, 40] can significantly bias the clustering process and drastically improve its performance. ATOLCluster implements a novel semi-supervised clustering algorithm that extends existing semi-supervised clustering approaches to be able to incorporate both labels and constraints, derived from multiple external knowledge bases, into the algorithm and improve the performance, e.g., it gives a 7% improvement over state-of-the-art semi-supervised clustering algorithms on the LIGHTS dataset (similar gains were also obtained on the 20 newsgroups dataset). We also show how a variant of ATOLCluster that bootstraps from unsupervised clustering results (in the absence of domain knowledge) can give better results than unsupervised clustering alone.

Roadmap. The rest of this paper is organized as follows. Section 2 provides an overview of OnionCrawler and ATOL. Section 3, 4 and 5 describe ATOLKeyword, ATOLClassify, and ATOLCluster respectively. Section 6 performs a thorough experimental evaluation of ATOL using data collected from the darkweb, as well as on a public dataset (20 newsgroups) for reproducibility. Section 7 discusses some relevant related work, while Section 8 outlines how ATOL is being deployed in actual systems. Section 9 concludes the paper and outlines promising areas of future work.

2 SYSTEM AND DATA OVERVIEW

We first provide a brief overview of the LIGHTS acquisition infrastructure constructed to discover new onion websites, crawl their content, and integrate them into our index repository. We developed two tools, HSProbe (Tor Hidden Service Prober) and OnionCrawler, to check the operational status of onion sites and crawl active onions.

1) HSProbe uses Tor's stem API [39] for accessing onion sites over the Tor protocol and interpret a broad range of HS protocol-status messages to determine how to proceed, as it encounters errors and unresponsive interactions with target hidden services. 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.

2) OnionCrawler is a fully automated crawling infrastructure to acquire new Tor onion domains (see Figure 1). We employ the OnionCrawler system continually, twice per day to address diurnal patterns in onion site availability. Our sources of seed data include various published onion datasets([32], [5], [25], [22]), .onion references from a large collection of recursive DNS resolvers [17], and an open repository of (non-onion) web crawling data, called Common Crawl [11]. Using these data sources as starting points, we developed tools to acquire additional onion addresses from the onion web. 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 is parsed and indexed into an ElasticSearch [16] database.

Data. ElasticSearch provides a query API for the indexed data, that we used to generate some of the analysis describing in the forthcoming sections. The data that we use in this paper is a subset of the data in the LIGHTS repository. Specifically, it includes upto 300 pages of extracted HTTP data from each of the 23,585 onion sites that we successfully deep-crawled. Due to the complex legal and ethical considerations involved in crawling the darkweb, our measurement study and resulting analysis was approved by our Institutional Review Board (IRB).

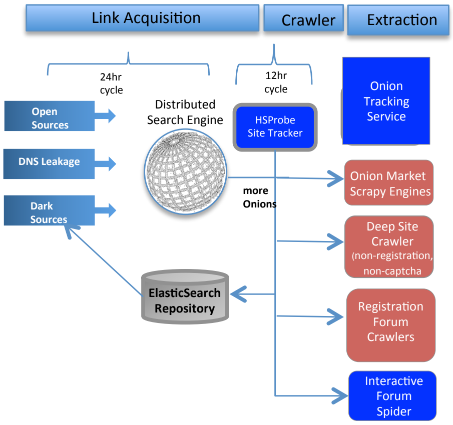

Figure 1: Overview of the LIGHTS acquisition infrastructure

2.1 ATOL

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 conduct different types of analysis. In this paper, we consider two instantiations of such analysis — *keyword discovery* and *categorization*. Keyword discovery is used to discover new keywords to characterize thematic categories — these keywords can then be used to extract relevant and important terms from onion sites. A key analysis task in ATOL is categorizing onions in different ways, e.g., according to their thematic labels, functional roles (e.g., hosted by seller), content type (e.g., blog). For categorization, we can use either classification or clustering — the former is more appropriate when we have a known set of categories and a reasonable-sized labeled training dataset, and the latter is relevant when we want to discover new categories and when the labeled dataset is small in size. We will discuss the details of the classification and clustering components of ATOL in Sections 4 and 5, especially a novel keyword-based classification approach for thematic labeling and a novel semi-supervised clustering algorithm.

3 ATOL: KEYWORD DISCOVERY

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/content words from onion text as well as existing manually-tuned keywords. We do this using a method that we call ATOLKeyword.

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 6 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, as shown below:

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

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 [30], 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 sufficient statistic matrix M, which is used to compute the final TFICF score. Each row of M gives us the high-scoring keywords for a particular category, which can then be used to discover new keywords for that category.

4 ATOL: CLASSIFICATION

We developed a new classification approach in ATOL using which we performed experiments on thematic categorization of onions. Using the keyword weights inferred by the TFICF algorithm, ATOL reweights the training data and uses that to train a classifier to

Algorithm 1: ATOL: TFICF Weighting for Keyword Discovery

```
function atolKeywords ($K_{expert}$,$C_{train}$,$L_{train}$,$\lambda$)
  Input  : $K_{expert}$ ← Seed list of keyword vectors (one vector
           per category) from domain expert; $C_{train}$ ← 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; $L_{train}$ ← set of class labels
           assigned to the corpus based on rater labelings; $\lambda$ ←
           weight multiplier on title words.
  Output : $M$ ← 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 $k \in K_{expert}$ 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 $c_1 \ldots c_m$ in training data then
          for category $c \in \{c_1 \ldots c_m\}$ do
              for word $w \in d$ 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 $t \in$ title of $d$ do
                  $M[c,t]$ += $\lambda \times$ titleCount($t$,$d$)/$m$ // count
                  of word in onion title, scaled up by
                  multiplier $\lambda$
              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
          $ICF_w = \log(\frac{|C|}{ICF_w})$ // |C| = number of
          categories
      end
      for each category $c$ in row($M$) do
          $M[w,c] = M[w,c] \times ICF_w$ // compute TF x ICF
      end
  end
  return $M$
```

Algorithm 2: ATOL: Classification into Thematic Labels

```
function atolClassify ($M$,$C_{train}$,$C_{test}$,$C_{unlabeled}$,$T$)
  Input  : $M$ ← Matrix where each row is a weighted vector of
           keywords in a category generated by the
           tficfKeywords function; $C_{train}$,$C_{test}$ ← Corpus of
           training and test documents, where each document $d$
           is represented by a $n$-dimensional bag-of-words
           vector $X \in \mathbb{R}^n$, $n$ is the vocabulary size and $l$ is the
           class label assigned to $d$; $C_{unlabeled}$ ← Corpus of
           unlabeled documents, for which ATOL will try to
           discover categories; $T$ ← threshold for category
           discovery.
  Output : $ML$ ← ML classifier trained using $C_{train}$; $L_{test}$ ←
           Labels assigned by $ML$ for every onion $d \in C_{test}$;
           accuracy ← onion classification accuracy in $C_{test}$.
  Train:
  $D_{train} \leftarrow \emptyset$
  for $d \in C_{train}$ do
      $X := \mathbb{R}^n$ bag-of-words vector for $d$ // Get BOW feature
      vector for $d$
      $l$ := class label for $d$
      $D_{train} \leftarrow D_{train} \bigcup$ tficfFeature($X$,$M$,$l$) // Get
      TFICF-weighted feature vector for $d$
  end
  Fit a classifier $ML$ on $D_{train}$.

  tficfFeature:
  $tficf\_features \leftarrow \emptyset$
  for $(k,w) \in M[l,:]$ do
      if $k \in X$ then
          $tficf\_features[k] = w \times X[k]$ // Scales up
          counts of words in $X$ by the corresponding
          weight of matching keywords in $M$, for the
          label $l$
      end
  end

  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 ← $\frac{correct}{|C_{test}|}$ // compute accuracy
  return ($ML$,$L_{test}$,accuracy)
```

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 these experiments are outlined in Section 6.

5 ATOL: CLUSTERING

In this section, we outline the approach of unsupervised and semi-supervised clustering used in ATOL. Unsupervised clustering is used in ATOL in the absence of labeled training data — when we have small amounts of supervision available in the form of labels or constraints, ATOL uses semi-supervised clustering. When we have sufficient labeled training data to train a classifier, we use the classification approaches outlined in Section 4. Note that when we have sufficient training data, both ATOLClassify and ATOLCluster give comparable results, as shown in Section 6.5.

5.1 Problem Formulation

We first present the different knowledge sources and then the overall objective function which we need to optimize.

(1) *Onion sites:* We represent the content of onions using normalized TFIDF vectors [15, 27]. We denote these vectors as $\mathbf{X}_i$ for $i \in [1, N]$, where N is the number of onions.

(2) *Domain Expertise:* For each category (cluster), the domain experts provide a specific set of keywords (all with same weights). We represent these set of words as a vector in the same feature space as the onion sites and then normalize them. We call these normalized vectors as *manual topics*, denoting them by $\mathbf{M}_j$ for $j \in [1, K]$, where K is the number of clusters.

(3) *Seeded Data:* Some of the data points $\mathbf{X}_i$ are manually annotated by the domain experts with labels S_i — we will use these points as data seeds in our clustering.

(4) *Must-link constraints:* During clustering, each data point $\mathbf{X}_i$ are assigned their corresponding label L_i. Data provenance also sometimes indicates that two data points $\mathbf{X}_i$ and $\mathbf{X}_j$ should be in the same cluster, i.e., there are must-link constraints enforcing that L_i and L_j should be same. We represent the must-link constraints using an adjacency matrix A where $A_{ij} = 1$ indicates must-link constraint between i and j, $A_{ij} = 0$ otherwise. We consider must-link constraints to be transitive in nature, so the transitive closure on A represents the complete set of must-link constraints.

Based on the above sources of data, we present our overall objective function where $\mathbf{C}_{1:K}$ represents the corresponding cluster centroids:

$$\theta = \sum_{j=1}^{K} \sum_{i \in \text{cluster } j} \mathbf{X}_i^T \mathbf{C}_j + \lambda_1 \sum_{j=1}^{K} \mathbf{C}_j^T \mathbf{M}_j +$$

$$\lambda_2 \sum_{i \in \text{seeds}} \mathbb{1}[S_i = L_i] + \lambda_3 \sum_{i,j \in \text{must-link}} \mathbb{1}[L_i = L_j].$$

The four terms in the objective function measures the following: (1) closeness of points to centroids, (2) closeness of topics to initial keyword lists, (3) satisfying seed constraints, and (4) satisfying provenance constraints.

5.2 Semi-supervised clustering

ATOL uses a novel semi-supervised clustering algorithm. A key feature that separates it from other semi-supervised clustering approaches, is that during every iteration it attempts to ensure that the cluster center is closely aligned to the manual topic assigned by the domain experts. The manual topics are used in the initialization as well as in the subsequent cluster assignment and center update state, to direct the clustering process better. Algorithm 3 outlines the novel semi-supervised clustering algorithm used in ATOL. In the absence of supervision, the unsupervised version of ATOLCluster has better performance than unsupervised clustering methods.

5.3 Convergence guarantee

The traditional K-Means [29] method can be looked upon as a hard assignment version of the Expectation Maximization algorithm (EM algorithm) [1, 7, 14]. We show that our cluster assignment and centroid update steps directly follows from the E-step and M-step of the EM algorithm and this guarantees its theoretical convergence. The deduction is shown below:

In the objective function θ, we can rewrite $\sum_{j=1}^{K} \sum_{i \in \text{cluster } j} \mathbf{X}_i^T \mathbf{M}_j$ as $N - \frac{1}{2} \times \sum_{i=1}^{N} (\mathbf{X}_i - \mathbf{C}_{L_i})^2$. Now, to get the optimal cluster means $\mathbf{C}_j$, we take the partial derivative of θ w.r.t. $\mathbf{C}_j$ for each j and set it to 0. This gives us,

$$\mathbf{C}_j \leftarrow [\sum_{L_i = j} \mathbf{X}_i + \lambda_1 \mathbf{M}_j]/[\sum_{L_i = j} 1]$$

This is the cluster re-estimation step (M-step) presented in the above algorithm — so in the M-step, the update gives us optimal cluster centroids. Now, looking at the function θ — for each $\mathbf{X}_i$, the L_i which maximizes its contribution to θ is given as,

$$L_i \leftarrow \text{argmax}_k [\mathbf{X}_i^T \mathbf{C}_k + \lambda_2 * \mathbb{1}[L_i = S_i] + \lambda_3 * \sum_{j < i \text{ and } A_{ij} > 0} \mathbb{1}[L_i = L_j]$$

This is the update in the cluster assignment step (E-step). Thus, in both the cluster centroid re-estimation and label assignment steps, the objective function is maximized, thereby ensuring the theoretical convergence of the algorithm to a local optimum of the objective function.

5.4 Unsupervised ATOLCluster

When domain knowledge is unavailable, we perform unsupervised topic modeling using Latent Dirichlet Allocation [6, 24] or Non-negative Matrix Factorization [44] using the same number of topics as clusters. We follow an unsupervised strategy to bootstrap and generate initial domain expertise using the output of generative models like LDA or NMF. The output from the LDA or NMF model is used as supervision in ATOLCluster. This bootstrap strategy

Algorithm 3: ATOL: Multi-source Semi-supervised Clustering

```
function atolCluster $(\mathbf{X}, \mathbf{M}, S, A, K, \epsilon, \lambda_1, \lambda_2, \lambda_3)$
  Input  : $\mathbf{X}_{1:N} \leftarrow$ normalized TFIDF vectors for documents,
           $K \leftarrow$ number of clusters, $\mathbf{M}_{1:K} \leftarrow$ manual topics
           assigned by domain experts, $S = \cup_i S_i \leftarrow$ labels of
           seeded data points $\mathbf{X}_i$, $A \leftarrow$ must-link constraints
           presented as an adjacency matrix, $\lambda_1, \lambda_2, \lambda_3 \leftarrow$
           hyper-parameters
  Output : $\mathbf{C}_{1:K} \leftarrow$ Final centroids of clusters, $\mathbf{L}_{1:N} \leftarrow$ final
           cluster assignments of $\mathbf{X}_{1:N}$
  $\mathbf{C}_{1:K} \leftarrow \mathbf{M}_{1:K}$
  $A \leftarrow \text{transitive\_closure}(A)$
  $t \leftarrow 0$
  $\theta^t \leftarrow 0$
  for $i \in [1, N]$ do
      $L_i \leftarrow \text{argmax}_k \{\mathbf{X}_i^T \mathbf{C}_k + \lambda_2 * \mathbb{1}[L_i = S_i] + \lambda_3 * \sum_{j<i \text{and} A_{ij}>0} \mathbb{1}[L_i = L_j]\}$
  end
  for $j \in [1, K]$ do
      $\mathbf{C}_j \leftarrow [\sum_{L_i=j} \mathbf{X}_i + \lambda_1 \mathbf{M}_j] / [\sum_{L_i=j} 1]$
      Normalize $\mathbf{C}_j$
  end
  $\theta^{t+1} \leftarrow$ Objective function value
  Repeat till $|\theta^{t+1} - \theta^t| \leq \epsilon$
  return $(\mathbf{C}, \mathbf{L})$
```

of generating domain knowledge beats unsupervised clustering methods, as we will show in Section 6.

6 EXPERIMENTAL RESULTS

We ran experiments to evaluate the effectiveness of ATOLClassify and ATOLCluster. This section describes the different experiments and analyses performed using ATOL code, which has been made available at: http://www.csl.sri.com/users/shalini/atol/.

6.1 Methodology

For the experiments using OnionCrawler data, we considered a dataset sampled from the LIGHTS repository snapshot of February 19th, 2016. Analysts annotated a sample of 481 onion *sites* with 3 labels — 163 sites labeled as Drugs, 255 as Hacker and 63 as Weapons. 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. Since, this was an expensive and time consuming process that could not be farmed off to crowd-sourcing services like Amazon mTurk (due to data sensitivity issues), only a small amount of ground truth could be generated. However, as indicated by our experimental results, even a small fraction of the labeled data can lead to good performance of the ATOLClassify and ATOLCluster algorithms.

We ran experiments with 5-fold cross-validation and stratified sampling of the labels. For the keyword discovery in Section 6.3 we use an existing training/test split used by the analysts. Our domain experts explicitly provided a specific set of keywords and phrases for all the three categories, which was used for both the classification and clustering algorithms (examples in Table 1. Furthermore, we computed must-link constraints [13, 40] for this data set from the Bitcoin transaction provenance data, which were used in the semi-supervised clustering algorithm.

Table 1: Keywords for different categories in the LIGHTS data, specified by analysts.

Category	Keywords/Phrases
Weapons	paperwork, background check, firearms, ak-47, kel-tek, bullet, armor piercing, weapons, luger, ruger, rifle, silencer, caliber
Drugs	psychedelic, hallucination, hanfplantage, steroids, cannabis, drugs, seed, weed, drugstore, hash, marijuana, lsd
Hacker	covertly, intercepts, confidential, secret, anonymity, crypto, encryption, security, keystroke logging, trojans, virus, malware, hackerware, warez, ransom

Both ATOLClassify and ATOLCluster have multiple hyper-parameters — we performed grid search on the parameters using a small validation data sample, and the results reported below are for the optimal set of hyperparameters.

6.2 Thematic Labeling with ATOLClassify

Table 2 shows the test-set performance of the algorithms using different feature weighting schemes.

Table 2: Test-set performance of ATOL classifiers on onion category prediction.

Features	Classifier	5-fold Accuracy
BOW	Multinomial Naive Bayes	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	Multinomial Naive Bayes	**0.964 ± 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

As outlined in Section 4, we have 2 phases of ATOLClassify—keyword generation and core classification. As shown in Table 2, we compare 4 methods of keyword generation:

(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 [30] 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 2 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[2] in ATOL:

(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 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 high accuracy of the ATOL thematic labeling classifier approach.

6.3 Keyword Discovery with ATOLClassify

Table 3 shows the top 10 keywords (sorted by TFICF) that were found by the keyword discovery algorithm in ATOL for the Hacker category when run on the analyst-provided train/test split, with explanations of why the discovered keywords are relevant for the corresponding categories. Similar keywords were also discovered for the Drugs and Weapons categories.

Table 3: Top 10 keywords discovered in the "Hacker" category by ATOLClassify on the LIGHTS data.

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

6.4 Comparative analysis of ATOLCluster

In our first experiment with semi-supervised clustering, we compare the performance of ATOLCluster to existing semi-supervised approaches viz. Unsupervised KMeans [29], Seeded KMeans [1], Constrained KMeans [2], COP-KMeans [40], and their combinations. The results are reported in Table 4. The evaluation metrics we used here are pairwise precision, pairwise recall and pairwise F1 score, where these metrics aim to see how accurately we can predict that two points that are in the same category in the ground truth are also in the same cluster in the clustering output. These pairwise measures are more appropriate for evaluating clustering algorithms [2].

The provenance-based versions use the must-link constraints from the bitcoin provenance data. As the results show, using just the domain expertise (without any seeded data or provenance information), ATOLCluster obtains almost 2% higher in F1-score than the its best competitor. Combining all the knowledge sources, ATOLCluster yields a F1-score which is at least 7% higher than other semi-supervised approaches, and significantly more than unsupervised KMeans.

Note that unsupervised ATOLCluster, which uses the output of unsupervised topic modeling to bootstrap the ATOLCluster algorithm in the absence of supervised data, gives significant performance improvement over unsupervised KMeans — this can be a promising categorization approach in the absence of labeled training data.

Table 4: Comparison of ATOLCluster to other clustering algorithms on the LIGHTS data.

Clustering Method	Pairwise Precision	Pairwise Recall	Pairwise F1 Score
Unsupervised KMeans	51.83	41.63	45.77
Seeded KMeans	63.96	70.14	66.91
Seeded + Constrained KMeans	64.30	70.20	67.12
Provenance-based COP-KMeans	56.63	50.70	53.50
Seeded + Provenance-based COP-KMeans	63.78	69.71	66.61
Seeded + Constrained + Provenance-based COP-KMeans	64.11	69.77	66.82
ATOLCluster (unsupervised)	61.68	71.75	68.66
ATOLCluster (using only domain expertise)	65.26	72.88	68.86
ATOLCluster (using all sources)	**70.21**	**77.40**	**73.63**

We also did two more studies measuring the performance of ATOLCluster: (a) As we increase the fraction of labeled training data, ATOLCluster reached high F1-score with a small label fraction and then the performance saturated. (b) As we increased label noise (by randomly permuting labels) ATOLCluster suffers only from minor performance degradation, demonstrating the robustness of ATOLCluster to noise.

6.5 Comparing ATOLCluster to ATOLClassify

Note that the precision, recall and F1 scores used to evaluate ATOLClassify are not directly comparable to the corresponding pairwise metrics used in ATOLCluster. Moreover, ATOLClassify and ATOLCluster are used in 2 different scenarios — the former is used when we have enough labeled data and fixed set of categories, while the latter is used in the absence of labeled training data (or when the amount of labeled data is small) or when the number of categories to discover in the data is not fixed apriori. However, to compare the performance of ATOLClassify and ATOLCluster, we consider the LIGHTS data where enough labeled training data is available. We already know the F1 score for ATOLClassify on this dataset (96%). From the ATOLCluster results, we map the clusters to the majority class labels and use that to compute the F1 score based on the labels — we see that the ATOLCluster algorithm using all the labels and constraints gives us accuracy of 95%, which is comparable in performance to ATOLClassify.

[2] using SciKit-Learn (http://scikit-learn.org)

6.6 20 Newsgroups (20NG) dataset

We also evaluated ATOL on the publicly available 20 Newsgroups (20NG) dataset,[3] which has been widely used in many previous works [1, 13]. This dataset contains document (in form of emails) from 20 different categories. In order to rigorously evaluate ATOL on the 20 Newsgroups dataset, we selected 3 similar categories that are difficult to categorize — comp.graphics, comp.os.ms-windows.misc, and comp.windows.x — having overall 2938 documents (973 in graphics, 980 in windows.misc, and 985 in windows.x). We used the header information present in each of these emails (specifically, the sender's email address) to deduce the must-link constraints.

6.6.1 ATOLClassify on 20NG data. We ran Multinomial Naive Bayes with TFICF weighting, the best-performing ATOLClassify algorithm in the onion analysis, on the 20NG data subset. Table 5 shows the comparison of performance of ATOLClassify to Multinomial Naive Bayes with BOW and TFIDF feature weighting, using 1 run on a 4:1 train/test split.

Table 5: Test-set performance of ATOL classifiers on 20NG.

Features	Classifier	Precision	Recall	F-measure
BOW	Multinomial Naive Bayes	0.844	0.814	0.810
TFIDF	Multinomial Naive Bayes	0.880	0.879	0.879
TFICF	Multinomial Naive Bayes	**0.897**	**0.893**	**0.893**

6.6.2 ATOLCluster on 20NG data. Table 6 compares the results of different variants of the semi-supervised ATOLCluster algorithm on the 20NG dataset. As we saw for the LIGHTS dataset, the full ATOLCluster algorithm that uses all sources of data (seeds, constraints) gives the best overall performance in terms of pairwise F1 score. When we ablate different parts of this overall algorithm (e.g., seeded initialization, constraints), the performance degrades.

Table 6: Comparison of ATOLCluster to other clustering algorithms on 20NG data.

Clustering Method	Pairwise Precision	Pairwise Recall	Pairwise F1 Score
Unsupervised KMeans	55.22	58.07	56.61
Seeded KMeans	64.47	66.37	65.41
Seeded Constrained KMeans	64.47	66.37	65.41
ATOLCluster (unsupervised)	67.42	68.57	67.99
ATOLCluster (using only domain expertise)	69.42	69.82	69.62
ATOLCluster (using all sources)	**70.20**	**71.43**	**70.68**

7 RELATED WORK

Darkweb: 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 [10, 37], as well as studies that have exploited flaws in Tor's hidden service design of onion domains [4, 5, 34], to reveal private .onion domains, including botnet C&Cs. Systems like DeepDive [33], 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 [10]. 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 [4] 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 guidelines [38]. 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 and classifying onion sites.

ML for cyber-security: Machine learning (ML) research in cyber-security has focused on different applications of ML to the openweb, e.g., modeling threat propagation for detecting malicious activities [9], adaptive trust modeling for cyber security [35], game-theoretic modeling of cyber security threats like information leakage [43], adaptive attacker strategy evolution [41], privacy-preserving data analysis [18], attacks on ML classifiers [8], or detecting user authenticity and spammy names in social networks [19, 20, 42]. However, not a lot of work has been done on analyzing the darkweb. Sabbah et al. [36] 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. In contrast, the TFICF-based keyword weighting scheme can use prior keyword distributions effectively.

8 DEPLOYMENT AND EXTERNAL EVALUATION

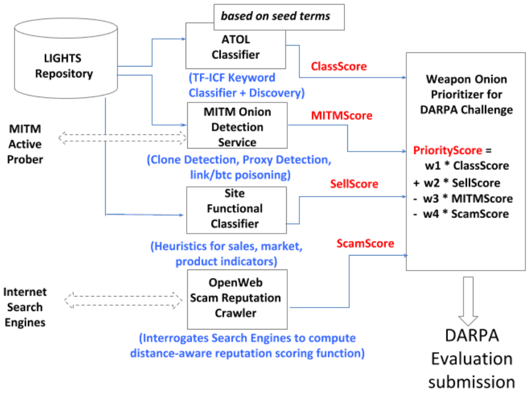
based on seed terms
LIGHTS Repository
ATOL Classifier
(TF-ICF Keyword Classifier + Discovery)
ClassScore
MITM Active Prober
MITM Onion Detection Service
(Clone Detection, Proxy Detection, link/btc poisoning)
MITMScore
Site Functional Classifier
(Heuristics for sales, market, product indicators)
SellScore
ScamScore
Internet Search Engines
OpenWeb Scam Reputation Crawler
(Interrogates Search Engines to compute distance-aware reputation scoring function)
Weapon Onion Prioritizer for DARPA Challenge
PriorityScore =
w1 * ClassScore
+ w2 * SellScore
- w3 * MITMScore
- w4 * ScamScore
DARPA Evaluation submission

Figure 2: LIGHTS architecture for DARPA Hackathon.

The LIGHTS system has been operationally deployed at SRI for over two years and has approximately 100 users across various organizations. The (OnionCrawler and ATOL) systems were evaluated in a DARPA 2016 Hackathon. The task was to come up with a prioritized list of new weapons-related domains based on 3rd-party-provided seed keywords, which were then assessed by independent 3rd-party evaluators. Each team could make use of image, text, site

[3] http://qwone.com/~jason/20Newsgroups

metadata or any other salient factors associated with 3rd-party-labeled seeds, to propose new sites that are potential locations for illegal weapons sales. Figure 2 shows the overall architecture of the LIGHTS crawling + analysis framework used in the Hackathon. ATOL provided >22% false positive reduction over the classification provided by the rest of the system. In addition, it identified a significant number of weapon-relevant sites and successfully completed the Hackathon evaluation task. The prototypes for the core algorithms in ATOL will be open sourced and made available soon.

9 CONCLUSIONS AND FUTURE WORK

This paper presented an automated system for crawling and analyzing content in the public Tor HS ecosystem. During the last two years, the combination of OnionCrawler (for extracting darkweb content) and ATOL (for analyzing the onion content) has automatically crawled and thematically categorized millions of pages in the darkweb. Our empirical evaluation on a snapshot of the LIGHTS repository shows the effectiveness of our keyword discovery algorithm. We demonstrate how ATOL's novel classification algorithms significantly outperform a keyword-based baseline algorithm used by analysts. We also developed a novel semi-supervised clustering framework, which shows promising initial results in the presence of limited or no training data. Empirical evaluation on the publicly available 20 Newsgroup dataset (for reproducibility) confirmed the efficacy of the ATOLClassify and ATOLCluster algorithms. Our experiments on the LIGHTS dataset showed that ATOLClassify gives a 12% performance gain over an analyst-provided baseline, while ATOLCluster gives a 7% improvement over state-of-the-art semi-supervised clustering algorithms.

In the future, we would like to extend the categorization framework of ATOL to include sequence analysis approaches such as LSTMs [23] and contextual LSTMs [21]. We would like to use the automated categorization in ATOL for automated portal generation, where sites (e.g., blogs, forums, wikis) can be added automatically to portals of different categories. Topic-driven extraction of various search terms can enable persona tracking on the darkweb (e.g., finding email or IM handles that are associated with weapons sales). We can also use ATOL for multi-classifier analysis, theme learning, graph analysis, and thematic census mining, leading to other far-reaching applications in darkweb analysis.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Patrick Lincoln for his valuable feedback regarding this work, Dr. Steven Cheung for his help developing the onion crawler, and Ken Nitz for his help with extracting the data and providing training labels. The ATOL software was partially funded by the National Science Foundation (NSF) under grants CNS-1314956, IIS-1116414 and ACI-1547467, by DARPA under Air Force Research Laboratory (AFRL) contract no. FA8750-14-C-0237, and by the Department of Homeland Security (DHS) Science and Technology Directorate. The views and conclusions contained herein are the authors' and do not represent the official views of DARPA, AFRL, DHS, NSF, or the U.S. government.

REFERENCES

[1] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. 2002. Semi-supervised Clustering by Seeding. In *ICML*.

[2] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. 2004. A probabilistic framework for semi-supervised clustering. In *KDD*.

[3] Natasha Bertrand. 2015. ISIS is taking full advantage of darkest corners of internet. *Business Insider* (2015).

[4] A. Biryukov, I. Pustogarov, F. Thill, and R. P. Weinmann. 2014. Content and Popularity Analysis of Tor Hidden Services. In *ICDCSW*.

[5] A. Biryukov, I. Pustogarov, and R.-P. Weinmann. 2013. Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization. In *IEEE-SP*.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *JMLR* (2003).

[7] Léon Bottou and Yoshua Bengio. 1995. Convergence Properties of K-Means Algorithms. In *NIPS*.

[8] Igor Burago and Daniel Lowd. 2015. Automated Attacks on Compression-Based Classifiers. In *AISec*.

[9] Kevin M. Carter, Nwokedi C. Idika, and William W. Streilein. 2013. Probabilistic threat propagation for malicious activity detection. In *ICASSP*.

[10] N. Christin. 2013. Traveling the Silk Road: A Measurement Analysis of a Large Anonymous Online Marketplace. In *WWW*.

[11] Common Crawl Foundation. 2016. Common Crawl. (2016). http://commoncrawl.org.

[12] Ariyam Das, Chittaranjan Mandal, and Chris Reade. 2013. Determining the User Intent Behind Web Search Queries by Learning from Past User Interactions with Search Results. In *COMAD*.

[13] Ian Davidson and S. S. Ravi. 2005. Clustering with Constraints: Feasibility Issues and the k-Means Algorithm. In *SDM*.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society, Series B* 39, 1 (1977).

[15] Inderjit S. Dhillon, Yuqiang Guan, and J. Fan. 2001. Data Mining for Scientific and Engg. Applications. Chapter Efficient Clustering of Very Large Document Collections.

[16] Elastic. 2016. Elasticsearch. (2016). https://www.elastic.co/.

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

[18] James R. Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. 2016. On the Theory and Practice of Privacy-Preserving Bayesian Data Analysis. *CoRR* abs/1603.07294 (2016).

[19] David Freeman, Sakshi Jain, Markus Dürmuth, Battista Biggio, and Giorgio Giacinto. 2016. Who Are You? A Statistical Approach to Measuring User Authenticity. In *NDSS*.

[20] David Mandell Freeman. 2013. Using Naive Bayes to Detect Spammy Names in Social Networks. In *AISec*.

[21] Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual LSTM (CLSTM) models for Large scale NLP tasks. In *KDD-DLKDD Workshop*.

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

[23] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997).

[24] Matthew Hoffman, David M. Blei, and Francis Bach. 2010. Online Learning for Latent Dirichlet Allocation. In *NIPS*.

[25] J. Nurmi. 2016. Ahmia Search Engine. (2016). https://ahmia.fi/.

[26] B.J. Jansen and U. Pooch. 2001. A review of Web searching studies and a framework for future research. *J. American Society of Information Science and Technology* 52, 3 (2001).

[27] In-Ho Kang and GilChang Kim. 2003. Query Type Classification for Web Document Retrieval. In *SIGIR*.

[28] Uichin Lee, Zhenyu Liu, and Junghoo Cho. 2005. Automatic Identification of User Goals in Web Search. In *WWW*.

[29] J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symp. on Mathematical Statistics and Probability.*

[30] Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

[31] Tor Metrics. 2016. Unique .onion Addresses. *https://metrics.torproject.org/hidserv-dir-onions-seen.html* (2016).

[32] Dark Net. 2011-2015. Market Archives. *www.gwern.net/Black-market%20archives* (2011-2015).

[33] F. Niu, C. Zhang, C. Re, and J. W. Shavlik. 2012. DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. In *VLDS*.

[34] G. Owen and N. Savage. 2016. Empirical analysis of Tor Hidden Services. *IET Info. Sec.* 10 (2016). Issue 3.

[35] Paul Robertson and Robert Laddaga. 2012. Adaptive Security and Trust. In *SASOW*.

[36] Thabit Sabbah, Ali Selamat, Md. Hafiz Selamat, Roliana Ibrahim, and Hamido Fujita. 2016. Hybridized term-weighting method for Dark Web classification. *Neurocomputing* 173, 3 (2016).

[37] K. Soska and N. Christin. 2015. Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem. In *USENIX*.

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

[39] Tor Project. 2016. Stem. (2016). https://stem.torproject.org/.

[40] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained K-means Clustering with Background Knowledge. In *ICML*.

[41] Michael L. Winterrose, Kevin M. Carter, Neal Wagner, and William W. Streilein. 2014. Adaptive Attacker Strategy Development Against Moving Target Cyber Defenses. *CoRR* abs/1407.8540 (2014).

[42] Cao Xiao, David Mandell Freeman, and Theodore Hwa. 2015. Detecting Clusters of Fake Accounts in Online Social Networks. In *AISec*.

[43] Haifeng Xu, Albert Xin Jiang, Arunesh Sinha, Zinovi Rabinovich, Shaddin Dughmi, and Milind Tambe. 2015. Security Games with Information Leakage: Modeling and Computation. In *IJCAI*.

[44] Wei Xu, Xin Liu, and Yihong Gong. 2003. Document Clustering Based on Non-negative Matrix Factorization. In *SIGIR*.

[45] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. 2004. Learning to Cluster Web Search Results. In *SIGIR*.