Future Generation Computer Systems 157 (2024) 67–81

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A Big Data architecture for early identification and categorization of dark web sites

Javier Pastor-Galindo [a,*], Hông-Ân Sandlin [b], Félix Gómez Mármol [a], Gérôme Bovet [b], Gregorio Martínez Pérez [a]

[a] Department of Information and Communications Engineering, University of Murcia, 30100 Murcia, Spain
[b] Cyber-Defence Campus, Armasuisse Science and Technology, Switzerland

ARTICLE INFO

Dataset link: https://doi.org/10.17632/9npmf5v6kr.1

Keywords:
Dark web
Big Data
Web content categorization
BERT
Tor network analysis
Threat intelligence

ABSTRACT

The dark web has become notorious for its association with illicit activities and there is a growing need for systems to automate the monitoring of this space. This paper proposes an end-to-end scalable architecture for the continuous early identification of new Tor sites and the daily analysis of their content. The solution is built using an Open Source Big Data stack for data serving with Kubernetes, Kafka, Kubeflow, and MinIO, continuously discovering onion addresses in different sources (threat intelligence, code repositories, web-Tor gateways, and Tor repositories), downloading the HTML from Tor and deduplicating the content using MinHash LSH, and categorizing with the BERTopic modeling (SBERT embedding, UMAP dimensionality reduction, HDBSCAN document clustering and c-TF-IDF topic keywords). In 93 days, the system identified 80,049 onion services and characterized 90% of them, addressing the challenge of Tor volatility. A disproportionate amount of repeated content is found, with only 6.1% unique sites. From the HTML files of the dark sites, 31 different low-topics are extracted, manually labeled, and grouped into 11 high-level topics. The five most popular included sexual and violent content, repositories and search engines, carding, cryptocurrencies, and marketplaces. During the experiments, we identified 14 sites with 13,946 clones that shared a suspiciously similar mirroring rate per day, suggesting an extensive common phishing network. Among the related works, this study is the most representative characterization of onion services based on topics to date.

1. Introduction

The dark web is the anonymous portion of the Internet, not indexed by well-known search engines, that has gained notoriety due to its association with illegal activities such as drug trafficking, human trafficking, and cybercrime [1]. Recently, this space has been exploited in the Ukrainian–Russian conflict as a platform for data leaks, propaganda dissemination, cyberwarfare, illicit arms trading, organized activism or censorship avoidance, highlighting the importance of the cyberdimension in nowadays problems [2].

In particular, Tor (The Onion Routing) is the most popular dark web network used for anonymous communication and web browsing [3], which is the focus of this paper. Given the illicit nature of Tor, there is a growing need for automatic systems and digital platforms to automate the monitoring of this space to contribute to situational awareness, generate intelligence, support decision-making, identify emerging threats or perform a quick reaction [4]. Manual search and individual categorization prove impractical, necessitating technological assistance for online identification and topic extraction. This is particularly crucial for early and scalable detection of illicit services, assisting investigators and law enforcement agencies.

Several crawlers and platforms in the literature collect and analyze Tor onion services. However, they employ traditional schemes and outdated techniques, which are not efficient and scalable for continuous and intensive monitoring [5]. This work addresses these weaknesses and contributes to the state-of-the-art by proposing an innovative solution with modern techniques for effective dark web monitoring, facing the challenges of volatility, redundancy and variety related to the nature of Tor sites [6].

Firstly, earlier work has found that after 24 h of publication, fewer than half of the observed onions are reachable [7]. Highly volatile and short-lived services impact crawling performance, in which the number of onion addresses found is much higher than the number of onion addresses that are still reachable for analysis afterwards [8]. Most discovered links are already offline by the time they are published in

* Corresponding author.
E-mail addresses: javierpg@um.es (J. Pastor-Galindo), hongan.sandlin@ar.admin.ch (H.-Â. Sandlin), felixgm@um.es (F.G. Mármol), gerome.bovet@ar.admin.ch (G. Bovet), gregorio@um.es (G.M. Pérez).

https://doi.org/10.1016/j.future.2024.03.025
Received 3 July 2023; Received in revised form 10 January 2024; Accepted 15 March 2024
Available online 20 March 2024

related work, having a huge discrepancy between identified services and active ones due to the time between deployment, identification and subsequent monitoring [9].

Secondly, the prevalence of duplicate websites, such as mirrors or phishing sites, can lead to an over-representation of specific services in the data set, making it difficult to portray the distribution of web content [8] accurately. Phishing is typical due to the infeasibility of checking the authenticity of onion domains due to their anonymous nature [10]. Mirrors differ from phishing sites since the owners of a web service duplicate them to mitigate DDoS attacks [11].

Thirdly, the topics and categories extraction of Tor onion services poses another significant challenge. Manual categorization is time-consuming and labor-intensive, and relying on keyword or bag-of-words approaches may not always yield accurate results. Some studies have utilized machine learning models, such as Sparse Composite Document Vector, Random Forest, Support Vector Machines, Naive Bayes, and LightGBM, to infer the categories of Tor onion services [11]. Others have employed probabilistic models based on word distributions or third-party software that automatically extract topics without human intervention [12]. However, the challenge of accurately and efficiently categorizing Tor onion services persists, especially when there are advanced techniques available today in the area of Natural Language Processing (NLP) based on Hidden Markov Models, Neural Networks and Bi-directional Encoder Representations from Transformers (BERT) [13], or the recent paradigm of prompt learning [14].

This paper proposes a new scalable, efficient, and cost-effective architecture for automated collection and analysis of large-scale contents of the Tor network in near-real time. The purpose is to design and prototype a state-of-the-art solution for large-scale categorization of dark web sites, particularly addressing the three aforementioned challenges identified in the literature. In particular, the contributions are as follows:

1. Design, implement and deploy a novel Kubernetes solution for the large-scale near real-time collection and analysis of Tor onion services. The solution includes (i) four different types of sources to early identify fresh onion addresses to mitigate high volatility and capture before dying, (ii) the application of the MinHash LSH (Locally Sensitive Hashing) algorithm to deduplicate dark sites efficiently, and (iii) the categorization employing BERTopic unsupervised approach.
2. Representative research results on extensive experiments on Tor v3 landscape, (i) collecting, to the best of our knowledge, the second most representative sample of Tor literature up to date, (ii) revealing relevant insights of the considerable proportion of duplicates on the Tor network, and (iii) developing, as far as we know, the most representative characterization of onion services based on topics up to date.

2. Related work

2.1. Tools for analyzing Tor onion services

Many investigations implement crawlers to explore the dark web. However, this section highlights more sophisticated solutions with additional functions for crawling, as depicted in Table 1:

- *Dark Crawler* [15] is a crawling tool that starts at user-specified websites, retrieving and analyzing HTML content from discovered onion services, while storing statistics, content, and image hashes. It identified 10,163 onion sites.
- *Automated Tool for Onion Labeling (ATOL)* [16] is an infrastructure that crawls pages twice per day from seed sets and web searches, extracts the main themes, and indexes the results on Elasticsearch. The project analyzed 529 unique Tor pages.

Table 1
Comparison of tools for analysis of Tor onion services.

Tool	Main feature	Onions
Dark Crawler [15]	Analyzes HTML	**10,163**
ATOL [16]	Twice daily crawl	529
Web Mining Toolkit [17]	Java-based	5,144
AF for Darkweb [18]	Includes Maltego	–
Docker Tor crawler [19]	Uses Docker	2,527
MASSDEAL [11]	Classifies resources	7,831
DWTIA [20]	Large data volume	8,000
Web crawl system [21]	Uses Selenium	3,000
Black Widow [22]	Uses Scrapy, Docker	2,066
Monitoring App [23]	Visualizes plots	3,000

- *Tor-oriented Web Mining Toolkit* [17] is a Java-based application with four components and two external modules that facilitates massive web crawling, indexing, and text mining. It analyzed 5,144 onion services in six weeks.
- *Analytical Framework for Darkweb scraping and analysis* [18] proposes a methodology for scraping targeted marketplaces. This tool is not designed for large-scale monitoring, and AppleScript programs are adapted accordingly for the analysis of each target. Notably, it includes Maltego to extend knowledge about vendors found.
- *Docker-based Tor crawler* [19] uses Docker and paralleled collectors, an analyzer, and a cloud manager to download and analyze dark sites. It monitored the status of 2,527 onion services for five months.
- *MASSDEAL* [11] is an exploration and analysis tool built with Python and SQLite that retrieves onions, detects duplicates, classifies resources, discovers new services, and replenishes known services. It was deployed over 105 days and collected 7,831 sites.
- *Dark Web Threat Intelligence Analysis (DWTIA) Platform* [20] is built with data acquisition, indexing, analysis and visualization modules to process large volumes of data collected from the dark and surface web to identify illicit services. It inspected more than 8,000 sites on the dark web.
- *Dark Web crawling system* [21] uses Selenium, a seed URL collector that periodically visits public dark web services to obtain onion seeds, and a sub URL collector that stores the address, content, and screenshots related to each seed URL. It identified 3,000 onion services in one month and a half.
- *Black Widow crawler* [22] is an architecture, which searches, identifies, and indexes secret services, black markets, and criminal patterns using a combination of technologies like Scrapy, Docker, Apache Solr, and MongoDB. The crawler obtained 2,066 onion services in one week.
- *Darkweb Monitoring Application* [23] offers a methodology for analyzing word occurrence, category classification, and visualization of resulting plots. This Python-based tool was tested once with a bulk of 3,000 onion sites.

This paper proposes a novel Big Data architecture, improving upon the limited scalability and efficacy of existing methods, for near real-time identification and categorization of Tor sites.

2.2. Analysis of tor onion services

2.2.1. Temporality and lifetime

High availability, fixed location, and pseudo-infinite longevity are not expected in Tor. An experiment launched in 2018 determined that approximately 30% are never reachable, 50% of the observed onion services were deactivated after 24 h, and 60% die after 300 hours [7]. Another research revealed that only 36% of the identified sites were alive for 18 weeks [24].

These aspects limit the study of services on the dark web, as seen in Table 2. For instance, the number of concurrently online

Table 2
Tor accessibility in previous studies.

Study	Total onions	Active onions	Portion
[25]	7,000	1,450	20.7%
[26]	198,050	7,257	3.7%
[12]	47,439	**14,232**	30%
[27]	250,000	7,000	2.8%
[28]	124,589	3,536	2.8%
[29]	12,882	4,509	**35%**
[30]	25,742	6,227	24.2%
[31]	15,503	4,089	26.4%
[19]	25,261	2,527	10%

services measured was around 1,450 out of more than 7,000 identified addresses [25], 7,257 onion links were active out of 198,050 [26], 30% was online at least 90% of the experiment with 47,439 onions identified [12], 7 K Tor pages were alive out of more than 250 K addresses [27], or strategies that returned 124,589 addresses with only 3,536 active [28].

Other studies highlighted the difficulty in reaching onion services due to their transient nature; only a small percentage were accessible. For example, only 35% of sites could be accessed successfully out of 12,882 onion addresses [29], 6,227 were online and accessible at the time of the crawl out of 25,742 onion services discovered [30], 4,089 were up and responding to crawler's request out of 15,503 [31], or 2,527 were open from a total of 25,261 onion addresses [19].

In addressing coverage and access issues, the proposed architecture is explicitly configured with different types of live data sources which are continuously monitored.

2.2.2. Duplicates, mirrors and phishing sites

Some studies measure dark content redundancy, as shown in Table 3, demonstrating that the same service may be mirrored in different onion addresses:

- In an analysis of 2,527 onion services [19], the authors decided to consider two onion services to be the same when the cosine similarity of titles and HTML files were over 90% and 80%, respectively, reducing the sample to 2,014 unique sites.
- A four-month experiment [11] used the hashes of HTML documents, screenshots and Jaccard distance of titles (over 0.8) to show that approximately 80% of 7,831 monitored onion services were unique and 20% had more than one mirror URL (on average, 4.89 onion addresses).
- The Jaccard distance method was also employed in a large-scale study [12] of 45,135 dark sites to cluster HTML pages and extract 33,217 duplicates (73.59%) divided into 1,021 clusters with different average similarity coefficients.

The mirroring is specifically addressed in the literature from a cybersecurity perspective:

- Barr-Smith and Wright [32] studied the imitation of phishing sites using the *html-similarity*[1] library to estimate the page structure-based similarity. Of 11,533 services, 33.573% were unique, 45.192% were imitations (duplicates or clones), and 21.23% were default pages.
- Brenner et al. [33] evidenced that darknet websites can be effectively compared for similarity using category structure features such as HTML-Tag, HTML-Class, and HTML-DOM-Tree, along with metadata features like File Content and Links-To. They demonstrated that out of 258 single vendor shops, 20% were found to be duplicates, while 31% exhibited high levels of similarity.

[1] https://github.com/matiskay/html-similarity

Table 3
Studies deduplicating onion services.

Study	Technique	Onions	Duplicates
[19]	Content Similarity	2,527	80%
[11]	Content Similarity	7,831	20%
[12]	Content Similarity	45,135	73.59%
[32]	Structural Similarity	11,533	45.92%
[33]	Structural Similarity	258	20%
[10]	Content Clustering	28,928	80.23%
[34]	Image Similarity	4,210	**95.44%**

- A method to identify phishing candidates [10] was employed with clustering to a large number of domains based on shared titles and content, analyzing the equivalence of content among onion domains. The study revealed that out of 28,928 onion domains, only 5,718 website groups with distinct content were present, and 901 phishing domains with duplicated content were identified.
- Alternatively, Steinebach et al. [34] demonstrated that comparing images was the best duplication detection for 4,210 onion services in contrast to comparing identical texts or onion addresses.

In our framework, we integrate content similarity based on Jaccard score with MinHash LSH algorithm in the processing pipeline to identify duplicates efficiently and reduce the computing overload of the workflow.

2.2.3. Topic modeling and document classification

Understanding and categorizing Tor services is crucial due to the vast array of complex and often illicit content within the dark web. Primarily text-based, these services pose substantial challenges in terms of volume and ethical implications for manual interpretation.

Despite the need for automatic approaches to process big amounts of documents and avoid controversial human analysis, early studies utilized manual categorization, as seen in Table 4:

- Guitton [35] examined over a thousand onion services and classified them into 23 categories.
- Owen and Savage [36] conducted a detailed analysis of onion sites and grouped them into 22 classes.
- An even larger manual inspection [30] led to the classification of more than four thousand onion services into 31 categories.
- Two studies [27,28] focused on creating and extending the DUTA dataset, manually tagging onions in 25 classes.
- Lee et al. [21] classified three thousand dark sites into 15 categories.
- Another project [31] implemented a Java application to assist in manually categorizing 14 topics.

Some methodologies are based on the presence of determined keywords in the text or employing the distribution in bag-of-words (BOW):

- Kinder et al. [29] identified eight categories from over six thousand onion services.
- Alaidi et al. [23] followed a similar approach, identifying six subjects.

Other procedures involved clustering based on the frequency of word occurrences:

- Sánchez-Rola et al. [26] led to the manual tagging of onion services into seven categories, and an approach that employed Naive Bayes to classify onion sites into six categories [38].
- The ATOL project [16] identified three categories based on Term Frequency - Inverse Corpus Frequency (TF-ICF) with clustering.
- A similar approach but with Term Frequency — Inverse Document Frequency — Inverse Category Score (TF-IDF-ICS) led to the identification of six classes [11].

Table 4
Studies categorizing Tor onion services.

Study	Type	Onions	Topics	Sample
[35]	Manual	1,171	23	*Child abuse, personal, hacking, blackmarket, porn*
[36]	Manual	–	22	*Drugs, market, fraud, bitcoin, mail, wikis, media*
[30]	Manual	4,102	31	*Adult, bitcoin, directory, drugs, electronics*
[27,28]	Manual	–	25	*Drugs, credit cards, porn, hacking, cryptos*
[21]	Manual	3,000	15	*Counterfeits, drugs, hacking, forgery*
[31]	Manual	2,419	14	*Finance, search engine, drugs, credentials*
[29]	Keywords	**6,227**	8	*Marketplaces, drugs, fraud, cybercrime, porn*
[37]	Keywords	4,000	4	*Dark markets, socially unjust content, bitcoins*
[26]	BOW + Clustering	5,883	7	*Directories, default messages, market, bitcoins*
[38]	BOW + Naive Bayes	2,542	6	*Hacking, drug, develop, porn, news, casino*
[16]	TF-ICF + Keywords + Clustering	481	3	*Weapons, drugs, hacker*
[11]	TF-IDF-ICS + Keywords + Clustering	445	6	*Listing, login, market, security, porn, other*
[25]	LDA	1,481	250	*Trading, financial, politics, intelligence, porn*
[39]	LDA	3,288	9	*Directory, bitcoin, news, email, multimedia*
[40]	FastText + SCDV + LigthGBM	–	15	*Adult, communication, cryptos*
[23]	TF-IDF + SVM	3,115	4	*Drug, fake id, hacking, weapon*
[41]	*Mallet + uClassify*	1,813	18	*Drugs, adult, counterfeit, weapons, politics*
[17]	*Cogito* software	–	17	*Cybersecurity, fraud, drugs, inf. systems, media*

Probabilistic and machine learning models have been less employed in Tor research for topic extraction:

- Latent Dirichlet Allocation (LDA) was used in Tor-based projects like one that identified 250 themes [25], and another that identified nine categories [39].
- Kawaguchi and Ozawa [40] used a combination of FastText, Sparse Composite Document Vector (SCDV), and LightGBM to classify onion services into 15 categories.
- Nair and Kannimoola [37] used automatic labeling and classification methods to discover four categories in dark sites.

Finally, applied projects use third-party software for automated extraction of topics:

- Biryukov et al. [41] used Mallet and uClassify tools to derive 18 categories.
- Celestini and Guarino [17] used the Cogito semantic engine to extract tags based on their topic taxonomy, resulting in 17 classes.

However, the current state of the art in topic modeling and document classification is marked by significant advancements in neural networks and word embeddings [42], techniques not present in the categorization of Tor services reviewed so far. Traditional methods such as LDA have been enriched with word embeddings to enhance their effectiveness [43].

Additionally, models primarily built around embeddings have gained prominence, showcasing the potential of such methodologies for topic modeling [44]. Another emerging trend is simplifying the topic-building process by clustering word and document embeddings [45]. One of the most recent developments is BERTopic [46], a clustering approach that integrates a class-based variant of Term Frequency-Inverse Document Frequency (TF-IDF) to create topic representations that are being increasingly adopted [47,48].

In order to drive a significant shift in dark site categorization, our architecture integrates the aforementioned BERTopic technique.

3. Architecture for identification and categorization of tor sites

The proposed system for the automatic collection and analysis of onion services is divided into separate layers, components, and technologies, illustrated in Fig. 1. Adopting a microservices architecture managed via Kubernetes is fundamental for addressing critical aspects such as scalability, resilience, and maintainability. This decision was motivated by the flexibility and scalability provided by the microservices approach, which allows individual components to be independently scaled according to demand. As a proven orchestration platform, Kubernetes ensures reliable deployment, networking, and scaling of these microservices, improving the overall resilience. Additionally, the loose coupling inherent in the microservices design simplifies maintenance, as each service can be independently updated or replaced without impacting the entire system.

The architecture includes a real-time ingestion layer with a pool of spiders to crawl multiple data sources and extract undiscovered onion addresses. A group of downloaders saves the HTML file of each Tor site using Tor proxies. At the end of the day, a batch processing job analyzes the accumulated onion services, identifies duplicates, classifies languages, and extracts topics. The architecture follows the ELT (Extract, Load, Transform) paradigm. The code is publicly available at github.com/javier-pg/dark-web-architecture and detailed information on each layer and component is provided in the following sections.

3.1. Data sources

The role of data sources is pivotal to facilitating the constant acquisition of new onion addresses (complex links composed of 56 base32-coded characters with the ".onion" top-level domain). In this regard, four types of data sources are considered to ensure a frequent supply of new Tor sites to be ingested and analyzed:

- *Threat intelligence*. Provides valuable information on known threat actors, their tactics, techniques, and procedures (TTPs), and the infrastructure they use. To the best of our knowledge, we are the first to inspect indicators of compromise (IoC), such as domain names, hostnames, URLs, DNS records and advertising links to match onion domains. The latter is used by cybercriminals to host anonymous illicit services or command and control servers on the Tor network. The architecture visits six different threat intelligence feeds.[2]
- *Code repositories*. Versioning platforms offer a rich resource of open-source software projects, scripts, files, and configurations. For the first time among related works, as far as we know, GitHub repositories are inspected to extract onion addresses hard-coded by developers.
- *Web-Tor gateways*. Services that act as proxies between traditional explorers and the Tor network, such as Tor2web, dump Tor pages to the surface web. This fact can be exploited with the keyword search algorithm [49], based on the dorking *site* operation, such as *site:tor2web.org* to extract onion domains indexed in search

[2] Including AlienVault OTX, MalwareWorld, Maltrail, Notracking blocklists, Little Snitch blocklists, and USOM (Computer Emergency Response Team of Turkey) blocklists.

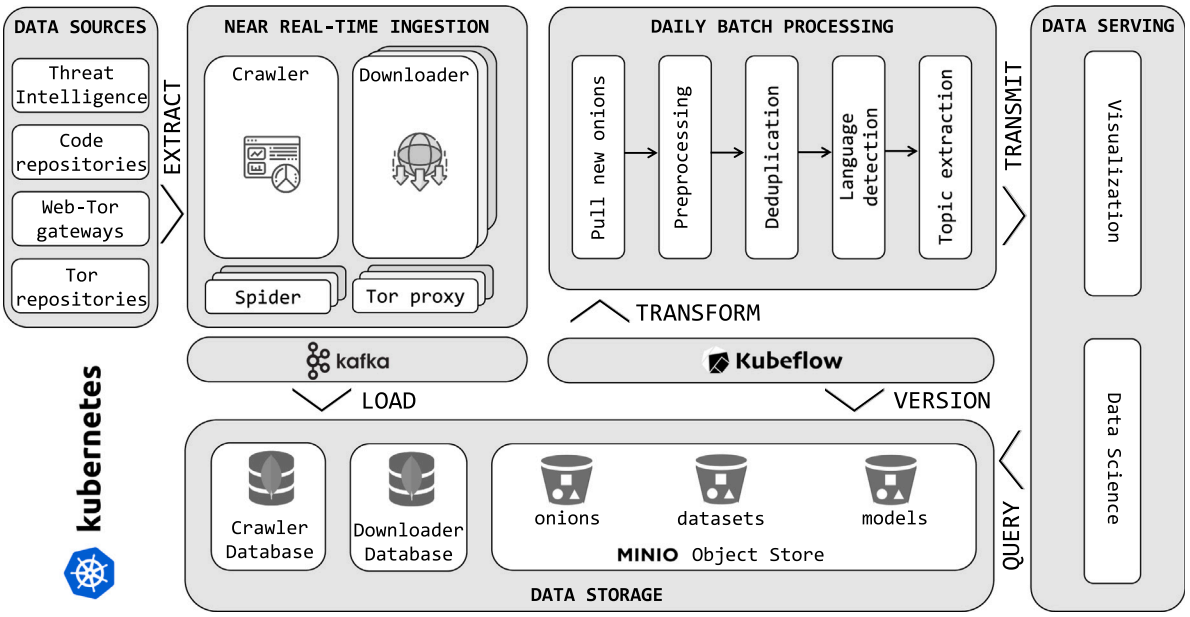

Fig. 1. Architecture for the continuous collection and analysis of Tor onion services.

engines under Web-Tor proxies. The architecture considers 19 different alternatives of proxy services.[3]

- *Tor repositories*. Compilations of Tor links are widely used in the literature to gather onion addresses due to the accuracy, speed, and simplicity of consulting ready-to-use compilations [9]. The architecture is configured with 13 different repositories.[4]

The four types of onion sources provide a quantity and variety of onions daily, reducing the search bias potential of this type of platform.

3.2. Near real-time ingestion

The Near Real-time Ingestion layer extracts onion addresses from data sources and downloads the HTML document. The onion service is not immediately identified as soon as it appears online. Instead, it is immediately detected after it is advertised in the data sources and accessed by the system.

3.2.1. Crawler and spiders

The architecture deploys a virtual instance of Crawlab,[5] a Web Crawler Management Platform that runs custom web crawlers. This platform provides a user-friendly interface for importing, running, managing, and monitoring spiders, making them traceable, scalable, and stable. Our solution imports, configures and schedules four Python spiders:

- *Threat intelligence spider*: Scrapes the six feeds based on regular expressions of the listings every six hours.
- *Code repositories spider*: Searches with Grep.app[6] and Sourcegraph[7] for GitHub content matching the string *d.onion* (the last compulsory characters of Tor domains), every six hours.
- *Web-Tor gateways spider*: Uses DuckDuckGo to identify indexed onion links from the 19 web-Tor proxies every six hours. Programmatic search permissions were requested from Google and Bing without success.
- *Tor repositories spider*: Employs Scrapy[8] to crawl the 13 repositories with a depth of 3 and a download delay of 0.2 s every twelve hours.

Each spider is independent, can be easily extended to request for more resources, and maintains the information of its operations in the Crawler Database. The execution frequencies are configurable, and current decisions are based on tests for a good trade-off between intensive search and efficacy to capture new inclusions at an early stage, especially for those volatile onion addresses that tend to disappear or be eliminated quickly in very short periods.

3.2.2. Downloaders and Tor proxies

The downloader is a Python-based Kafka consumer that retrieves the new onion addresses, connects to the Tor network through the SOCKS proxy, and saves the associated raw HTML page (without media and Javascript) and the MinIO object storage instance. Particularly, the architecture runs a set of downloaders scaled with Kubernetes to the desired number of parallel replicas (ten in our experimental deployment). Similarly, it scales Tor proxies according to the needs (five in our running cluster).

A Kafka source connector on the Crawler Database creates one Kafka topic per data source to forward new Tor links to the downloaders automatically. Therefore, the latter receives the new onion addresses when the Crawler inserts new entries in each data source collection (event-driven change data capture). Each replica receives one Kafka partition per topic.

On the one hand, MinIO saves the raw HTML file in the *onions* bucket. On the other hand, the Downloader Database keeps track of every onion in the system that has been found. In this sense, an HTML page will only be downloaded if it has not been saved before.

[3] Including onion.city, onion.direct, onion.gq, onion.link, onion.nu, onion.pw, onion.top, onion.pet, tor2web.to, tor2web.fyi, tor2web.io, onion.ws, onion.foundation, onion.dog, onion.moe, onion.re, onion.ly, onion.pet, and onion.cab.

[4] Including Ahmia, OnionRanks, TheHiddenWiki, Dark.Fail, DarknetLive, TheDeepSearches, FreshOnions, Torch, Tor Links, Tor66, OnionLand Search, TorDex, and H-Indexer.

[5] https://github.com/crawlab-team/crawlab.

[6] https://grep.app

[7] https://about.sourcegraph.com/code-search

[8] https://scrapy.org/

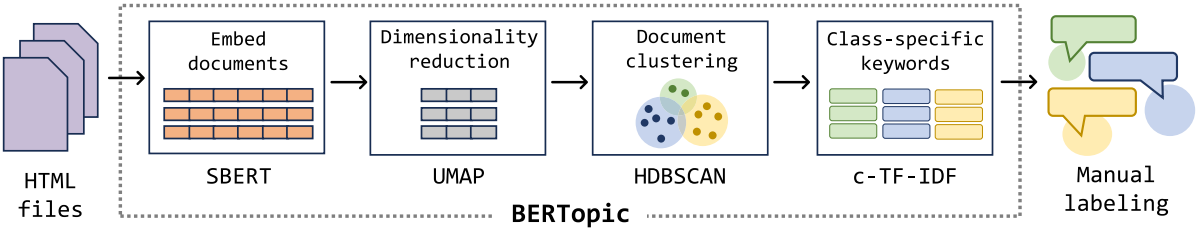

Fig. 2. BERTopic methodology to classify onion preprocessed HTML files.

3.3. Daily batch processing

The pipeline of this layer is automated and virtualized using Kubeflow Pipelines SDK v2 to dynamically deploy and execute batch processing at the end of each day, enabling transparent coordination and data sharing between the different phases.

With this approach, resources are not reserved throughout the day but are automatically demanded only for a limited period to build up the pipeline, analyze the new onions, and destroy the associated containers when finished. The following are the steps involved in the daily batch processing:

3.3.1. Pull new onion sites

The first action is to retrieve HTML documents from the *onions* bucket of the MinIO store by date. The HTML is parsed with BeautifulSoup, and the valuable part is extracted with Trafilatura [50], the framework that gave us the most refined results.

Automatically extracting meaningful text from raw HTML content and discarding tags, metadata, descriptions, footers, or style code, is critical particularly in the context of the dark web. Unlike the standard surface pages that adhere to well-established best practices, dark sites exhibit a less sophisticated implementation, limiting parsing frameworks' performance. It is worth mentioning that some dark pages are bad-encoded or empty.

3.3.2. Preprocessing of textual content

The meaningful text is preprocessed using the NLTK sentence tokenizer to remove worthless text, discarding onion addresses, PGP keys, email addresses, monetary values, URLs, bitcoin addresses, and HTML tags. Line breaks, tabulations, special characters, contiguous duplicated characters, words, and sentences are also removed.

By using document embeddings, there is no need to do more NLP preprocessing to understand the general topic of the document. The result is a cleaned and compacted text.

3.3.3. Efficient deduplication of onion sites

The preprocessed pages are deduplicated based on document similarity in the absolute dataset. Firstly, exact duplicates are identified through the identical matching of texts. For near deduplication of the rest of the documents, MinHash LSH [51] is adopted to map the set into smaller buckets to reduce the number of comparisons to perform. In particular, Hash signatures are calculated with MinHash algorithm for each document, being grouped via LSH. Finally, each bucket has a subset to compare, the content similarity of documents is performed with a similarity measurement.

After testing two of the most common measurements, Cosine Similarity and Jaccard Similarity [52], the selection of the latter with a threshold of 0.9 and 128 permutation functions emerged as the configuration for the best performance.

3.3.4. Language detection

The never-seen documents are analyzed to extract the language of the new onions using the language model[9] from Facebook AI's *fastext* framework, one of the best alternatives in terms of performance and scalability [53].

3.3.5. Topic extraction

To categorize English onion services, we employ BERTopic modeling [46], a technique that provides insights into the underlying themes present in a collection of documents. Fig. 2 illustrates the process.

The first step involves leveraging the SBERT [54] multilingual model, specifically the *paraphrase-multilingual-MiniLM-L12-v2* variant. This model transforms documents into a high-dimensional subspace, capturing semantic relationships and contextual information. Subsequently, UMAP [55] is utilized to reduce the dimensionality of the embedded documents. This step is crucial for visualizing and understanding the data in a more manageable space.

For document clustering, we employ HDBSCAN [56], a hierarchical density-based clustering algorithm. This algorithm not only identifies clusters but also allows for the detection of outliers, enhancing the robustness of the categorization process. In order to extract meaningful topic keywords within each cluster, we turn to c-TF-IDF [57]. This technique emphasizes terms that are not only frequent within a specific document but also discriminative across the entire corpus, providing a more nuanced representation of topics.

The resulting clusters are manually labeled with categories by the authors, providing a human-in-the-loop validation of the algorithm's outputs. This step adds a layer of interpretability and ensures that the generated categories align with real-world semantic distinctions.

In the Kubeflow pipeline, each phase discussed above is deployed on a Kubernetes Pod to carry out the programmed task and pass the output to the next component, but also versions the newly processed data on MinIO *datasets* bucket. Therefore, each component (i) generates the daily dataset, statistical plots, and metrics, and (ii) merges the new dataset with the updated general dataset with global statistical plots and metrics.[10] The Pods are automatically removed when associated activities have finished.

3.4. Data storage

This layer is responsible for persistently storing the data that results from the architecture. The system uses two non-relational MongoDB databases for ingestion and MinIO, a high-performance Kubernetes-native object storage, for processing.

[9] lid.176.bin

[10] Datasets are modeled with Kubeflow Output Artifacts, plots with Kubeflow Output HTML, metrics with Kubeflow Output Metrics.

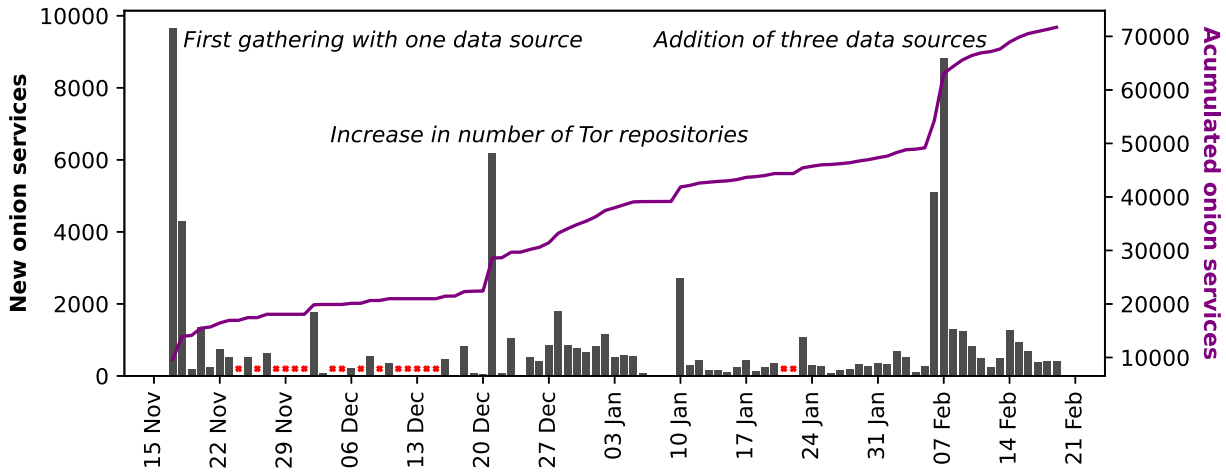

Fig. 3. Identification and analysis of new onion services.

3.4.1. Crawler database

This database stores the data related to crawling functions, configurations, and results. Specifically, each of the four source-based spiders saves its findings in a separate collection, including the *onion address* (a unique index), *advertiser*, and *identification timestamp*.

The four collections are configured with the MongoDB Kafka Source Connector to raise events with new insertions and feed the downloaders.

3.4.2. Downloader database

This database maintains the details to manage the download of the HTML files such as the *onion address*, *downloaded* (whether the onion has been downloaded), *downloaded timestamp* (date when the onion was downloaded), and four booleans for each data source indicating where the onion has been identified.

3.4.3. MinIO object store

This component is used to store various raw artifacts accessed throughout the architecture. The *onions* bucket contains every downloaded HTML page, and the *datasets* bucket keeps the incremental versions of the preprocessed, deduplicated, languages, and topic datasets generated by the daily pipeline. Both buckets index objects by date (*yyyy/mm/dd*) for easy access by batch processing. On the other hand, the *models* bucket stores the language and topic models to be used.

Notably, all this persistence through copies in MinIO allows replication of the pipeline from any point in time in the occurrence of a catastrophic event.

3.5. Data serving

The system provides a comprehensive suite of metrics and visualizations covering daily and global operations, all thanks to the user-friendly Kubeflow visual interface. Additionally, the architecture exposes the MinIO instance that enables data scientists to perform potential complex operations easily.

4. Experimental results

The architecture is implemented on a physical server with four virtual machines, all integrated with a self-hosted Kubernetes cluster (one master node and three worker nodes). The storage is provisioned with NFS to ensure reliable data storage and retrieval.

In addition to the core components, the system incorporates management with Kubernetes Dashboard, continuous integration and deployment (CI/CD) with GitLab Agent and Runners, and node resource monitoring with Prometheus and Grafana. In total, the system runs 66 pods to ensure efficient and seamless operation.

4.1. Ingestion and preprocessing of onion services

4.1.1. Ingestion

Our solution operated from November 17, 2022, to February 19, 2023,[11] and uncovered a total of 72,045 active Tor onion sites, as shown in Fig. 3 In the first month of execution, we used only one resource as a data source (Ahmia), which resulted in days without updates and an average rate of 659.4 onions/day. On December 21, we added the rest of the listings of the Tor repository source, which increased the rate to 816.7 onions/day. On February 6 and 7, we included the other three types of data sources (threat intelligence, code repositories, and Web-Tor gateways), which doubled the gathering to 1608.9 onions/day.

Table 5 shows the number of identified sites per data source, with Tor repositories being the most successful, with 71,742 active onions discovered (771.4 per day). Interestingly, 303 onions are identified in alternative data sources, with 29 uniquely discovered by threat intelligence, 153 in code repositories, and 65 through Web-Tor gateways.

4.1.2. Preprocessing

The raw dataset contains 72,045 HTML documents. The platform has identified 153 bad-encoded documents and 149 empty onions along the daily batches. The preprocessing step causes 39 new HTML files to be empty as well.

The final preprocessed sample consists of 71,704 documents, which are exactly deduplicated with perfect hash matching and near deduplicated with MinHash LSH. In particular, there are 56,674 exact duplicates (78.7%) and 10,640 near duplicates (14.8%),[12] reaching a total sum of 67,314 duplicates (93.5%). The deduplicated dataset contains 4,390 unique onion services, a 6.1% of the dataset.

4.2. Extraction of languages and topics

4.2.1. Language distribution

The pipeline is configured to extract the predominant language of each unique document. More than 30 different languages were detected, and the five most predominant among the 4,390 sites were the following:

- English: 3,869 (88.1%)

[11] Except for January 21 and 22, due to technical problems in the deployment.

[12] Related to the efficiency of MinHash LSH, the near deduplication of onions processed last day considering 70 K processed documents took around 5 s.

Table 5
Ingestion per data source and preprocessing stats.

Source	Days	Total onions	Active onions
Threat Int.	13	952	296 (31.1%)
Code repos.	13	1,741	819 (47%)
Gateways	13	759	677 (89.2%)
Tor repos.	93	78,656	71,742 (91.2%)
Total	93	80,049	**72,045** (90%)
Bad/Empty	Exact dup.	Near dup.	Unique
341 (0.4%)	56,674 (78.7%)	10,640 (14.8%)	4,390 (6.1%)

- Russian: 126 (2.9%)
- French: 84 (1.9%)
- German: 61 (1.4%)
- Spanish: 50 (1.1%)

The distribution changes with the propagation of language to the dataset of 71,704 documents with duplicates:

- English: 69,499 (96.9%)
- German: 1,060 (1.5%)
- Romanian: 354 (0.5%)
- Russian: 220 (0.3%)
- Spanish: 157 (0.2%)

Given the representativeness of the English language and the fact that BERTopic works best with it [46], the architecture applies the topic model only to English documents.[13]

4.2.2. Low-level topics

The BERTopic process groups the 3,869 English unique onions in 35 different clusters, as shown in Table 6, together with the words extracted from the documents under class-based TF-IDF (c-TF-IDF) and the label manually placed. The set of keywords may not be representative enough of each grouping, so the authors do the labeling manually based on the keywords, documents, and embedding distances.

As a result, the top-5 most common are sexual content (1,086), repositories and search engines (877), carding (394), cryptocurrencies (369), and hacking (169). In general, this indicates the predominance of apparently illicit buying and selling sites.

On the other hand, Fig. 4 illustrates the hierarchical structure of topics based on the cosine distance between their embeddings. The cosine distance is calculated by measuring the cosine of the angle between two vectors. Specifically, for each pair of topic embeddings, the cosine distance is computed as the dot product of the vectors divided by the product of their magnitudes.

While alternative distance metrics, such as Euclidean distance and Jaccard similarity, were considered, the cosine distance emerged as the preferred choice for our task. Euclidean distance measures the straight-line distance between two points in space, which may be sensitive to the magnitude of the vectors and less effective in capturing semantic similarity. Jaccard similarity, on the other hand, focuses on the intersection and union of sets, which might not fully capture the nuances of semantic relationships.

The cosine distance, in contrast, emphasizes the directional similarity between vectors, making it robust to variations in document lengths. This characteristic is particularly advantageous in our context, where documents may differ in terms of length but still convey similar semantic content. Consequently, the choice of cosine distance enhances the reliability of our hierarchical topic structure, providing a more accurate representation of the semantic relationships between different categories.

In the tree of Fig. 4, related topics tend to be grouped and have smaller cosine distances, as indicated by the colors. For example, there is a closeness between topics related to (i) credit cards or documents, (ii) hacking and data leaks, (iii) sexual and violent content, or (iv) marketplaces, hiring services or crypto services.

4.2.3. High-level topics

Considering the cosine distances between topic embeddings and author's expertise, the fine-grained clusters are manually merged into fewer groups to facilitate the analysis. As shown in Table 7, 11 high-level topics raise from the 35 low-level categories:

1. Sexual and violent content: *sexual content*, and *violent content*.
2. Repositories and search engines: *repositories and search engines*.
3. Carding: *CVV marketplaces*, *card dumps and fullz*, *banking*, and *carding*.
4. Cryptocurrencies: *cryptocurrencies*, and *crypto swapping and exchanges*.
5. Marketplaces: *hardware*, *mobile and device*, *drug*, *firearm*, and *generic marketplaces*.
6. Media, forums and personal websites: *media*, *personal websites and blogs*, *news and media*, *debian community*, and *debian conferences*.
7. Navigation pages: *javascript*, *login and register*, *DDoS protection*, *redirecting*, *error*, *one-line*, and *'index of' pages*.
8. Hacking: *data leaks*, and *hacking*.
9. Counterfeits: *passports and certificates*, and *counterfeits*.
10. Privacy-preserving services: *image and file hosting*, and *privacy-preserving services*.
11. Hiring services: *betting*, *hitman services*, and *escrow services*.

From this point, we focus on high-level topics, whose distribution both in the unique dataset and total dataset with duplicates is also presented in Table 7. Considering the 3,869 unique sites without mirrors, the top-5 high-level topics sum 79.6%:

1. Sexual and violent content: 1,104 (28.5%)
2. Repositories and search engines: 877 (22.7%)
3. Carding: 463 (12%)
4. Cryptocurrencies: 392 (10.1%)
5. Marketplaces: 245 (6.3%)

The rest of the topics are media, forums and personal websites, navigation pages, hacking, counterfeits, privacy-preserving services and hiring services.

However, if we consider the replication of onion services, the total distribution of topics becomes very unbalanced, increasing even more the presence in the dark web of the first five topics to 95,3%:

1. Sexual and violent content: 33,530 (48.2%)
2. Repositories and search engines: 19,098 (27.5%)
3. Carding: 6,912 (9.9%)
4. Cryptocurrencies: 4,829 (6.9%)
5. Marketplaces: 1,859 (2.7%)

These types of onion services exhibit a significant mirroring, particularly 29.37, 20.78, 13.93, 11.32 and 6.59 duplicates per single onion, respectively.

The architecture identifies new unique Tor sites for each theme over time. Fig. 5 displays the cumulative discovery of new onion services per day, with two significant peaks on 21 December and 6–7 February, corresponding to the addition of new monitoring sources in the architecture (noted in Fig. 3). As duplicates of previously seen onion services are not included in the graph, it highlights that new Tor pages are being created daily. The trend is uniform across each

[13] A multilingual sentence-transformer is used due to alternative languages present in English documents.

Table 6
Low-level topics of unique onion sample.

Cluster	Onions	Ratio	Top-5 class-specific words	Manual low-level topic label
0	1,086	28.1%	porn, video, sex, teen, girl	Sexual content
1	877	22.7%	csv, uniquevoteshashes, distribution, git, diff	Repositories and search engines
2	394	10.2%	card, covid, transfer, money, paypal	Carding
3	369	9.5%	bitcoin, ethereum, monero, crypto, tether	Cryptocurrencies
4	169	4.4%	hacking, whatsapp, phone, hire, hackers	Hacking
5	136	3.5%	cocaine, drugs, quality, shipping, lsd	Drug marketplaces
6	82	2.1%	returns, patch, mcdonald, human, ronald	Personal websites and blogs
7	76	2%	male, coconut, news, read, oil	News and media
8	47	1.2%	guns, rifle, magazine, gun, pistol	Firearm marketplaces
9	44	1.1%	download, upload, file, image, mins	Image and file hosting
10	43	1.1%	gz, tar, live, jan, sqxz	'Index of' pages
11	40	1%	nitter, irg, vehicles, rcmp, totoro	One-line pages
12	37	1%	counterfeit, money, banknotes, bills, fake	Counterfeits
13	35	0.9%	documents, passport, fake, license, certificate	Passports and certificates
14	34	0.9%	login, javascript, password, sign, register	Javascript pages
15	32	0.8%	error, nginx, server, forbidden, var	Error pages
16	31	0.8%	dumps, pin, card, track, fullz	Card dumps and fullz
17	31	0.8%	iphone, apple, delivery, phone, samsung	Mobile and device marketplaces
18	27	0.7%	debian, backports, ports, translations, key	Debian community
19	26	0.7%	hitman, killer, hire, person, job	Hitman services
20	24	0.6%	escrow, seller, buyer, dispute, transaction	Escrow services
21	23	0.6%	token, finance, coin, protocol, network	Crypto swapping and exchanges
22	22	0.6%	bank, logs, transfer, account, money	Banking
23	21	0.5%	privacy, cwtch, security, server, metadata	Privacy-preserving services
24	20	0.5%	rog, gaming, performance, strix, zephyrus	Hardware marketplaces
25	19	0.5%	dpp, login, admin, password, png	Login and register pages
26	18	0.5%	debian, debconf, conference, info, talks	Debian conferences
27	18	0.5%	incest, eddie, gallery, mindy, tiger	Violent content
28	16	0.4%	cvv, dumps, fullz, bases, bazar	CVV marketplaces
29	15	0.4%	redirected, queue, wait, refresh, automatically	Redirecting pages
30	13	0.3%	ddos, mirror, anti, protection, bank	DDoS protection pages
31	11	0.3%	matches, fixed, cup, fifa, betting	Betting services
32	11	0.3%	instagram, hack, facebook, database, accounts	Data leaks
33	11	0.3%	post, topics, view, neneh, subforums	Forums
34	11	0.3%	ammo, min, read, collection, backup	Generic markets
Total	3,869	100		

topic, indicating that the Tor network gradually incorporates new and distinct content over time, making it a living space where existing onion services are not only disappearing and getting replicated.

4.3. Exploration of duplicates

4.3.1. Most replicated sites

Based on the disproportionate amount of duplicates discovered by the architecture, we study which particular onion services were the most replicated. Fig. 6 shows the duplicates over time of the top-20 most replicated sites (along with the page title and inferred topic), which sums 26,042 instances out of 69,499 Tor sites (37.47%).

In this ranking, 12 are related to carding, 3 with sexual and violent content, 3 with cryptocurrencies, 1 with hacking and 1 with repositories and search engines. The "*Prepaid Debit Card buy*" has more than 4,000 replicas, the "*Paypal Account*" and "*Porn Movie*" pages have more than 2,000 replicas, and the rest ranges between 800 and 1000 replicas.

At the bottom of the line plot, a set of sites share the same temporal pattern of duplicate generation, a suspicious phenomenon that attracts the authors' attention for further inspection.

4.3.2. Duplicated sites with similar replication ratio

Focusing on websites with similar replication trends, Fig. 7 details the detection of duplicate pages per onion site over time. Each unique 14 sites were manually verified and found to have completely different content. However, except for the Torch repository, all sites were related to topics such as credit cards, bitcoin, gift cards, cryptocurrencies, wallets, and money, and classified by the solution under the categories of carding and cryptocurrencies.

The ratios of duplicate pages per day were very close, where the main difference lies in the initial offset caused by the number of onions detected on the first day. On average, seven pages generated between 8 and 9 duplicates per day, five pages created 11 replicas per day, one mirrored 12 times per day, and the first one was replicated 15 times per day.

Although we have no more evidence, these figures are highly coincidental, and we hypothesize that a common actor coordinates these different pages (and duplicates). Specifically, the replication may not be intended to maximize the availability or protect against DDoS attacks. Instead, it could be a massive phishing campaign through different types of sites that contain fraud cryptocurrency addresses as attack vectors, increasing the attack surface by replicating them across the Tor network. For example, "*Light Money*" and "*Money Transfers*" are two of these services that belong to different topics but coincide in the number of duplicates (796) and replication ratio (8.47 duplicates/day), showing an identical trend throughout the collection window. Moreover, the Torch repositories could be deployed to reference the illegitimate pages and amplify the threat landscape.

Generally, phishing is common and effective on the Tor network due to the anonymity provided, being exploited by cybercriminals that evade detection through fake pages that cannot be authenticated.

5. Discussion

The literature presents several approaches for analyzing Tor sites but needs to be more in adopting modern data ingestion, processing, and storage frameworks. Solutions are limited in scalability and efficacy, leading to a shortage of collections from 529 to 10,163 sites. In order to address this gap, this paper proposes an Open Source Big Data architecture for identifying and categorizing onion services in near real-time, representing unprecedented performance in Tor monitoring with 72 K analyzed onion services.

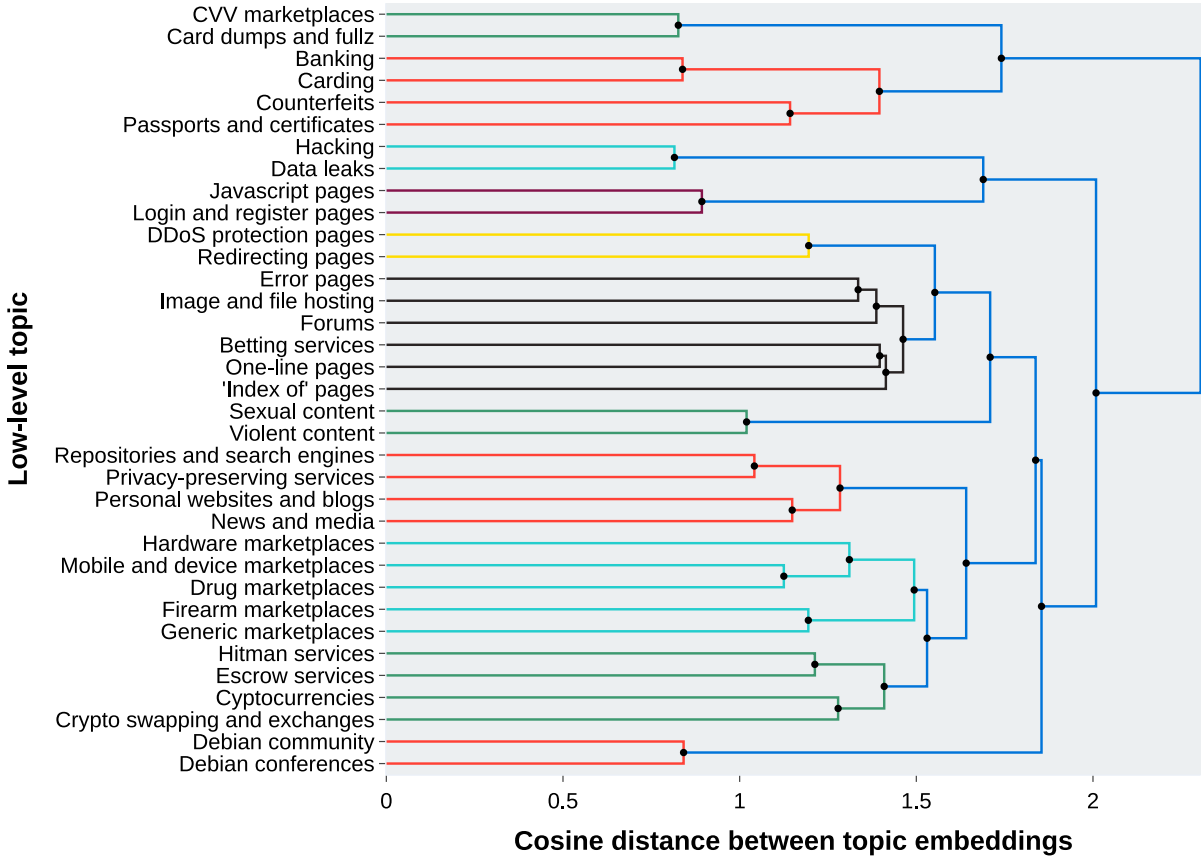

Fig. 4. Hierarchical clustering based on the cosine distance matrix between topic embeddings.

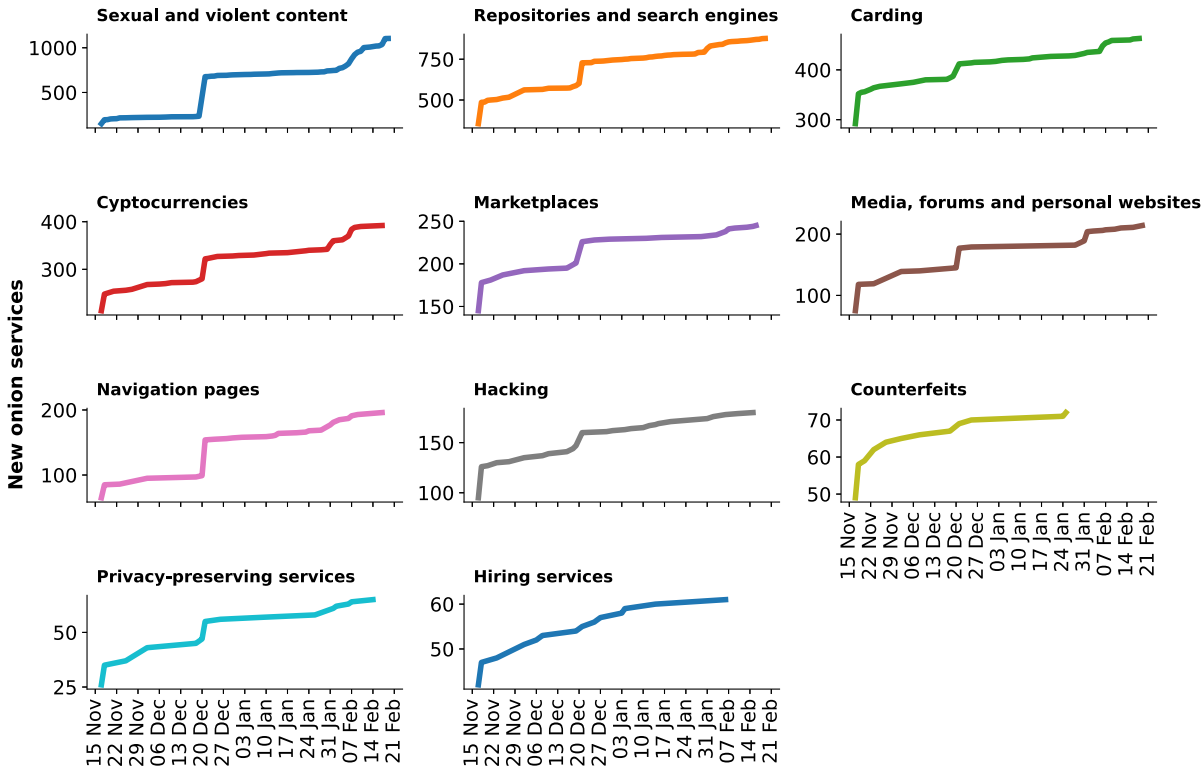

Fig. 5. Accumulated identification of unique onion services per high-level topic.

Table 7
Onion services distribution per high-level topic.

High-level topic	Low-level topics	Uniques	Mirrors	Total
Sexual and violent content	Sexual content Violent content	1,104 (28.5%)	32,426	33,530 (48.2%)
Repositories and search engines	Repositories and search engines	877 (22.7%)	18,221	19,098 (27.5%)
Carding	CVV marketplaces Card dumps and fullz Banking Carding	463 (12%)	6,449	6,912 (9.9%)
Cryptocurrencies	Cryptocurrencies Crypto swapping and exchanges	392 (10.1%)	4,437	4,829 (6.9%)
Marketplaces	Hardware marketplaces Mobile and device marketplaces Drug marketplaces Firearm marketplaces Generic markets	245 (6.3%)	1,614	1,859 (2.7%)
Media, forums and personal websites	Personal websites and blogs News and media Debian community Debian conferences	214 (5.5%)	1,491	1,705 (2.5%)
Navigation pages	Javascript pages Login and register pages DDoS protection pages Redirecting pages Error pages One-line pages 'Index of' pages	196 (5%)	330	526 (0.8%)
Hacking	Data leaks Hacking	180 (4.7%)	332	512 (0.7%)
Counterfeits	Passports and certificates Counterfeits	72 (1.9%)	195	267 (0.4%)
Privacy-preserving services	Image and file hosting Privacy-preserving services	65 (1.7%)	79	144 (0.2%)
Hiring services	Betting services Hitman services Escrow services	61 (1.6%)	56	117 (0.2%)
Total		3,869 (100%)	65,630	69,499 (100%)

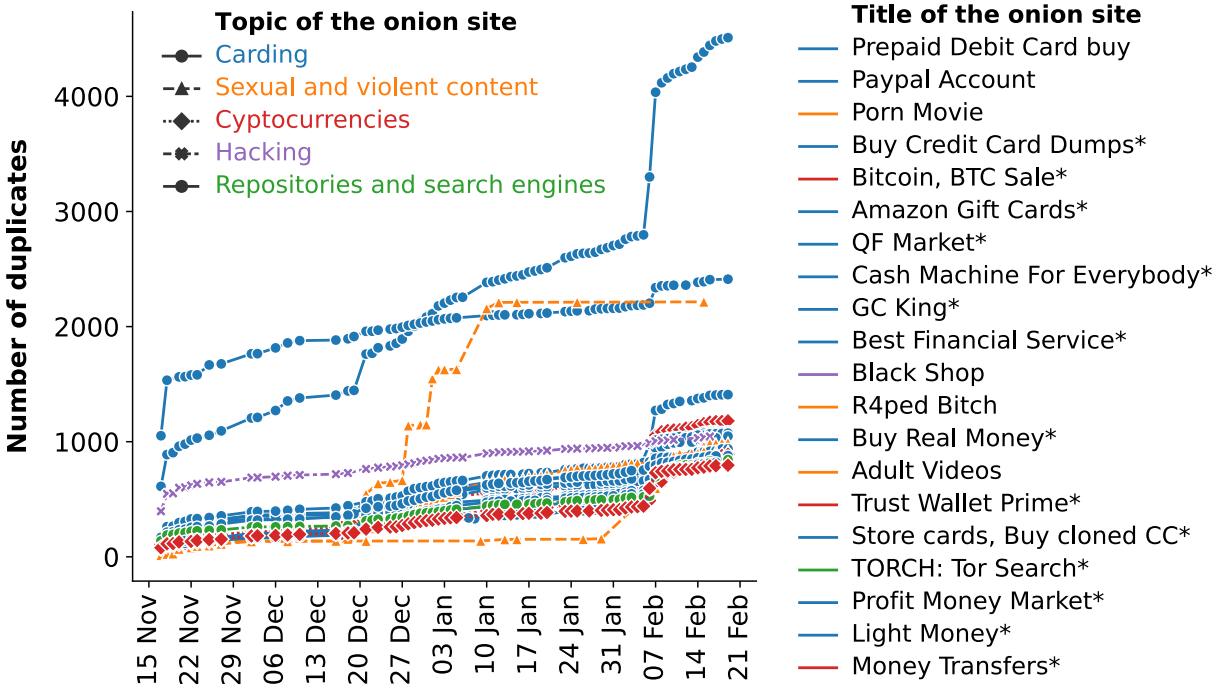

Fig. 6. Duplicates of the top-20 most replicated onion services over time (asterisks indicate onions with similar trend).

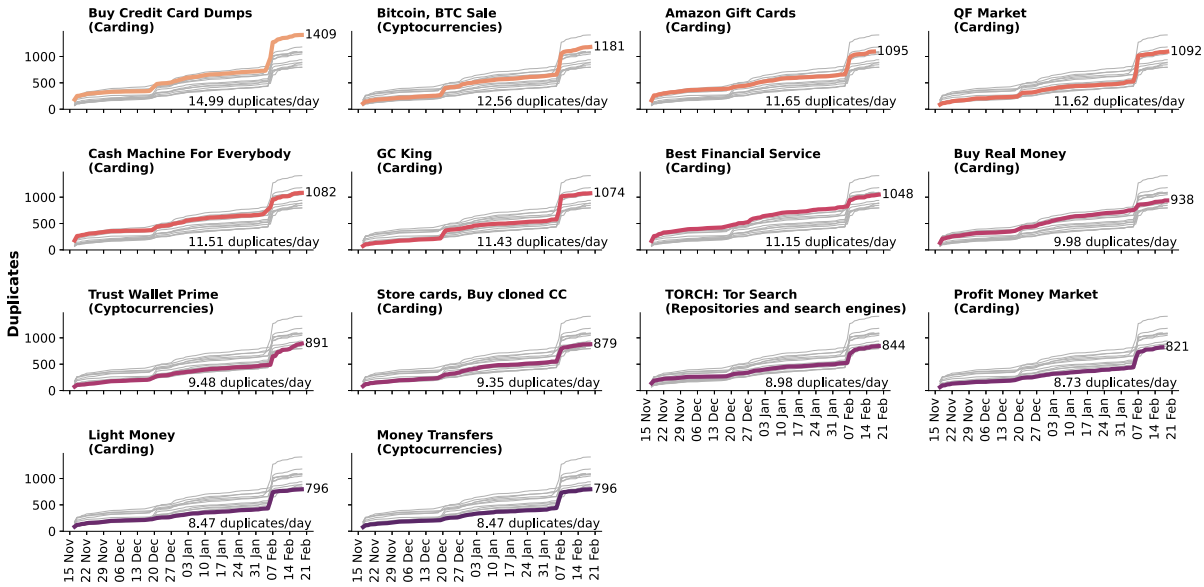

Fig. 7. Duplicates of 14 onion services with a similar replication pattern.

Kubernetes is a key technological enabler that facilitates a direct deployment of Kafka or integration of native complements such as MinIO and Kubeflow. Additionally, the resources of the computing nodes can be managed efficiently. In this sense, it guarantees that the continuous monitoring part is always active to detect and download the content of the onions. On the other hand, the batch processing task is punctual and dynamically launched in a finite way to characterize the sample collected every day, having most of the time those resources available in the cluster. Otherwise, the latter would be exclusively dedicated to processing and machine learning processes in streaming.

However, developing an end-to-end pipeline from scratch for community-driven projects posed significant challenges due to gaps and technical issues. For instance, modifying low-level components like the MongoDB source connector for Kafka or Crawlab spiders was complex. Additionally, while the orchestration and deployment of microservices are theoretically simple in virtualized frameworks, integrating and communicating all the distributed elements into a common logic is really hard. Therefore, from September 1 to November 17, we spent more than two months to have the architecture up and running without inconsistencies and errors through GitLab CI/CD operations within the cluster. The result is a sophisticated platform, but not overly romanticized against monolithic architectures that can be much easier to develop for most use cases. Despite these obstacles, adopting DevOps (GitLab CI/CD with Kubernetes) and MLOps (Kubeflow) allowed us to define operations by software, integrating operation and development into one workflow.

Regarding the challenges of the Tor use case, the volatility and short life of onion services make it difficult to implement tools for monitoring the Tor network landscape. In the related works, the discovered onion services showed a low availability (from 3% to 35%), having a big discrepancy between the addresses identified and those active and analyzed. This paper addresses these coverage and access problems with early identification under continuous monitoring. By adopting the strategy in the pipeline, the categorization is executed before onion services cease to exist, having reached 72,045 onion sites out of 80,330 (90% of availability), the second most representative research of the literature [9] (after 82,145 active sites considered by [12]).

Another critical aspect faced is extracting meaningful and valuable text from raw HTML pages containing too much noise. Particularly, dark web sites are not well-formed and break the HTML best practices, which is different from standard surface sites. While the study of surface webpages is widely more common in the literature, dark web research is a bit retarded in the application of modern NLP strategies. Downloading UTF-8, parsing with BautifulSoup, extracting relevant text with Trafilatura and application of regular expressions to remove noise (addresses, links, weird strings and characters, etc.) was hard in order to get a good performance in preprocessing messy dark HTML files.

Some studies demonstrate a substantial prevalence of duplicated dark web sites, from 20% to 95%, yet the majority of analytical experiments fail to account for this phenomenon, leading to an overload of processing the same content multiple times, but also generating distorted research results and distributions. In order to address this limitation, our proposed framework integrates content similarity with Jaccard distance to analyze only new never-seen documents, and MinHash LSH to increase scalability due to the reduction of document comparisons to perform. In our case, we found 93.9% of repeated content, reducing unnecessary processing overhead and being in line with literature results.

Regarding categorization, related works rely on manual, keyword, or probabilistic-based models, with a noticeable absence of modern topic extraction or document classification based on neural networks or embeddings. Additionally, the sample sizes used in previous classifications, from 445 to 6,227 dark sites, may not be sufficient to capture the topic diversity of the Tor network. In this work, we apply BERTopic to analyze 72,045 onion services, marking the largest categorization effort among the reviewed works. This framework has shown promising results in various domains [47,48], and our study serves as another example.

It is worth noting that we identified several different types of navigation pages, such as error, index-of, DDoS protection, register and logging, and redirecting, and javascript. Other findings include file hosting, privacy services, Debian communities, and Debian conference pages. In terms of defense, some were directly related to cyber warfare (hacking), data leaks, propaganda (news and media), weapons trading (firearms), or organized communities (forums). However, considering the high percentage of mirrored sites within the dark web, experimental results indicate a 95% probability of encountering services or repositories associated with illicit trading, such as sexual and violent content,

cards, cryptocurrencies, or marketplaces. Conversely, potentially ethical resources, such as forums, websites, media, or privacy services, are limited to approximately 2.5%.

Finally, one might argue that dark web monitoring does not fit squarely within the Big Data paradigm, given the existence of around 800 thousand onions simultaneously, according to Tor metrics.[14] While this scenario may seem limited due to the slow influx of new onions daily, the modular and virtualized architecture allows for deploying more advanced Tor use cases. Examples include daily downloading and categorizing all onion samples to track changes or extending real-time ingestion to new darknets like I2P, Freenet, ZeroNet, or even the surface web. In our project, the design is sufficiently generic and extensible to support various types of workloads, from constrained use cases for easy deployment and management of the application to high-demanding scenarios requiring scalability, robustness, and adaptability.

6. Conclusion

In this paper, we propose an architecture for continuously obtaining and analyzing Tor onion services detected in diverse data sources, including threat intelligence, code repositories, Web-Tor gateways, and Tor repositories. We include a near real-time ingestion with a crawler to visit the data sources and use Kafka to coordinate the download of HTML pages by a set of downloaders. At the end of the day, a batch processing pipeline based on Kubeflow preprocesses the HTML content, deduplicates with MinHash LSH, and extracts languages with fasttext while inferring topics with BERTopic. The architecture is deployed under DevOps (GitLab CI/CD) and MLOps (Kubeflow) paradigms, easily monitored and configured with the Kubernetes Dashboard (data engineering), Crawlab user-friendly framework (data ingestion) and Kubeflow interface (data processing).

The solution has been running for 93 days, categorizing 72,045 onion services out of 80,049 identified. In this sense, early identification of sites using the crawler's scheduled spiders is highly effective, enabling the solution to connect and characterize 90% of the Tor sites. This is a significant improvement from the volatility ratios reported in related works. The architecture identified 56,674 (78.7%) exact duplicates (hash matching) and 10,640 (14.8%) near duplicates (MinHash LSH), having only 4,390 (6.1%) unique sites. This disproportional amount of repeated content is even higher than in other literature reviews and exposes most Tor-based studies that do not consider this phenomenon in their interpretations and results.

The predominant language is English, with 88.09% in the unique sample and 96,9% in the overall dataset with duplicates. The BERTopic methodology is applied to this subset of documents, returning 35 low-level topics, which the authors manually merged into 11 considering the cosine distance between topic embeddings: sexual and violent content; repositories and search engines; carding; cryptocurrencies; marketplaces; media, forums and personal websites; navigation pages; hacking; counterfeits; privacy-preserving services; and hiring services (in order of prevalence). Over the days, the appearance of onions in each topic has a similar pattern and does not show significant peaks or anomalies.

In the exploration of duplicates, we specifically studied the clone distribution of the 20 most replicated onion services (37.47% of the dataset), discovering a subset of 14 Tor sites of different nature that maintain an almost identical daily appearance pattern. While the detailed study of these remains as future work, it seems to us that these duplicates of different onion services would not be focused on mirroring or protection against DDoS attacks, but part of a coordinated phishing network that automatically multiplies its exposure on the dark web through different types of content and appearance.

[14] https://metrics.torproject.org/hidserv-dir-v3-onions-seen.html

7. Future work

Our proposed architecture, in its current state, has proven fruitful in aggregating and characterizing Tor onion services. Nonetheless, several promising directions for future work remain to be explored.

A prevalent observation from our study is the existence of a substantial number of duplicated dark websites. The development of advanced techniques to discriminate between genuine and phishing sites is necessary to be integrated into the ML pipeline. Such an advancement would not only enhance data accuracy but also fortify subsequent analyses' reliability. Additionally, potential advancements could stem from further enhancing the categorization process. The exploration of alternative topic modeling algorithms and comparison of their efficacy with BERTopic could yield interesting insights. Additionally, considering the global nature of Tor networks, broadening the scope of our analysis to include non-English onion services could shed light on region-specific behaviors and trends, offering a more holistic understanding of the Tor landscape.

Beyond Tor services, exploring other anonymous spaces, including a wider variety of darknets like I2P, Freenet or ZeroNet, promises a more encompassing interpretation of the dark web sphere. Concurrently, our architecture's scalability and robustness should undergo additional rigorous testing with larger data volumes and extended operational periods. These steps would accurately assess our architecture's ability to handle larger datasets and lead to the developing of more resilient systems.

Finally, the proposed platform is independent of its current Tor-based application. We foresee its potential in exploring alternative Open Source Intelligence (OSINT) use cases, from tracking cyber threats to monitoring illicit activities within the darknet. In particular, the platform could be extended with a live dashboard exhibiting real-time updates and thorough statistical analyses would significantly enhance the interpretability and accessibility of the data. Such a tool could provide invaluable insights for researchers and law enforcement agencies.

CRediT authorship contribution statement

Javier Pastor-Galindo: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Hông-Ân Sandlin:** Conceptualization, Investigation, Methodology, Supervision, Validation, Visualization, Writing – original draft. **Félix Gómez Mármol:** Conceptualization, Investigation, Supervision. **Gérôme Bovet:** Funding acquisition, Project administration, Supervision. **Gregorio Martínez Pérez:** Funding acquisition, Project administration, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is available at Mendeley data: https://doi.org/10.17632/9npmf5v6kr.1.

Acknowledgments

This study was partially funded by (a) the Spanish Government with the FPU18/00304 contract and EST22/00738 mobility grant, and by *(b)* the strategic project "Development of Professionals and Researchers in Cybersecurity, Cyberdefense and Data Science (CDL-TALENTUM)" from the Spanish National Institute of Cybersecurity (INCIBE) and by the Recovery, Transformation and Resilience Plan, Next Generation EU. We would also like to thank all the colleagues from the Cyber-Defence campus office in Lausanne for their kind support during this internship project.

References

[1] J. Pastor-Galindo, P. Nespoli, F. Gómez Mármol, G. Martínez Pérez, The not yet exploited goldmine of OSINT: Opportunities, open challenges and future trends, IEEE Access 8 (2020) 10282–10304.

[2] M. Willett, The cyber dimension of the Russia–Ukraine war, Survival 64 (5) (2022) 7–26.

[3] D.L. Huete Trujillo, A. Ruiz-Martínez, Tor hidden services: A systematic literature review, J. Cybersecur. Priv. 1 (3) (2021) 496–518.

[4] J.M. Ruiz Ródenas, J. Pastor-Galindo, F. Gómez Mármol, A general and modular framework for dark web analysis, Cluster Comput. (2023) 1–17.

[5] J. Pastor-Galindo, R. Sáez Ruiz, J. Maestre Vidal, M. Sotelo Monge, F. Gómez Mármol, G. Martínez Pérez, Designing a platform for discovering TOR onion services, in: 7th National Conference on Cybersecurity Research, JNIC 2022, Bilbao, Spain, 2022.

[6] A. Buitrago López, J. Pastor-Galindo, F. Gómez Mármol, Updated exploration of the Tor network: advertising, availability and protocols of onion services, Wireless Netw. (2024) 1–15.

[7] G. Owenson, S. Cortes, A. Lewman, The darknet's smaller than we thought: The life cycle of Tor Hidden Services, Digit. Investig. 27 (2018) 17–22.

[8] F. Platzer, A. Lux, A synopsis of critical aspects for darknet research, in: ACM International Conference Proceeding Series, (no. 20) 2022.

[9] J. Pastor-Galindo, F. Gómez Mármol, G. Martínez Pérez, On the gathering of Tor onion addresses, Future Gener. Comput. Syst. 145 (2023) 12–26.

[10] C. Yoon, K. Kim, Y. Kim, S. Shin, S. Son, Doppelgängers on the dark web: A large-scale assessment on phishing hidden web services, in: The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019, 2019, pp. 2225–2235.

[11] P. Burda, C. Boot, L. Allodi, Characterizing the redundancy of DarkWeb .Onion services, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES '19, Association for Computing Machinery, New York, NY, USA, 2019.

[12] M. Steinebach, M. Schäfer, A. Karakuz, K. Brandl, Y. Yannikos, Detection and analysis of Tor onion services, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES '19, Association for Computing Machinery, New York, NY, USA, 2019.

[13] D. Khurana, A. Koli, K. Khatter, S. Singh, Natural language processing: State of the art, current trends and challenges, Multimedia Tools Appl. 82 (3) (2023) 3713–3744.

[14] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Comput. Surv. 55 (9) (2023).

[15] A.T. Zulkarnine, R. Frank, B. Monk, J. Mitchell, G. Davies, Surfacing collaborated networks in dark web to find illicit and criminal content, in: 2016 IEEE Conference on Intelligence and Security Informatics, ISI, 2016, pp. 109–114.

[16] S. Ghosh, A. Das, P. Porras, V. Yegneswaran, A. Gehani, Automated categorization of onion sites for analyzing the darkweb ecosystem, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. Part F1296, 2017, pp. 1793–1802.

[17] A. Celestini, S. Guarino, Design, Implementation and Test of a Flexible Tor-Oriented Web Mining Toolkit, in: ACM International Conference Proceeding Series, vol. Part F1294, (no. August 2018) 2017.

[18] G. Cherubin, J. Hayes, M. Juarez, Website fingerprinting defenses at the application layer, Proc. Priv. Enhanc. Technol. 2017 (2) (2017) 186–203.

[19] J. Park, H. Mun, Y. Lee, Improving tor hidden service crawler performance, in: 2018 IEEE Conference on Dependable and Secure Computing, DSC, 2018, pp. 1–8.

[20] X. Zhang, K.P. Chow, A framework for dark web threat intelligence analysis, in: I.R.M. Association (Ed.), Cyber Warfare and Terrorism: Concepts, Methodologies, Tools, and Applications, IGI Global, Hershey, PA, USA, 2020, pp. 266–276.

[21] J. Lee, Y. Hong, H. Kwon, J. Hur, Shedding Light on Dark Korea: An In-Depth Analysis and Profiling of the Dark Web in Korea, in: I. You (Ed.), Information Security Applications, Springer International Publishing, Cham, 2020, pp. 357–369.

[22] S.M.M. Monterrubio, J.E.A. Naranjo, L.I.B. Lopez, A.L.V. Caraguay, Black Widow Crawler for TOR network to search for criminal patterns, ICI2ST 2021, Proceedings - 2021 2nd International Conference on Information Systems and Software Technologies (2021) 108–113.

[23] A.H.M. Alaidi, R.M. Alairaji, H.T.H.S. ALRikabi, I.A. Aljazaery, S.H. Abbood, Dark web illegal activities crawling and classifying using data mining techniques, Int. J. Interact. Mob. Technol. 16 (10) (2022) 122–139.

[24] M. Bernaschi, A. Celestini, S. Guarino, F. Lombardi, E. Mastrostefano, Spiders like Onions: On the network of tor hidden services, in: The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019, 2019, pp. 105–115.

[25] M. Spitters, S. Verbruggen, M. Van Staalduinen, Towards a comprehensive insight into the thematic organization of the Tor hidden services, in: 2014 IEEE Joint Intelligence and Security Informatics Conference, 2014, pp. 220–223.

[26] I. Sanchez-Rola, D. Balzarotti, I. Santos, The Onions Have Eyes: A Comprehensive Structure and Privacy Analysis of Tor Hidden Services, in: Proceedings of the 26th International Conference on World Wide Web, WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, pp. 1251–1260.

[27] M.W. Al Nabki, E. Fidalgo, E. Alegre, I. de Paz, Classifying illegal activities on tor network based on web textual contents, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 35–43.

[28] M.W. Al-Nabki, E. Fidalgo, E. Alegre, L. Fernández-Robles, ToRank: Identifying the most influential suspicious domains in the Tor network, Expert Syst. Appl. 123 (2019) 212–226.

[29] A. Kinder, K.-K.R. Choo, N.-A. Le-Khac, Towards an automated process to categorise Tor's hidden services, in: N.-A. Le-Khac, K.-K.R. Choo (Eds.), Cyber and Digital Forensic Investigations: A Law Enforcement Practitioner's Perspective, Springer International Publishing, Cham, 2020, pp. 221–246.

[30] M. Faizan, R.A. Khan, Exploring and analyzing the dark Web: A new alchemy, First Monday 24 (5) (2019).

[31] J. Dalins, C. Wilson, M. Carman, Criminal motivation on the dark web: A categorisation model for law enforcement, Digit. Investig. 24 (2018) 62–71.

[32] F. Barr-Smith, J. Wright, Phishing with a darknet: Imitation of onion services, in: 2020 APWG Symposium on Electronic Crime Research, ECrime, 2020, pp. 1–13.

[33] F. Brenner, F. Platzer, M. Steinebach, Discovery of single-vendor marketplace operators in the tor-network, in: Proceedings of the 16th International Conference on Availability, Reliability and Security, in: ARES 21, Association for Computing Machinery, New York, NY, USA, 2021.

[34] M. Steinebach, S. Zenglein, K. Brandl, Phishing detection on tor hidden services, Forensic Sci. Int. Digit. Investig. 36 (2021) 301117.

[35] C. Guitton, A review of the available content on Tor hidden services: The case against further development, Comput. Hum. Behav. 29 (6) (2013) 2805–2815.

[36] G. Owen, N. Savage, Empirical analysis of Tor hidden services, IET Inf. Secur. 10 (3) (2016) 113–118.

[37] V. Nair, J.M. Kannimoola, A Tool to Extract Onion Links from Tor Hidden Services and Identify Illegal Activities, in: S. Smys, V.E. Balas, R. Palanisamy (Eds.), Inventive Computation and Information Technologies, Springer Nature Singapore, Singapore, 2022, pp. 29–37.

[38] S. Takaaki, I. Atsuo, Dark Web Content Analysis and Visualization, in: Proceedings of the ACM International Workshop on Security and Privacy Analytics, IWSPA '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 53–59.

[39] M. Zabihimayvan, D. Doran, A first look at references from the dark to the surface web world: a case study in Tor, Int. J. Inf. Secur. 21 (4) (2022) 739–755.

[40] Y. Kawaguchi, S. Ozawa, Exploring and identifying malicious sites in dark web using machine learning, in: T. Gedeon, K.W. Wong, M. Lee (Eds.), Neural Information Processing, Springer International Publishing, Cham, 2019, pp. 319–327.

[41] A. Biryukov, I. Pustogarov, F. Thill, R.-P. Weinmann, Content and popularity analysis of tor hidden services, in: 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops, ICDCSW, 2014, pp. 188–193.

[42] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du, W. Buntine, Topic modelling meets deep neural networks: A survey, in: Z.-H. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 4713–4720, Survey Track.

[43] M. Shi, J. Liu, D. Zhou, M. Tang, B. Cao, WE-LDA: A word embeddings augmented LDA model for web services clustering, in: 2017 IEEE International Conference on Web Services, ICWS, 2017, pp. 9–16.

[44] A.B. Dieng, F.J.R. Ruiz, D.M. Blei, Topic modeling in embedding spaces, Trans. Assoc. Comput. Linguist. 8 (2020) 439–453.

[45] D. Angelov, Top2Vec: Distributed representations of topics, 2020, CoRR abs/2008.09470.

[46] M. Grootendorst, BERTopic: Neural topic modeling with a class-based TF-IDF procedure, 2022, arXiv preprint arXiv:2203.05794.

[47] H.W.A. Hanley, D. Kumar, Z. Durumeric, Happenstance: Utilizing semantic search to track Russian state media narratives about the russo-ukrainian war on reddit, Proc. Int. AAAI Conf. Web Soc. Media 17 (1) (2023) 327–338.

[48] R. Egger, J. Yu, A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts, Front. Sociol. 7 (2022).

[49] K. Li, P. Liu, Q. Tan, J. Shi, Y. Gao, X. Wang, Out-of-band discovery and evaluation for tor hidden services, in: Proceedings of the 31st Annual ACM Symposium on Applied Computing, SAC '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 2057–2062.

[50] A. Barbaresi, Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction, in: Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2021, pp. 122–131.

[51] A. Gionis, P. Indyk, R. Motwani, et al., Similarity search in high dimensions via hashing, in: Vldb, vol. 99, (no. 6) 1999, pp. 518–529.

[52] W.H. Gomaa, A.A. Fahmy, et al., A survey of text similarity approaches, Int. J. Comput. Appl. Technol. 68 (13) (2013) 13–18.
[53] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, 2016, arXiv preprint arXiv:1607.01759.
[54] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992.
[55] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I.W. Kwok, L.G. Ng, F. Ginhoux, E.W. Newell, Dimensionality reduction for visualizing single-cell data using UMAP, Nature Biotechnol. 37 (1) (2019) 38–44.
[56] L. McInnes, J. Healy, S. Astels, hdbscan: Hierarchical density based clustering, J. Open Source Softw. 2 (11) (2017) 205.
[57] A. Özgür, L. Özgür, T. Güngör, Text categorization with class-based and corpus-based keyword selection, in: Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26–28, 2005. Proceedings 20, Springer, 2005, pp. 606–615.

Javier Pastor-Galindo received his Ph.D. in Computer Science from the University of Murcia (Spain) in 2023, where he is currently postdoctoral researcher. His scientific interests focus on intelligence, cyberdefence, cybersecurity, data science and disinformation.

Hông-Ân Sandlin is senior scientific project manager at armasuisse Science & Technology. Her interests are focused on machine learning, data mining, data visualization, context awareness, IoT, big data, database systems and sustainable development. She received her Ph.D. from ETH Zûrich in 2015.

Félix Gómez Mármol received a M.Sc. and Ph.D. in Computer Science from the University of Murcia. He is currently Associate Professor in the Department of Information and Communications Engineering at the University of Murcia, Spain. His research interests include cybersecurity, internet of things, machine learning and bio-inspired algorithms.

Gérôme Bovet is the head of data science for the Swiss DoD, where he leads a research team and a portfolio of about 30 projects. His work focuses on machine/deep learning approaches applied to cyber-defense use cases, with emphasis on anomaly detection, adversarial and collaborative learning. He received his Ph.D. in networks and systems from Telecom ParisTech, France, in 2015.

Gregorio Martínez Pérez received a Ph.D. degree in Computer Science at the University of Murcia, where he is Full Professor since 2014. His scientific activity is mainly devoted to cybersecurity and data science. He is working on different national and European IST research projects related to these topics, being Principal Investigator for UMU in most of them.