The study of open source search engines, open source forensic tools and keyword search with respect to the needs of digital forensics

Joachim Hansen

NTNU
Norwegian University of
Science and Technology

Master's Thesis Project Description - IMT4601
NTNU in Gjøvik, 2017

Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology
NTNU in Gjøvik
PO box 191
NO-2802 Gjøvik, Norway

Abstract

Contents

Contents iii
List of Tables v
1 Introduction 1
1.1 Topic covered by the project (TODO consider rephrasing) 1
1.2 Keywords 1
1.3 Problem description (TODO reprase) 1
1.4 Justification, Motivation and benefits (TODO rephrase) 1
1.5 Research question (TODO rephrase the reseach questions) 2
1.6 Planned contributions (ask about feedback... I should maby add more in this section) 2
1.7 Outline 2
2 Related work 3
2.1 Systematic literature review 1 - Search in digital forensics and Open source search engines in the wild 3
2.1.1 Application/experimental use of search in digital forensic investigation 3
Collection phase 3
Examination phase 4
Analysis 7
2.1.2 Open source Search engines in the wild 8
2.1.3 Search utility - short overview 8
2.1.4 How search engines should perform in a digital forensic domain - short overview 9
2.1.5 Handling problems 10
2.2 Systematic Literature review 2 - Digital forensic related datasets 10
2.2.1 Purpose of the literature review 10
2.2.2 Protocol/methodology 11
2.2.3 Search phrases and justification 13
2.2.4 Search summary - datasets: 15
2.3 Systematic literature review 3 - Search capability comparison between open source search engines and open source forensic tools 28
2.3.1 Purpose of the literature review 28
2.3.2 Protocol/methodology 28
The candidate set of open source search engines 28
The candidate set of open source forensic tools 29
The inspection set of open source forensic tools and open source search engines 30
The inspection process 30
2.3.3 Inspection summary forensic tools 33
Inspection of Sluth kit - Autopsy 33
Inspection of Volatility 35

Inspection of Mozilla InvestiGator 36
Inspection of Hachoir 37
2.3.4 Inspection summary search engines 38
Inspection of Elasticsearch 38
Inspection of Solr 40
Inspection of Sphinx 41
Inspection of Dezi 42
2.3.5 Search capability comparison 43
3 Choice of methods 45
4 Experimental design 47
5 Performing the experiments and results 49
6 General discussion and conclusion 51
6.1 Discussion 51
Bibliography 53

List of Tables

1 Open source desktop/intranet search engines and their default search capabilities . . . 8
2 How the sources was located . . . 10
3 Example of datasets . . . 11
4 Where the datasets/collections can be found . . . 13
5 Search summary . . . 15
6 Datasets . . . 15
6 Datasets . . . 28
7 A candidate list of open source search engines sorted on sum . . . 29
8 The set of open source forensic tools for inspection . . . 30

Glossary

AF	Anti Forensics
DFI	Digital Forensic Investigation
FP	Forensic Practitioner
IR	Information Retrieval
TC	Time Complexity
MC	Memory Complexity
MVT	Memory Visualisation Tool
RBT	Red Black Tree
SE	Search Engine

1 Introduction

The purpose of this chapter is to present the reader with the topic of the thesis, the problem description, justification for doing the research, the research questions that will guide the research and the planned contributions of the research.

1.1 Topic covered by the project (TODO consider rephrasing)

Digital forensics investigations have to deal with a digital landscape where the amount of data increases in volume each year [1]. The big data problem introduces problems such as how can forensic practitioners (FP) process the data collected in their investigation in a reasonable amount of time and figuring out how to best handle the storage requirement of the data. Using relational databases to process the data is not appropriate as large portion of the data is unstructured [2].

Information retrieval systems like search engines (SE) have been used to help locate enterprise data. SE used in enterprises also have to deal with large volumes of heterogeneous data [3].

This master thesis proposal aims to evaluate the performance of search engines and search engine functionality on forensic data.

1.2 Keywords

Digital forensics, search engines, benchmarking, open source, recall and precision

1.3 Problem description (TODO reprase)

Forensic practitioners in digital forensics have to process large quantitative of structured and unstructured data. The processing of data have to be reliable, forensically sound and preferably be solved using a algorithm with a low memory and time complexity. Forensic practitioners can use one of many Search Engines (SE) to aid them on this task.

By knowing which SE that is out there, which algorithms they use and their performance, then the forensic practitioners can make a conscious decision on which SE that best aid them on the forensic process.

Forensic practitioners should also know the search features/capabilities of open source applications that can aid them on search tasks... REPHRASE THIS LATER AS WELL AS THE SECTION

1.4 Justification, Motivation and benefits (TODO rephrase)

Digital forensic investigation have a big data problem. Without tools that can search the data within a small time frame and provide relevant hits, then forensic investigation cannot examine the evidence in a timely manner. Which in turn can negatively effect the justice system capability of convicting criminals.

The ability for search engines to process large amount of forensic data will be benchmarked in this paper. The benchmark can provide relevant information needed to determine whether or not to invest resources on integrating search engines into the digital forensic investigation process.

1.5 Research question (TODO rephrase the reseach questions)

1. What is the purpose of search in digital forensics?
    (a) How can search be applied in a digital forensic setting?
    (b) What are the open source search engines that are still in development?
    (c) What are the open source forensic tools capable of search, that are still in development?
    (d) What are the advertised search capabilities/features of the open source forensic tools and open source search engines?
2. What is the performance of keyword search using the open source software on forensic related datasets and forensic use cases?
    (a) What is the publicly available forensic related datasets for testing?

1.6 Planned contributions (ask about feedback... I should maby add more in this section)

The contribution is a framework to evaluate the performance of a selection of search engines and their search functionality in the domain of digital forensics.

Theoretical novelty
Practical novelty

1.7 Outline

2 Related work

2.1 Systematic literature review 1 - Search in digital forensics and Open source search engines in the wild

In regards to the problem description it was unfortunately infeasible to obtain information on low level algorithms used by the search engines in table 1. This is because this information was not made accessible in the software documentation and undergoing a source code inspection would be too time consuming. But information on the search engines search functionality was possible to find on the search engine website and documentation pages. This trusted information sources will be later verified in search engines used in the master thesis experiment. The research questions is closely linked to the experiments. This chapter includes independent study into search engines found in the wild, their search capabilities and search utilities. This is information is necessary in order to begin to answer the research questions in the master thesis.

The literature review is divided up in the following subsections:

1. Application of search in digital forensic investigation: A review on the literature for the last 5 years on how search can be applied to digital forensic investigations. This section is further divided into collection, examination and analysis. Which are phases in the digital forensic investigation process model discussed in [4].**There seem to be a lack of recent surveys on the topic of usage of search in digital forensic. Many of the paper used in this systematic literature review are using search with experimental methodologies in a forensic setting**
2. Search engines: A overview of the search capabilities for a number of search engines that are open source, recently in development and that are not primarily web search engines.
3. Search utility: A look into the utility of the search engines search functionality.
4. How search engines should perform in a digital forensic domain

2.1.1 Application/experimental use of search in digital forensic investigation

Collection phase

Privacy law can regulate what method FP can use when collecting evidence. One paper [5] created a privacy protected scheme, where FP can perform a keyword search on encrypted emails. The individual emails could only be decrypted if the amount of exact matching non-blacklisted keywords provided by the FP are equal or above a certain threshold. Blacklisting or whitelisting certain keywords can make it harder for an attacker to perform a dictionary attack.

The paper by [6] argued that volume information found in the open source distributed file system platform XtreemFS is of interest to FP. The information can be used to search to find particular volumes of interest and the size of the volumes to determine if acquisition is practical. FP can search for the string "xtreemfs@" to find out if a node is connected to XtreemFS.

Evidence amongst junk:

Email spam folders are often overlooked by FP as they mostly consist of junk [7]. Criminals can craft their messages in such a way that it will to be picked up by the spam filter and hide their activities from law enforcement. Keyword searches and manual review of the spam emails is therefore important to find obfuscated evidence. The folder could be a way for criminals to obfuscate their activities, and should therefore be collected and searched.

Examination phase

Searchable hash databases:

It was claimed in [8] that it is commonplace for Forensics Practitioners (FP) to maintain a database of hashes of know illegal images and videos. FP can hash media collected in a investigation and search the database for matches. This approach has obvious limitation against anti forensics (AF) approaches such as resizing of the images. To improve upon this scheme the paper creates a custom database called *hashdb*, that stores hashes of the individual data blocks of files. This solution is more resistance against small file modification, as many of the data blocks would remain unchanged. Searching the database for matches of crime media can return a single match or a candidate list.

Searchable reference database

One study [9] showed that usernames and passwords found in computer memory can be used to identify which websites the credentials belongs to. A search condition like “&Email” and “&Passwd” can be used to search for usernames and passwords in memory. Some usernames and passwords that belongs to particular websistes can be retrived with a unique search pattern, others can be found by using the same search condition. The non-unique search conditions can use the session component found in memory to uniquely identify the website. Having a reference database for this mapping can be useful for forensics examiners that want to understand suspect activity online. Maintaining the referance database beyond the most common websites would be impractical.

Approximate hash based matching:

While not being widely adopted by the digital forensics community, approximate matching can be used to detect semantically and syntactical similar files and match it against a reference dataset [10]. Semantically similar files are files such as images that look alike in the eyes of humans. For example otherwise identical images, one in white and black and the other in colour would be perceptually the same file. The application of searching for semantically similar files can aid FP to find the origin of files of interest. Syntactical similar files are files that look similar on the byte level. Approximate hash based matching (AHBM) is not appropriate for images as they can look the same, but have different encodings. But are well suited for dealing with unstructured data such as text files, memory dumps and fragmented files. The paper concludes that the same results can be accomplished with string search as with approximate matching, but this would require far more from the FP.

Inexact search

One paper [11] created a search algorithm called *ScalClone* that aims to find exact and inexact code fragments between analysed and un-analyzed malicious assembly files. Exact fragments are identified by searching for regions with the same hash value. Inexact fragments are fragments that share many mnemonics and operand types. They are identified by first constructing a binary vector with respect to feature frequency and features mean value, and then comparing the co-occurrences of the fragments. If the co-occurrences count is greater or equal to the similarity threshold, then the fragment is considered a inexact clone. Inexact search is not effected by reordering as the frequency of the mnemonics remains unchanged. Obfuscation by adding do-nothing instruction drops the recall rate to 90% and compiler optimization drops it to 62%.

Deduplication:

One issue with collected forensic image of a storage device like hard disk drive (HDD) is duplicated files [12]. Processing duplicated files leads to unnecessary overhead in the examination phase. One way to solve this issue is by arranging the files in a red black tree structure (RBT). Duplicate nodes in this structure can be found by searching using wildcards. After identifying duplicate nodes their child nodes will be rearranged in the tree and then the duplicate node will be removed from the structure. The time complexity for searching, inserting and removing nodes in RBT is $O(log_2(n))$ for the average and worst case. This proposed solution do not state in detail how their scheme identifies files with the same content. While identifying the same file names using wildcard seems resonable, hash matching is more appropriate for telling if two files have identical content.

A proposal was made in [13] to identify duplicate images where the file name, file extension or file attributes (e.g hidden, compressed, encrypted and protected Operating System File) did not match the source image. The proposal used the source modified timestamp to search for duplicate files. 1000 files spread across 30 folders totalling 3.09 GB in size was processed in 1 minute and 32 seconds. The same files spread across 300 folders took 16 minutes 23 seconds longer to process. So its application is limited to environments with a small number of folders. The proposal is also vulnerable to tampering done to the modified timestamp attribute.

Examination limited by law:

According to [14] the United State Supreme Court are beginning to demand that the examination process are limited in its scope. This means that the goals and objectives must be clearly stated, as well as a justification for what the examiner will search for and the boundary of the search. Failure to comply could negatively effect their case in court. This restriction might force a better resource management of the examiner resources. But it can also make it more difficult to examine evidence that is hidden in unusual locations, as its examination would be difficult to justify. Simply searching for everything in a Gigabytes or Terabytes search space would not solve the problem as this task is infeasible even when using common digital forensics tools or automated tools [15, 16]. The courts also put constraints on how long seized data can be processed by the examiner, before it is returned to its owner [17]. It is argued in [18] that the searching by the examiner, can be aborted after the most probable places have been processed. More specific search criteria can reduce privacy violations and reduce number of false positive hits [19]. The question then arises how specific can you be before negativity impacting the recall rate.

RAM search

A survey [20] stated that string search in volatile memory examination is useful in order to find residue of user activity, passwords, encryption keys and side effects of malicious scripts. Searching in swapped out memory pages in *windows* can potentially provide evidence of old user activity, as the swapped file is often not cleared after system reboot [21].

Pool tag scanning is a type of exhaustive search on volatile memory that is used to find data structures such as direct kernel object manipulation (DKOM) which is used by malware to hide processes [22]. The study [22] stated that exhaustive search might not be appropriate for time sensitive investigations. They therefore created pool tag quick scanning, which reduces the search space to memory pages related to pool allocations. The search space reduction can be "multiple orders of magnitude" and the accuracy of the search results remains high.

The use of visualization to aid search:

A comparison was done in [23] to test the accuracy and speed of which experienced participants in networking, windows operating system, malware and incident response, are to solve forensics tasks. The participants where given the same tasks and the same forensics image. They where split into two groups, one that used normal text search and the other that searched using a memory visualization tool (MVT). The MVT showed relationships between the data and had a whitelisting algorithm that removed known good files from the search space. The results showed that the participants that used the MVT completed the tasks faster and more accurate. I infer from the text that the number of participants are 10 (minus one outlier). Laying to much weight from the results on this low sample size might not be appropriate.

Issues with keyword searches

The study [24] compared the state of the system before and after forensics examination using the following bootable forensics environments: *Knoppix v7.0, Helix 3 Pro 2009R3 and Kali Linux v1.0*. Keyword searches was used during the examination process to simulate an investigation. The hash value taken on the forensics image before and after examination, did not match in any case. It was mainly the “last accessed” timestamps on files that was altered after the examination. Performing keyword searches in those environments can therefore be problematic in cases where establishing a timeline is important.

It is argued in [25] that keyword searches resulting in large number of false positive hits, can be reduced by using background knowledge from the investigation.Also while keyword search algorithm are useful, they are inept at processing terabytes of data. An alternative to keyword search is *Fuzzy search/fuzzy matching*. This search can be used to find elements missed by the normal keyword search such as misspelled words and slang terms. [15].

The use of clustering to improve the usefulness of search and suggestion in search:

One study [16] used keywords search terms to cluster forensic data to reduce examination overhead. There is one cluster per search term. In order to help the examiner choose good search terms, the system returns the most frequent used search terms found in the forensics data. Both with and without suggestions, the system performs good with respect to average precision and recall. The system is also scalable as the runtime grows linearly with the number of documents.

TODO Reference Solr that implements a type of keyword suggester in their application.

FP have to search though large volumes of heterogeneous data. One study [26] evaluated the performance of clustering techniques on a forensic dataset containing 2640681 search hits. They achieved a precision improvement of a factor 15 over non-clustering and a overall average precision of 67%.

Analysis

Orphan Files - deleted files

Finding evidence of deletion of user activity on the suspect machine is of interest of FP [27]. Searching the Update Sequence Number (USN) Journal file on the NTFS can reveal when and where files have been created, viewed, renamed, moved or deleted. Another study [28] showed how searching for the string 'for deletion' in a Hadoop Distributed File System (HDFS) is useful to find evidence of deleted files. The paper [29] claimed that only the row directory is overwritten with a NULL value when a row is deleted in the database DB2 or SQL sever. This allows a FP to search these databases for the deleted rows and restore them by considering the valid row directory values of their previous and following row directory entry.

One study [30] mined 1100 chat logs to find the most significant terms, users and chat sessions. Two bigraphs are constructed. The mapping in the first bigraph is such that we can observe which term (Hub) has been said by which users (Authorities) and what terms (Hubs) have been said by a user (Authority). The second bigraph has similar mapping, but the Hub is the term and the authority is the chat session. A self-customized hyperlink-induced topic search (HITS) algorithm is used to iteratively set the Authority and Hub score. A selection of the highest scoring users, chat sessions and terms are used together with user metadata and session metadata to construct a social graph. Clustering is applied on the social graph to find shared interest and interactions between users.

One study [31] showed how traces found from volatile memory in IEEE 802.11 wireless devices, that is in radio range from each other can answer important forensics questions like Who, When and Where. There are two types of broadcast traffic frames that can answer these question. As their format is known, they can easily be found by using regular expression search. The probability that the frames are still in the devices volatile memory depends on external and internal conditions like the extent and nature of the broadcast traffic processed by the device and the configurations of the device. This methodology would therefore only work in a few real life scenarios and mostly in non-urban areas.

File carving:

Search helps file carving tools identify header, footer and fragments used to identify where a file begins and end and use this information to restore the file [32]. Some file carving tools are able to restore files independent on the underlying file system. Exhaustive search can be used to find each combination of header and footer of a video and then try to validate/decode on the restored file to see if it is a valid video. Search can be used to find the order of the fragments and codecs search codes to identify fragments belonging to videos.

Encoding:

The FP may encounter digital environments where the binary data is encoded using multiple different UNICODE encodings and that the type of UNICODE are unknown [33]. The share number of possible UNICODE encodings means that the same text can be represented in many different ways. Resolving the underlying encoding in the worst case can require number of search passes equal to the number of possible encodings. The average case is much better as many encodings are not widely used. The regular expression search engine *lightgrep* aims to deal with the encoding problem. Lightgrep uses UNICODE characters as string literals in the regex expression to be encoding independent. For handling the encoding Lightgrep uses multi pattern search enabling it to search for multiple encodings in parallel. The search engine currently support 180 encodings making it possible to perform UNICODE-aware searches.

2.1.2 Open source Search engines in the wild

In table 1 is a collection of open source search engines still in development. The columns of this table is explained in the list below. The update column is the last advertised change to the software from the point of this review.

- S_1 = *Full text search*
- S_2 = *Faceted search*
- S_3 = *Spatial/Geospatial search*
- S_4 = *Fuzzy search*
- S_5 = *Streamed search*
- S_6 = *Phonetic search*
- S_7 = *Semantic search*

Table 1: Open source desktop/intranet search engines and their default search capabilities
Source: [34–63]

Name	S_1	S_2	S_3	S_4	S_5	S_6	S_7	**Update**
Dezi	✓	✓						28.11.2016
Apache Solr	✓	✓	✓	✓	✓			06.03.2017
Sphinx	✓			✓				08.09.2016
Sifaka	✓							25.01.2017
OpenSearchServer	✓	✓	✓			✓		13.01.2017
Luwak					✓			06.03.2017
Datafari	✓	✓						23.03.2017
Elasticsearch	✓	✓	✓	✓		✓		24.04.2017
groonga	✓		✓					24.04.2017
tantivy	✓							23.04.2017
tntsearch	✓		✓	✓				20.04.2017
pouchdb-quick-search	✓							22.02.2017
OpenSemanticSearch	✓	✓		✓			✓	16.04.2017

2.1.3 Search utility - short overview

According to the whitepapers [64], [65] *Full text search (FTS)* is suitable for finding relevant documents in a large set of unstructured data. A lot of the data gathered in a forensic investigation is unstructured [66]. It is more appropriate to use FTS to respond to ad hoc request than requests with a predefined answer [64]. A document in FTS is considered a list of searchable terms (e.g. words and numbers) [65]. The terms are usually indexed in order to make them easier to search.

Faceted search is a way of traversing the corpus based on categories (facet) and subcategories (facet values) [67]. In faceted search it is possible to find the same the same data points by using different traversal paths. *Faceted search* is useful for exploring the corpus and the facet values aid the searcher to create more precise search phrases. It is common practice in *faceted search systems* that only the most frequent facet values are shown. This makes finding more obscure items difficult.

Fuzzy keyword search retrieves both documents that matches exactly with the search phrase and those within a similar distance [68]. The distance can be measured by using the Levenshtein distance. Which compares the minimum number of insertions, deletions or substitutions are needed for string A to equal string B. The paper [69] claims that fuzzy search is helpful when the searcher have do not have sufficient domain knowledge of the dataset he is searching. **TODO: Rewrite so to emphesize that this have to do with autocompletion**

Phonetic search is matching based on similar sounding words [70], [55]. One example of a phonetic algorithm is *Soundex*. It encodes a word into a 4 character code starting with the same character as the word [70]. Similar sounding characters like s,f,p and v are represented by the same number. Repeating characters, vowels and certain letters are ignored by the algorithm. Truncation and padding are used to make sure that all words are represented by a 4 character code. The limitation with this approach is that only words starting with the same letter would have a chance to match with the same code. Phonetic algorithms are designed to handle specific languages, making them limited in their utility [55]. The aim of Phonetic search is not improving precision but to increase the recall rate.

Geospatial search is searching a corpus where the documents have associated geographic data such as latitude and longitude. One example of using the location data is to search for registered criminals that lived in the vicinity of a crime scene [71]. It can also be used to find all previous search warrants on a address or all search warrants in some proximity to a given address.

Documents that do not contain the terms of the user query can still be relevant [72]. Classical retrieval based on lexicographic term matching will not retrieve documents that are lexicographically different but semantically similar. To improve information retrieval of documents Semantic search can find semantically similar terms that are often overlooked by using stemmed synonyms or Ontology. Ontology models a domain into concepts, attributes and relations [73].This model provides the semantic reasoning needed to retrieve meaningful documents with respect to the user query [74].

Streamed search was explained in [75, 76]. In traditional full text the documents are often indexed using inverted indexing to optimize the time it takes to find the queried documents. Running all possible queries on the documents works well if the complexity of the queries and the data velocity is low. Network log files is a example of a stream (continuous data flow) where traditional search is impractical. Stream search uses inverted indexes on queries instead of documents. By doing so it is possible to take the new log entry and query the inverted index to see which indexed queries match the new entry. Now the search have identified the minimum number of queries that need to run on the new entry. This approach could potentially save high amount of computer resources.

2.1.4 How search engines should perform in a digital forensic domain - short overview

The importance of the measurements precision and recall in Information Retrieval (IR) systems, like Search Engine (SE) depends on the application [77]. In the domain of

Digital Forensic Investigation (DFI) precision is more important in the early phases of the forensic investigation, as relevant evidence is vital to guide the process of finding new evidence. At the later stages of the DFI, recall becomes more significant than precision, as Forensic Practitioner (FP) wants all available evidence to build a court case.

2.1.5 Handling problems

The table 2 show which search phrases and search resources used to collect the sources and the number of resulting hits. Other sources like finding the search engines web pages was found in a snowball fashion. Where relevant project links on sourceforge and github was used to locate the search engines.

Table 2: How the sources was located

Query	Search resource	Hits
("Abstract":enterprise AND search AND engine), Year: 2014-2017	ieeexplore.ieee.org	27
"Enterprise search", in abstract, year: 2014-2017, source type: Scholarly Journals	search.proquest.com	24
recordAbstract:(+enterprise +search) , year: 2014-2017	dl.acm.org	44
in abstract (Solr OR ElasticSearch)	arxiv.org	6
in abstract, title and keywords: Enterprise search, year 2014-2017	sciencedirect.com	47
in abstract:Solr OR ElasticSearch, Scholary journals, year:2014-2017	search.proquest.com	47
in abstract (Solr OR ElasticSearch), year:2014-2017	ieeexplore.ieee.org	72
recordAbstract:(Solr Elasticsearch), year:2014-2017	dl.acm.org/	18
information retrieval unstructured data (general search), year:2014-2017	ieeexplore.ieee.org	122
"information retrieval" "unstructured data" survey	scholar.google.no/	2990
(+"Digital forensics" +search) - any fields	dl.acm.org	7
(+"Computer forensics" +search) - any fields	dl.acm.org	7
in journal "Digital Investigation" : search, 2014-2017	sciencedirect.com	161
in book "Digital Forensics Threatscape and Best Practices" - year: 2016 - search phrase: search	sciencedirect.com	10
in publication "IEEE Transactions on Information Forensics and Security", year:2014-2017	ieeexplore.ieee.org	42
basic search " Digital forensics search", year:2014-2017	ieeexplore.ieee.org	65

2.2 Systematic Literature review 2 - Digital forensic related datasets

2.2.1 Purpose of the literature review

Identify and summarize publicly available datasets that relates to digital forensic and consider their applicability for this thesis experiments. Table 3 shows examples of relevant datasets: (**TODO: change this paragraph, replace with research question... do not consider their applicability here ... do that in experimental design... or somewhere else**

Catagory	**Abbreviation**	**Example dataset**
Forensics images	IMG	*The Real Data Corpus (RDC)*
Files	FILE	*RAISE (RAw ImageS datasEt)*
RAM contents	RAM	*Memory Buddies Traces(MBT)*
Network	NET	*Common crawl*
Malware	MAL	*Kharon dataset*
Email	EM	*The Webb Spam Corpus 2011*
SMS	SMS	*The NUS SMS Corpus*
Password	PASS	*Yahoo Password Frequenc*
Phishing	PHI	*Phishing Websites Data*
Spam	SPAM	*TREC 2011*
Authorship	AUTH	*Personae*
Financial data/ fraud	FIN	*CMS dataset*
Forgery corpus	FORG	*MICC-F2000*
Collection of different datasets	COLL	*CAIDA data*

Table 3: Example of datasets

Decisions was made to limit the scope of the data collection, by excluding biometric datasets such as images of fingerprints, hand signature, gait, voice recognition and iris. But the review will include authorship attribution corpus.

2.2.2 Protocol/methodology

1. Search digital libraries and scan scientific articles for names, direct links or sources related to the datasets above and use this information on google search engine to identify individual datasets or repositories of datasets.
2. Document search phrases that resulted in identifying new datasets.
3. Repeat step 1 and 2 with other resources like github, keegle and figshare to locate more datsets.

In the planing phase of the literature my supervisors aided me by providing 3 sources to publicly available datasets resources

1. *Govdoc1*
2. *AZSecure-data*
3. *Enron dataset*

These datasets/collections have also been included in my review.

During the collection phase of the literature review, three related reviews was identified. The first review was from 2014 and identified 7 datasets [78]. The second review is as recent as 2017 and compiled a list online of 79 digital forensic related datasets [79], [80]. The third review [81] was from 2014 and focused on network related datasets. It identified 10 datasets, 2 of which are broken links, and 4 available datasets not considered for inclusion in this review due to time considerations.

This review expands on the three reviews and its findings where largely independent from the 3 previous works. The table 4 shows how this review expands the knowledge on the topic of publicly available forensic related datasets. In table 4 the index, name and catagory of the collected datasets is presented as well as which of the 4 reviews included it[where this paper is counted as one of the 4 reviews]. The assigned dataset indexes, names and categories will be the same throughout this review. Not all dataset creators explicitly named their creation, in those cases a new name was assign Ad hoc to the dataset. More information about each individual dataset or repository can be found in section 2.2.4. Two datasets in the review from 2014 [78] was not included in this paper. The first was a face recognition dataset and the second was a forensic image dataset called MemCorp which I could not identify online. The review from 2017 [79] included 39 datasets that was not identified in this systematic literature review. The 39 dataset has been included in this review as a collection of dataset that is summarized in table 6 with index 21 and name DFCF review. DFCF review contains 4 language corpora, and languages was not considered a category for forensic related corpora in this paper. The difference in methodology with this review and [79] seems to be that this review did not restrict the publishing year when searching though scientific databases, different categories was considered relevant, and dataset databases like Kaggle and figshare was searched in this review.

Index	Name	Catgory	This paper/review	in review [78]	in review [79]	in review [81]
1	BAC	AUTH	✓			
2	CTFAC	AUTH	✓			
3	PCSN	AUTH	✓			
4	TBGC	AUTH	✓			
5	Personae	AUTH	✓			
6	PAN-Enron	AUTH	✓			
7	PAN/CLEF12	AUTH	✓			
8	PAN13	AUTH	✓			
9	PAN14	AUTH	✓			
10	PAN15	AUTH	✓			
11	RCTAC	AUTH	✓			
12	MUD03	AUTH	✓			
13	netresec	COLL	✓			
14	MTA13-17	COLL	✓			
15	pcapr	COLL	✓			
16	PCAPsDB	COLL	✓			
17	CAIDA	COLL	✓		✓	✓
18	csmining	COLL	✓			
19	AZSecure-data	COLL	✓			
20	Digital Corpora	COLL	✓		✓	
21	DFCF review	COLL	✓		✓	
22	GIfiles	EM	✓	✓		
23	USHCE	EM	✓			
24	419 dataset	EM	✓			
25	MLE200	EM	✓			
26	Enrondata	EM	✓	✓		
27	Raise	FILE	✓			
28	SherLock	FILE	✓			
29	AndroZoo	FILE	✓			
30	CTD15	FIN	✓			
31	UCSD-FICO-09	FIN	✓			
32	CMS	FIN	✓			
33	PaySim	FIN	✓			
34	BankSim	FIN	✓			
35	MICC	FORG	✓			
36	Brian Carrier	IMG	✓		✓	
37	RDC	IMG	✓	✓	✓	
38	CFReDS	IMG	✓	✓	✓	
39	VirusShare	MAL	✓			
40	BIG15	MAL	✓			
41	Drebin	MAL	✓		✓	
42	DroidWare	MAL	✓			
43	MILCOM16	MAL	✓			
44	Kharon	MAL	✓			
45	Mudflow	MAL	✓			
46	ISOT2010	MAL	✓			
47	ECML/PKDD07	MAL	✓			
48	CSIC10	MAL	✓			
49	BlogPcap	MAL	✓		✓	
50	Malrec	MAL	✓			
51	CTU-13	MAL	✓			
52	ISCX Botnet	MAL	✓			
53	ISCXAB	MAL	✓			
54	DAROA98/99	NET	✓	✓		✓
55	DARPA2000	NET	✓			✓
56	MAWILab	NET	✓			✓
57	KDD Cup99	NET	✓			
58	UNSW-NB15	NET	✓			
59	NSA-CDX	NET	✓			
60	ADFA	NET	✓			
61	Kyoto data	NET	✓			
62	crawdad	NET	✓		✓	✓
63	ICS-pcap	NET	✓			
64	Common Crawl	NET	✓			
65	NSL-KDD	NET	✓			
66	ISCXTNT	NET	✓			
67	ISCXVNV	NET	✓			
68	ISCXIDS	NET	✓			
69	YPFC	PASS	✓			
70	VincentPassword	PASS	✓			
71	MBT08	RAM	✓			✓
72	NUSSC	SMS	✓			
73	WebbSC11	SPAM	✓			
74	DITSSC	SPAM	✓			
75	TREC05-07	SPAM	✓			
76	Hewlett spam	SPAM	✓			
77	WEBSPAMUK07	SPAM	✓			
78	microblogPCU	SPAM	✓			
79	TREC11	SPAM	✓			
80	SPAM/HAM	SPAM	✓			
81	phishtank	PHI	✓			
82	millersmiles	PHI	✓			
83	PWDS15	PHI	✓			

Table 4: Where the datasets/collections can be found

2.2.3 Search phrases and justification

Documents was excluded from consideration if their title had little relation to information security, and if the document format was not easily searchable. An example of the latter case is pdf documents scanned by a scanner machine, where full text search of the text content is not applicable. Without the assistance of search, the process of finding the datasets would be too time consuming.

In table 2.2.3 is a summary of the collection phase of the literature review. Entries included in this table all lead to finding new datasets. An entry has an ID number, search phrase + search options, database name (search resource) and the number of hits for the search phrase. Entries with ID 1-5 is essentially full text search (matching based on meta data and text content). Fulltext search lead to more false positives, then only meta search. But was used in cases where the number of hits was manageable. An example for when fulltext was deemed unmanageable can be seen in entry 6, where meta search was used instead. The phrase 'forensic dataset' was used to find different types of relevant datasets, but this phrase alone is not good enough. This is because relevant papers may use publicly available datasets, but does not contain the term 'forensic'. Therefore more specific search terms from list in subsection 2.2.1 was also used. In entry 9 the NOT operator was used to discard biometric datasets. Entry 10 in table 2.2.3 returned hits that both included the phrase 'IDS dataset' and the term 'Network' in the meta data, and excluded hits that contained some already known network datasets. The term IDS was used to reduce the number of non-network related articles. This term may exclude some relevant hits, but its usage is justified as the other search phrases also covered some network related datasets. In entry 16 the first 10 results was used on Google to look find datasets on Github. This was done as it was tricky to identify relevant repositories using Githubs internal search. In entry 18 figshare did not provide the number of hits. Therefore Not Available (N/A) is in the #Hits column for this entry.

ID	Search phrase (comma (,) separates search options)	DB	Hits
1	forensic corpora, exact phrase match	link.springer	22
2	forensic corpus', advanced search, both words must match (be present) in any field	dl.acm	9
3	forensic corpora', advanced search, both words must match (be present) in any field	dl.acm	3
4	forensic dataset', advanced search, both words must match (be present) in any field	dl.acm	61
5	forensic corpus, full text search	search.arxiv	112
6	forensic dataset, in metadata only	ieeexplore	94
7	malware dataset, in metadata only	ieeexplore	174
8	((password dataset) NOT biometrics), in metadata only	ieeexplore	19
9	Spam dataset, in metadata only	ieeexplore	173
10	(((((((((IDS dataset) AND Network) NOT DARPA) NOT KDD) NOT KDD99cup) NOT DARPA98) NOT DARPA99) NOT DARPA-98) NOT DARPA-99) NOT NSL-KDD), in metadata only	ieeexplore	118
11	fraud dataset, in metadata only	ieeexplore	104
12	Forensic dataset, in All Sources(Computer Science), no books	sciencedirect	1100
13	fraud	kaggle	10
14	spam	kaggle	3
15	email	kaggle	18
16	dataset github	google	576000
17	spam	figshare	107
18	network	figshare	N/A

Table 5: Search summary

2.2.4 Search summary - datasets:

Table 6 is a summary of the identified datasets in this review. An entry in this table is explained in the list below:

- Column I = Numbered index
- Column Name = The short name of the dataset
- Column Acc = Access, where P=public and R=By request
- Column DT = Data type, where S = Synthetic, R=Real and H=Hybrid Column CAT= Catagory, where the catagories abbriviations is shown in table 3
- Column Size= Size is either given in S=samples, MB, GB, (comrpressed/uncompressed) or Not avaliable (N/A)
- Column Description= A description that will include the name of the dataset, where it can be downloaded from, include original paper if available and additional details about the dataset.

Table 6: Datasets

I	Name	Acc	DT	Cat	Size	Description
1	BAC	P	R	AUTH	19320 S	The Blog Authorship Corpus (BAC): A authorship corpus of 681288 Blog entries and 19320 problems [82], [83] Obtain dataset here[1]

---

[1] http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm

. . . continued

I	Name	Acc	DT	Cat	Size	Description
2	CTFAC	P	S	AUTH	20 S	A capture the flag (CTF) authorship corpus (CTFAC) [84], [85]. The corpus have been used in the multi-classification problem of classifying the origin of the exploit attempts to one of 20 CTF teams. The data is available in JSON format and includes source and destination of attack, timing information and histogram of payload. Obtain dataset here [2]
3	PCSN	R	R	AUTH	609 S	Polish Corpus of Suicide Notes (PCSN) are real suicide letter written by both young and old polish men and women from the period of 1999-2009 [86], [87]. Obtain dataset here [3]
4	TBGC	P	R	AUTH	12 S	The Brennan-Greenstadt corpus (TBGC) contains two documents from each of the 12 participating authors [88], [89]. In the first text the authors attempted to obfuscate the characteristics of their writing. And in the second text the authors tried to imitate the writing style of a different writer. Obtain dataset here [4]
5	Personae	R	R	AUTH	145 S	The paper claims that the size of the German corpus Personae makes it possible to classify the author of the text as well as the author personality [90], [91]. Personae consist of 145 bachelor student essays with lengths around 1400 words. The students, took a personality test. This test made classification of their personality possible. But it is difficult to infer from the sources [90], [91] whether the personality test is part of the dataset or not. Obtain dataset here [5]
6	PAN-Enron	P	R	AUTH	12338 S	PAN-Enron corpus is a subset of the Enron dataset and can be used for authorship attribution and verification. 24% of the samples is from non-Enron authors while the rest is from the Enron set [92], [93]. Names and email addresses was omitted from the dataset. Obtain dataset here [6]

---

[2] `https://www.dropbox.com/sh/17d4eyg0cwoxg8s/AAA5g1NvQw-tUoZPvldloddRa?dl=0`

[3] `http://www.pcsn.uni.wroc.pl/`

[4] `https://psal.cs.drexel.edu/index.php/JStylo-Anonymouth`

[5] `https://www.clips.uantwerpen.be/datasets/personae-corpus`

[6] `http://pan.webis.de/data.html`

. . . continued

I	Name	Acc	DT	Cat	Size	Description
7	PAN/ CLEF12	P	R	AUTH	N/A	The dataset contains training and test data for several authorship attribution scenarios based on works of fiction. Each scenario has a different amount of authors, number of documents, and minimum word length [94], [93]. Obtain dataset here [6]
8	PAN13	P	R	AUTH	110 S	Authorship classification on English, Spanish and Greek texts. Most of the documents are in the word length range 1001-1500 words [95], [93]. Obtain dataset here [6]
9	PAN14	P	R	AUTH	4959 S	Authorship attribution corpus with documents written in English, Dutch, Spanish, and Greek [96], [93]. University students created the Dutch and English documents. And the Spanish and Greek documents was obtained from newspapers. Obtain dataset here [6]
10	PAN15	P	R	AUTH	3701 S	Authorship attribution corpus with documents written in English, Dutch, Spanish, and Greek. The authors of the Dutch documents was Students at a university in Belgium [97], [93]. English documents was taken from theatre plays. Spanish and Greek documents was obtained from opinion articles. Obtain dataset here [6]
11	RCTAC	P	R	AUTH	1000 S	Reddit Cross-Topic AV Corpus (RCTAC) consist of 1000 reddit users and their comments from 2010-2016 on 1388 different subjects [98], [99]. Obtain dataset here [7]
12	MUD03	P	N/A	AUTH	750000 S	Masquerading User Data 2003 (MUD03). A dataset of 750000 UNIX commands [100], [101]. The commands are either from one of 50 legitimate users or a user imitating one of the 50. Each legitimate user has 5000 commands that is theirs and a random proportion of 10000 commands that is attribute to them or a masquerading person. Obtain dataset here [8]
13	netre-sec	P	P,R	COLL	N/A	This website have compiled a list of available PCAP files online [102]. The list contain PCAP files that have been used in competitions, conferences, and PCAP files that contain malware or exploits. Obtain dataset here [9]
14	MTA13-17	P	N/A	COLL	≈ 1100 S	malware-traffic-analysis (MTA) website host a collection of PCAP files and malware samples [103]. Obtain dataset here [10]

---

[7] `https://www.dropbox.com/sh/f2mlp6u5vervx9b/AABr_c7qrmahCqUviIu3ORz6a?dl=0`

[8] `http://www.schonlau.net/intrusion.html`

[9] `https://www.netresec.com/?page=PcapFiles`

[10] `http://malware-traffic-analysis.net/`

. . . continued

I	Name	Acc	DT	Cat	Size	Description
15	pcapr	P	S,R	COLL	6,050,710,9 S_1 / 3465 S_2	A large publicly available searchable database that contains 60507109 packets and 3465 pcaps [104]. Obtain datasets here [11]
16	PCAPs-DB	P	S,R	COLL	N/A	PCAPsDB is a repository of different PCAP files, some of which are Malware [105]. Obtain dataset here [12]
17	CAIDA	P	R	COLL	71 S	CAIDA Data: A collection that contains a mixture of publicly availably and by request network datasets [106]. These network dataset focuses on Autonomous System (AS) topology, denial of service attacks, worms, anonymized passive network traffic monitoring and darknet traffic. Obtain dataset here [13]
18	cs-mining	P	R	COLL	10 S	A collection of spam datasets, news articles, system calls from malware and Reuters-21578 dataset used for text categorization [107]. Obtain datasets here [14]
19	AZ-Secure-data	P	R	COLL	111 S	Is a resource that contains list of geopolitical discussions, dark web forum threads, phising and legitimate sites, malware, network traffic and data from social media communication [108]. Obtain datasets here [15]
20	Digital Corpora	P,R	R	COLL	N/A	Is a collection that contain: • Cell phone images • Disk images and the already mentioned RDC dataset. • Network traffic including the already mentioned DARPA 1998,1999 and 2000 datasets. • Govdocs1 a dataset with ≈ 1000000 files. [109] Obtain datasets here [16]
21	DFCF review	P	R,S	COLL	79 S	DATASETS FOR CYBER FORENSICS (DFCF) contains 39 datasets of type: chat logs, APK, File, Malware, forensic images, picture, email, network, RAM dumps, language corpora not otherwise considered in this review [79], [80]. Obtain datasets here [17]

---

[11] http://www.pcapr.net/home

[12] https://www.evilfingers.com/repository/index.php

[13] https://www.caida.org/data/overview/

[14] http://csmining.org/index.php/data.html

[15] http://www.azsecure-data.org/other-data.html

[16] https://digitalcorpora.org/corpora

[17] http://datasets.fbreitinger.de/datasets/

. . . continued

I	Name	Acc	DT	Cat	Size	Description
22	GIfiles	P	R	EM	5,000, 000 S	The Global Intelligence files (GIfiles) are a collection of 5 million leaked emails from Stratfor, that gives insight into how the intelligence community operates [110], [111]. Obtain dataset here [18]
23	USHCE	P	R	EM	60 MB S	≈ 7000 PDF pages of [US president candidate] Hillary Clinton emails (USHCE) was released as a result of a freedom of information claim [112]. The dataset contains both email content and metadata. Obtain dataset here [19]
24	419 dataset	P	R	EM	2500 S	419 fraud emails: Content and metadata from 2500 fraud emails [113]. Obtain dataset here [20]
25	MLE-200	P	R	EM	200 S / ≈ 2MB	Multilanguage emails(MLE): Spanish, English Portuguese Emails [114]. Obtain dataset here [21]
26	Enron-data	P	R	EM	N/A	Enrondata is a website that has a compiled list that links to differernt versions of the ENRON dataset that is in PST or MIME format and with different record sizes [115]. Obtain dataset here [22]
27	RAISE	P	R	FILE	350GB N/A	RAw ImageS datasEt (RAISE): 8156 Unprocessed and high resolution images. The images are taken by the following cameras: Nikon D40, Nikon D90 and Nikon D7000 [116], [117]. The original paper states that this dataset can be useful to test image forgery algorithms [116]. Obtain dataset here [23]
28	Sher-Lock	R	R	FILE	10 billion S	SherLock is a Android Smartphone dataset that contains running application/process information, sensory data and OS data captured with normal user privileges [118], [119]. The dataset also have labels that can be assign to describe ongoing malicious activity on the phone. Obtain dataset here [24]

---

[18] https://wlstorage.net/torrent/gifiles/

[19] https://www.kaggle.com/kaggle/hillary-clinton-emails

[20] https://www.kaggle.com/rtatman/fraudulent-email-corpus

[21] https://figshare.com/articles/Corpus_200_Emails/1326662

[22] https://enrondata.readthedocs.io/en/latest/references/data/

[23] http://mmlab.science.unitn.it/RAISE/

[24] http://bigdata.ise.bgu.ac.il/sherlock/#/download

. . . continued

I	Name	Acc	DT	Cat	Size	Description
29	AndroZoo	R	R	FILE	5,546, 565 S	AndroZoo dataset includes over 5 million android applications (APKs) [120], [121]. The APKs was obtained by crawling multiple APKs distributors such as google play, AppChina, torrents etc. Efforts was made to avoid downloading duplicate files from the same vendor. But creators of the dataset gives no guarantees that the same file was not downloaded from multiple vendors. Each sample contains a zipped apk file, with its byte code, meta data, signed certificate and miscellaneous files. Obtain dataset here [25]
30	CTD15	P	R	FIN	68MB / 284807 S	Credit transaction dataset (CTD) A numerical dataset of 284807 bank transactions [122], [123]. Of all the transactions only 492 of them are fraudulent. In the feature vector there are 28 principal components, the elapsed time between transactions, the transactions amount, and the fraud/not fraud class label. Obtain dataset here [26]
31	UCSD-FICO-09	P	R	FIN	$>$ 100000 S	UCSD-FICO-09 is a electronic commerce fraud dataset with anonymized features [124], [125]. The numerical dataset has both labelled (training) and unlabelled(testing) samples. Obtain dataset here [27]
32	CMS	P	R	FIN	$\approx$ 6 GB	Real financial statements in .csv format from Centers for Medicare & Medicaid Services [US], in the year 2013-2016 [126], [127]. Obtain dataset here [28]
33	PaySim	P	S	FIN	182MB	A smaller subset of the Synthetic PaySim dataset is available on Kaggle [128], [129]. The synthetic data was generated based on real world examples. The feature vector contain time information, transaction type, identifier for the user(s) involved in a transaction, old and current state of the balance sheet and class labels regarding if the transactions is considered fraudulent. Obtain dataset here [29]
34	BankSim	P	S	FIN	13MB	BankSim is a synthetic fraud dataset with 587443 legitimate and 7200 illegitimate transactions/samples [130], [131]. The simulating agent generating the dataset is based on real bank transactions. The feature vector contains age range, gender catagory, what the transaction was spent on, zip code etc. Obtain dataset here [30]

---

[25] `https://androzoo.uni.lu/`
[26] `https://www.kaggle.com/dalpozz/creditcardfraud`
[27] `https://www.cs.purdue.edu/commugrate/data/credit_card/`
[28] `https://www.cms.gov/OpenPayments/Explore-the-Data/Dataset-Downloads.html`
[29] `https://www.kaggle.com/ntnu-testimon/paysim1`
[30] `https://www.kaggle.com/ntnu-testimon/banksim1`

. . . continued

I	Name	Acc	DT	Cat	Size	Description
35	MICC	P	R	FORG	220/ 2000 S	MICC-F220 and MICC-F2000 are datasets that contains untouched images and images where parts of the image is modified by scaling, rotating and scaling [132], [133]. The datasets have been used to benchmark a copy-move forgery algorithm. Obtain dataset here [31]
36	Brian Carrier	P	R	IMG	14 S	A collection of forensic images made/hosted by Brian Carrier [134]. The 14 forensic images can be divided up into the following categories: NTFS file systems, FAT file system, ISO9660 file system and a memory image. Brian created scenarios to test string search, partitions with multiple file systems, file carving etc. Obtain dataset here [32]
37	RDC	R	R	IMG	70TB Compressed	The Real Data Corpus (RDC) is data collected of digital devices from the secondary market [135]. The dataset contains hard drives images, flash memory images and CDROMS. According Obtain dataset here [33]
38	CFReDS	P	S	IMG	16 S	Computer Forensic Reference Data Sets (CFReDS) can be used for forensic tool testing [136]. CFReDS includes forensic images and simulated data for memory forensics, file carving, string search and file recovery. Obtain dataset here [34]
39	Virus-Share	R	R	MAL	2,938, 567,4 S	VirusShare.com is a virus sharing website with currently 29385674 malware samples [137]. Obtain dataset here [35]
40	BIG15	P	R	MAL	≈ 500GB	A dataset for classifying known malware and their associated malware family [138]. There are in total 500GB worth of malware samples, that belongs into one of 9 families of malware. Obtain dataset here [36]
41	Drebin	R	R	MAL	5560 S	The Drebin Dataset have 5560 malicious android applications that can be categorized into one of 179 malware families [139], [140]. Obtain dataset here [37]
42	Droid-Ware	P	S	MAL	399 S	DroidWare is a malware dataset for the android platform. The dataset is made up of 278 benign and 121 malicious samples [141], [142]. Each sample has a 152 feature vector of Android application permissions. Obtain dataset here [38]

---

[31] https://github.com/lambertoballan/sift-forensic/blob/master/README.md

[32] http://dftt.sourceforge.net/

[33] https://digitalcorpora.org/corpora/disk-images/real-data-corpus

[34] https://www.cfreds.nist.gov/

[35] https://virusshare.com/

[36] https://www.kaggle.com/c/malware-classification/data

[37] https://www.sec.cs.tu-bs.de/~danarp/drebin/

[38] https://github.com/RECOVI/DroidWare

. . . continued

I	Name	Acc	DT	Cat	Size	Description
43	MIL-COM16	P	S	MAL	4S	Synthetic dataset with 4 botnet samples. The botnet actions in each sample differs from injection, reconnaissance, command and control (C&C) communication channels and botnet prorogation [143], [144]. Obtain dataset here [39]
44	Kharon	P	R	MAL	7 S	Kharon dataset contains malware documentation, that has been used to benchmark GroddDroid capability to trigger malicious code [145], [146]. The documentation was obtained though Static and dynamic analysis on a set of malware samples. The documentation includes the location of the malicious code blocks, the trigger conditions, and how the malware acts when triggered. Obtain dataset here [40]
45	Mudflow	P	R	MAL	18204 S	The Mudflow dataset was used to train a "benign or malign" binary classifier, on the information flow from benign Android applications obtained from the Google store [147], [148]. Instead of focusing on what resources the applications request access to, the classifier determines if the usage of these resources are to be deemed normal. The dataset contains 2866 benign and 15338 malicious apps. Obtain dataset and scripts here [41]
46	ISOT-2010	P	R	MAL	N/A	ISOT is a dataset that is built up of benign and malicious network traffic, from multiple sources [149], [150], [151]. The malicious samples is off the Storm and Waledac botnets. Obtain dataset here [42]
47	ECML/PKD-D07	R	H	MAL	50000 S	ECML/PKDD-2007: A dataset made up of 40 000 normal queries, 9000 exploits with descriptions on the target environment and 1000 exploits without target information [152]. The dataset is in XML format and contain attacks from 7 different categories. Obtain dataset here [43]

---

[39] `https://cybervan.appcomsci.com:9000/datasets`

[40] `http://kharon.gforge.inria.fr/dataset/`

[41] `https://www.st.cs.uni-saarland.de/appmining/mudflow/`

[42] `https://www.uvic.ca/engineering/ece/isot/datasets/index.php#section0-0`

[43] `http://www.lirmm.fr/pkdd2007-challenge/index.html#dataset`

. . . continued

I	Name	Acc	DT	Cat	Size	Description
48	CSIC10	P	R	MAL	610000 S	The motivation behind the creation of the HTTP-CSIC-2010 dataset, was a shortage of malware datasets that exploits real web applications [153]. HTTP-CSIC-2010 contains malign and benign labelled request against a Spanish electronic commerce web application. The samples with malicious consequence target the confidentiality and integrity of the web application resources. Obtain dataset here [44]
49	Blog-Pcap	P	N/A	MAL	1000 S	A list of 1000 Malware PCAP files. The blog also contain other malware samples as well [154], [155]. Obtain dataset here [45]
50	Malrec	P	R	MAL	24389 S	The Malrec Dataset: A dataset of the system calls and arguments made by malicious software [156]. Obtain dataset her [46]
51	CTU-13	P	R	MAL	334 S	Malware Capture Facility Project: A repository of Botnet and benign network traffic [157]. Obtain dataset here [47]
52	ISCX Botnet	R	H	MAL	N/A	ISCX Botnet: Is a derivative dataset from multiple sources [158], [159]. The dataset contains normal traffic and malicious traffic from 16 different families of botnets. Obtain dataset here [48]
53	ISCXAB	R	R	MAL	1929 S	ISCX Android Botnet (ISCXAB): This dataset is built up of AnserverBot, Bmaster, DroidDream, Geinimi, MisoSMS, NickySpy, Not Compatible, PJapps, Pletor, RootSmart, Sandroid, TigerBot, Wroba and Zitmo botnets in the form of Android application package (APK) files [160], [161]. Obtain dataset here [49]
54	DARPA-98/99	P	S	NET	38/50 S	DARPA 1998 and 1999 is datasets of simulated network traffic used to assess the detection capabilities of intrusion detection systems [162], [163]. DARPA 1998 contains 38 categories of UNIX based attacks. DARPA 1999 increases the number of categories to 50 and added Windows NT based exploits as well. Obtain dataset here [50]
55	DARPA-2000	P	S	NET	2 S	DARPA 2000 has simulated data from two distributed denial of service attacks [164]. Obtain dataset here [51]

---

[44] http://www.isi.csic.es/dataset/

[45] https://www.dropbox.com/sh/7fo4efxhpenexqp/AACmuri_l-LDiVDUDJ3hVLqPa?dl=0

[46] http://moyix.blogspot.no/search?q=dataset

[47] https://stratosphereips.org/category/dataset.html

[48] http://www.unb.ca/cic/research/datasets/botnet.html

[49] http://www.unb.ca/cic/research/datasets/android-botnet.html

[50] https://ll.mit.edu/ideval/data/index.html

[51] https://ll.mit.edu/ideval/data/2000data.html

...continued

I	Name	Acc	DT	Cat	Size	Description
56	MAWI-Lab	P	R	NET	N/A	The MAWILab database contains labels, that categorize network anomalies. It can be used to assist in evaluating the performance of intrusion detection systems (IDS) [165], [166]. Obtain dataset here [52]
57	KDD Cup99	P	S	NET	743M	KDD Cup 1999 Data: A synthetic dataset that is made up off network traffic samples [167]. These samples is labelled benign or malign [168]. Malign samples are attempting to attack availability, to perform privilege escalation, to imitating a local user and to perform reconnaissance. The dataset is produced based on the DARPA98 dataset. Obtain dataset here [53]
58	UNSW-NB15	P	H	NET	2,540,044 S	UNSW-NB15: The samples are labelled malign or benign. Each sample has 49 features that includes variables such as time to live (TTL), IP information, sequence number, time between TCP SYN and TCP ACK etc [169], [170]. The malicious samples aims to identify vulnerabilities by perform active reconnaissance and by using fuzzed inputs. The malware samples also attempts to install backdoors, target the availability of services, opening a shell to run arbitrary code and to compromise new hosts. There are in total 2 540 044 samples spread across 4 .csv files, a smaller subset of this dataset is used to create a training and a test set. Obtain dataset here [54]
59	NSA-CDX	P	R	NET	5 S	A collection of Cyber Defense Exercises (CDX) from the National Security Agency (NSA) [171]. In the CDX 2009 collection there are logs of DNS, web server, and IDS. Obtain datasets here [55]
60	ADFA	P	N/A	NET	N/A	The ADFA Intrusion Detection Datasets: According to the author in [172] the ADFA dataset has higher complexity, more attack options, frequency of attacks/normal traffic is more evenly distributed, and more extensive in scope then the KDD 98/99 evaluation datasets [173]. Given the reasons above the author concludes that the ADFA is the better dataset to assess the performance of Host based IDS (HIDS). Obtain datasets here [56]

---

[52] `http://www.fukuda-lab.org/mawilab/data.html`

[53] `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`

[54] `https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/`

[55] `http://www.usma.edu/crc/sitepages/datasets.aspx`

[56] `https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-IDS-Datasets/`

. . . continued

I	Name	Acc	DT	Cat	Size	Description
61	Kyoto data	P	R	NET	19683 MB	A numerical dataset that tracks network activity from Honeypots and sensors that is in the management control of Kyoto University [174], [175]. The network activity has been tracked from the end of 2006 to December 2015. The dataset includes 24 features, 14 of them is based on the KDD Cup 99 feature vector and the last 10 features was added to better describe what happens on the network. The latter 10 features are described below: • Binary valued features that cover the observation of IDS alerts, Malicious connections and exploits of the network traffic. • Abnormal session feature state if the network traffic is benign or is a known/unknown attack. • Sanitized IP address, ports and session information was also captured. Obtain dataset here [57]
62	crawdad	P	R	NET	N/A	crawdad is a compiled list of publicly available datasets of wireless protocols [176]. Obtain dataset here [58]
63	ICS-pcap	P	N/A	NET	N/A	ICS-pcap: A repository of ICS/SCADA pcaps gathered from multiple sources [177]. Obtain dataset here [59]
64	Common Crawl	R	R	NET	≈ 2,000, 000 S	Common Crawl: A dataset of web content and metadata from 2000000 crawled webpages [178]. Obtain dataset here [60]
65	NSL-KDD	R	S	NET	N/A	NSL-KDD dataset: Is a subset of KDD99 [168], [179] . Preproccsing that was performed on NSL-KDD: • Deduplication of training and testing set samples in NSL-KDD • The count of samples that is hard to classify in the NSL-kDD set, is inversely proportional to the percentage of hard samples in KDD. Obtain dataset here [61]

---

[57] `http://www.takakura.com/Kyoto_data/`

[58] `https://crawdad.org/all-byname.html`

[59] `https://github.com/automayt/ICS-pcap`

[60] `https://aws.amazon.com/public-datasets/common-crawl/`

[61] `http://www.unb.ca/cic/research/datasets/nsl.html`

. . . continued

I	Name	Acc	DT	Cat	Size	Description
66	ISCX-TNT	R	R	NET	22GB	ISCX Tor-nonTor (ISCXTNT): Contains both normal and Tor traffic [180], [181]. The real Tor traffic was generated from network, email and filesharing protocols. And audio, and video streams from popular applications. The samples/traffic is assign a class label for what application or protocol they belong to. Obtain dataset here [61]
67	ISCX-VNV	R	R	NET	28GB	ISCX VPN-nonVPN (ISCXVNV) dataset: Similar to the ISCX Tor-nonTor dataset [182], [183]. It has non-VPN traffic and labelled VPN traffic from multiple applications and protocols. Obtain dataset here [61]
68	ISCX-IDS	R	H	NET	2,297, 128 MB	ISCX IDS (ISCXIDS): This dataset includes real traffic for IPv4, IPv6, UDP, TCP, ARP, DNS, ICMP, HTTP, SMTP, SSH, IMAP, POP3, and FTP generated by using agents that simulate users interacting with these protocols [184], [185]. Obtain dataset here [61]
69	YPFC	P	R	PASS	N/A	Yahoo Password Frequency Corpus(YPFC): A sanitized password frequency corpus that protect the privacy of the user accounts [186], [187] . The scheme also protects up to two duplicate accounts, that has similar passwords. The sanitization is performed to prevent adversaries to gain knowledge of individual users. Obtain dataset here [62]
70	Vincent-Password	P	N/A	PASS	2,000, 000 S / 20 MB	Password dataset with 2000000 samples [188]. Obtain dataset here [63]
71	MBT08	P	R	RAM	N/A	Memory Buddies Traces(MBT): This dataset is built up of the memory contents from computers and servers. The dataset contains metadata of the underlying platform, metadata of live processes, and the hash values of the 4KB memory pages [189], [190], [191]. The server hosting this dataset also host network related datasets (PCAP, TCP traffic, synthetic attacks etc). Obtain dataset here [64]

---

[62] https://figshare.com/articles/Yahoo_Password_Frequency_Corpus/2057937

[63] http://www.datasciencecentral.com/forum/topics/password-dataset-for-you-to-test-your-data-science-skills

[64] http://skuld.cs.umass.edu/traces/cpumem/memtraces/

. . . continued

I	Name	Acc	DT	Cat	Size	Description
72	NUSSC	P	R	SMS	87300 S	The NUS SMS Corpus (NUSSC) includes 55835 English and 31465 Chinese SMS messages [192], [193]. To avoid bias or promote message diversity in the sampling process, the individual SMS messages was captured without considering any particular topic. The SMS messages can be download in JSON, XML and SQL format. Obtain dataset here [65]
73	Webb-SC11	P	R	SPAM	≈ 350000 S/ 1GB compressed	The Webb Spam Corpus 2011(WebbSC11): A custom crawler was built to collect spam web pages [194], [195]. The resulting collection was preprocessed to remove instances of legitimate websites and websites that could not get resolved. The dataset contains both the spam and the HTTP sessions for the spam servers. Obtain dataset here [66]
74	DITSSC	P	R	SPAM	1353 S	The samples from DIT SMS spam corpus (DITSSC) is a collection of reported SMS spam, by UK mobile users. The corpus is stored in a XML format. Unique entries in the collection was assured by performing case insensitive depublication [196], [197]. Obtain dataset here [67]
75	TREC-05-07	R	N/A	SPAM	3 C	Spam corpora from TREC. This server host 3 spam corpus from 2005-2007 [198]. Obtain dataset here [68]
76	Hewlett spam	P	R	SPAM	4601 S	A spam dataset created by Hewlett-Packard Labs [199]. The feature vector contains: • frequencies of words and characters, • the average length, max length and total count of "uninterrupted sequences of capital letters" • class label Obtain dataset here [69]
77	WEB-SPAM-UK07	P	R	SPAM	1,058, 965,55 S	WEBSPAM-UK2007: A labeled spam dataset of 105896555 entries [200]. Obtain dataset here [70]
78	micro-blog-PCU	P	R	SPAM	221579 S	microblogPCU Data Set: A labeled spam dataset [201]. Example features: • Microblog poster gender, username, user id • Number of followers • Number of reposts Obtain dataset here [71]

---

[65] https://github.com/kite1988/nus-sms-corpus

[66] https://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html

[67] http://www.dit.ie/computing/research/resources/smsdata/

[68] http://trec.nist.gov/data/spam.html

[69] http://archive.ics.uci.edu/ml/datasets/Spambase?ref=datanews.io

[70] http://chato.cl/webspam/datasets/uk2007/

[71] https://archive.ics.uci.edu/ml/datasets/microblogPCU

. . . continued

I	Name	Acc	DT	Cat	Size	Description
79	TREC-11	R	R	SPAM	1,600,000,0 S	TREC 2011 microblog dataset contains 16 million normal and spam twitter posts [202]. Obtain dataset here [72]
80	SPAM/ HAM	P	R	SPAM	273 MB/ 5780 S	Image Spam Dataset: contains images that is originated from a normal (HAM) email message or a spam message (SPAM) [203], [204]. This dataset can be used as a training and testing set in a SPAM or HAM classification problem. Obtain dataset here [73]
81	phishtank	P	R	PHI	N/A	phishtank is a open community where the users can add, search and validate instances of network phsising [205]. Search repository here [74]
82	millersmiles	P	R	PHI	N/A	millersmiles is a repository of phising samples and allows users to add to the collection or search though the repository [206]. Search repository here [75]
83	PWDS-15	P	R	PHI	2456 S	Phishing Websites Data: A collection of 2456 phishing websites [207], [208]. Example features in this dataset: • Long URL, Tiny url or abnormal url. • Special symbols in URL: '@', '//', and '-' • The count of dots in URL • Domain registration date • Protocol information and ports Obtain dataset here [76]

Table 6: Datasets

2.3 Systematic literature review 3 - Search capability comparison between open source search engines and open source forensic tools

2.3.1 Purpose of the literature review

(TODO: Have a research question here)

Inspect a set of open source search engines and, open source forensic tools that has search functionality, and identify and compare their various search capabilities.

2.3.2 Protocol/methodology

The candidate set of open source search engines

For the candidate list of open source search engines, the table 1 of search engines from review 1 was used. Due to time constraints only handful of these search engines will be used for further inspection. Therefore in table 7 the search engines have been ranked after a popularity criteria.The goal of the popularity criteria is not to find the best representation of the software popularity, but to get some estimate/rank/criteria that can

---

[72] `http://trec.nist.gov/data/tweets/`
[73] `http://www.cs.jhu.edu/~mdredze/datasets/image_spam/`
[74] `http://www.phishtank.com/phish_archive.php`
[75] `http://www.millersmiles.co.uk/`
[76] `http://archive.ics.uci.edu/ml/datasets/Phishing+Websites#`

be used to decide which software to first inspect and perform experiments on. The popularity criteria is made up of two parts. The first part is data from google trends over the last 12 months for each of the software names in table 7. Trends was calculated on 08-10-2017 with catagory "computers and electronics". Categories are used to filter out unrelated searches. Then for each software name entered in google trends, a .csv file was downloaded. The .csv files was opened in Microsoft Excel 2010 editor and the formula

=summer(B4:B55)

was used to calculate the sum estimated number of times the software name was searched during the 12 month period. This number represent a way to quantify user interest for the software. The second part is made up of how many articles between 2016 and 2017 the software name is mentioned alongside the phrase "search engine" in ieeexplore. Example search:

(Search engine) AND Dezi), year: 2016-2017

This value is a way to measure the scientific interest of these applications. The rows of table 7 are then sorted on the sum of these 2 values.

Table 7: A candidate list of open source search engines sorted on sum

Name	Searched by users (12 months)	ieeexplore mentions	Sum/criteria
Elasticsearch	4,649	114	4,763
Solr	3,826	85	3,911
Sphinx	3,718	40	3,758
Dezi	2,936	1	2,937
Luwak	2,438	0	2,438
groonga	2,058	0	2,058
Sifaka	1,669	0	1,669
tntsearch	1,595	0	1,595
Datafari	1,566	0	1,566
OpenSearchServer	1,361	0	1,361
OpenSemanticSearch	679	1	680
tantivy	574	0	574
pouchdb-quick-search	N/A	0	0

The candidate set of open source forensic tools

In the selection process for finding the open source forensic tools to inspect, a different criteria was used. The forensic software had to be open source, must have some search functionality, have some degree of documentation and preferably still being maintained. The search phrases in the list below was used on 06-10-2017 (DD-MM-YYYY) to scan google for compiled lists of open source forensic tools:

- "open source forensic tools" "keyword search"
- open source live forensics
- open source forensic string search
- forensic wiki keyword search open source

The resulting compiled lists of forensic tools are [209–215]. From those list a set of forensic tools was selected. The selected forensic tools can be seen in table 8

Table 8: The set of open source forensic tools for inspection

Source: sources here

Name	**Short description**	**Update**
The Sleuth Kit (Autopsy)		
Hachoir		
Volitility		
GRR Rapid Response		
TestDisk and PhotoRec		
bulk_extractor		
MIG: Mozilla InvestiGator		
guymager		
Rekall Memory Forensic Framework		
flare-floss		
inVtero.net		
wireshark		

The inspection set of open source forensic tools and open source search engines

Forensic tools : **TODO state reasons**

1. Sluth kit - Autopsy
2. Volatility
3. MIG
4. Hachoir
5. Wireshark

Search engines : **TODO state reasons**

1. Elasticsearch
2. Solr
3. Sphinx
4. Dezi

TODO: Mention elk... kyle porter mention that it is a popular tool `https://github.com/cvandeplas/ELK-forensics`

The inspection process

The inspection process will be performed as a combination of targeted manual inspection and keyword searching of technical documentation and source code comments of the software being inspected. Targeted manual inspection is looking at portions of the documentation more likely to include relevant information. Some documentation pages might be very old or indicate that the documented feature is experimental. These sources might get excluded from the review. One issue with the inspection process is that inferences often had to be made on out of context images and text that describes the search capabilities. And often is the given description quite short. Ideally the software capabilities would be confirmed by practical tests, but this may not be feasible due to time constraints.

List of planned to use keywords/things to look after during manual review:

- character/symbol limit (max/min)
- truncation character
- match
- regex
- grep

- Wildcard
- scan
- filter
- sort
- encode/encoding (names of encodings)
- index
- rank/ranking
- string, substring
- name list of string matching algorithms
- keyword
- fuzzy
- preproccesing
- encryption search
- compression of search data (list of compression algorithms)
- search storage (RAM, CACHE, disk)
- Terms
- Hits
- multi threading and search
- list of known search capabilities
- plugins, forks, addon, module
- query
- search
- deduplication/hashing
- Visualization of search results
- Customized search
- Parameters
- algorithms
- What can it search, where does it search
- look at advertised set of features
- Byte level search
- concurrent search
- save search results (file formats)
- Keyword lists
- automate
- Hash search
- File search
- Supported file types
- Unicode searching
- Wizard
- Skip known files when indexing
- Can you search while indexing?
- Exact match and substring match.
- Related pages in the documentation
- truncation search / stemming
- Sdhash, fuzzy hash matching, approximate hash based matching (AHBM)
- Report search

- search tree view
- search partitions, boot sector
- search recovery
- search highlighting
- unallocated storage
- search archived files
- email search
- binary search / sequential search
- Periodical search
- rule/criteria based search
- Stemming
- Phonetic search
- Faceted search
- Semantic search
- fuzzy search
- Exhaustive search
- byte comparison / character comparison
- Boolean search
- Language detection
- structure of the search (e.g. string, JSON object)
- **Masked results (e.g. meta data) / obfuscation of sensitive data.**
- Stripping - removal of sensitive data
- Clustering of search results
- Scrolling
- Relevance score - (TF/IDF)
- Summary / Aggregated results
- "More like this" quary
- Index and field boosting (weighting)
- boolean operators
- Fixed relevancy score
- export search result

2.3.3 Inspection summary forensic tools

Inspection of Sluth kit - Autopsy

Name: sleuthkit autopsy

Sources: [216–232]

Positive methodology:

- Searches google on 08-10-2017 (DD-MM-YYYY) for:
  - Autopsy navigation tree
  - hash hits autopsy
  - sleuthkit autopsy search
  - keyword search ingest
- Searches google on 09-10-2017 (DD-MM-YYYY) for:
  - sluthkit case insensitive
  - symbol limit sleuthkit
  - EBCDIC encoding sleuthkit
  - sleuthkit keyword
- Looked at advertised features
- Looked at Autopsy User's Guide
  - Manual Analysis
  - Automated Analysis (Modules)
  - Reporting

Inspection Result

- Keyword search:
  - Bases on Apache Solr
  - Support for concurrent search on the same index
  - Do not perform byte level search, but preprocess the documents with Tika and perform the search on the output of this process.
  - Can automate search by creating lists of keywords. Keywords list can be imported/exported.
  - Supports both exact and substring matching. Substring matching is will not work with spaces and punctuation characters.
  - Users can decide between case sensitive and case insensitive search.
  - Can create HTML and Excel reports of search hits
  - Text Content Viewer: provides keyword match highlighting in matched files
  - Periodical searches are supported
- Regular expression:
  - Includes predefined regular expressions for emails, telephone numbers, URL and IP
  - Based on a perl implementation
- Rule based search allows to set a search criteria such as the file name, if files or directory should be searched, file extension, directory path
- Index:
  - First strings are searched to be extracted and then index. The default settings are english and UTF8 and UTF16. But these settings can be changed. UTF16LE and UTF16BE are also supported for string extraction.
  - EBCDIC are **NOT** supported.
  - Searches are done on indexes
  - There is a option to skip known files (HASH) when indexing (deduplication).
  - Can search while indexing (index is incomplete)
- Supported files:
  - RTF
  - PDF
  - HTML

  - DOC, DOCX
- File/directory search:
  - can search on file name
  - Can provide a date interval [from,to] and match the files that has their file attributes modified, accessed, changed or created file attributes within this interval.
  - Can search for files that matches a size criteria
  - Can search for known bad hashes, known hashes and unknown hashes. This is made possible with a hash database. Hash lookups is done by binary search.
  - Can search for deleted files.
  - can search unallocated storage
  - can search archived files
- Can search for partitions and boot sector
- Search results:
  - Store results as XML or HTML?
- 3rd party software/plugins/modules with search capability:
  - Approximate Hash Based Matching (AHBM) or fuzzy hash matching with sdhash
  - Reference database: VirusTotalOnlineChecker is a addon to Autopsy and can check the hash of files against the VirusTotal Database.
  - PTK is a addon for sluethkit that allows keyword search of memory dumps.

Inspection of Volatility

Name: Volatility

Sources: [233–243]

Positive methodology:

- Looking for the word scan in the "doxygen-generated" manual
- Searched google on 11-10-2017 (DD-MM-YYYY) for
    - BaseScanner volatility
    - volitility framework regex
- search Volatility github repository on 11-10-2017 (DD-MM-YYYY) for
    - search
- Looking at the Command-Reference-Mal in github repository for Volatility

Inspection Result

- RAM scans:
    - PSDispScan searched physical RAM for _EPROCESS data structures. This can reveal information of killed and hidden processes.
    - MultiPoolScanner can find many different types of pooltags using different pool tag scanners.
    - BaseScanner is a type of exhaustive search that process one and one byte, it is a more general scanner.
    - malfind: Looks for malware in processes by searching for VAD tags and permissions.
    - kdbgscan and kpcrscan searches for values that look like kdbg abd kpcr values
    - Have scans for tcp connections, files, synlinks, drivers, sockets etc.
- Regular expression:
    - Yarascan allows making search based on Regular expressions, wildcards or **YARA rules**. Search by yarascan can be made **case insensitive**
- The find_module function uses **binary search on sorted input** to find the mapping between modules and virtual memory.
- Evolve is a interface of Volatility that can be run in network browser. Evolve enables SQL queries to be performed on the data from the scans, and searching the table view of the results.
- The html renderer allows you to view the output in a browser in a table view and sort by any field or search the output.

Inspection of Mozilla InvestiGator

Name MIG: Mozilla InvestiGator
Sources: [244–247]
Positive methodology:

- Looked though:
  - Mozilla InvestiGator: File module
  - "Concepts & Internal Components" documentation.
  - Mozilla InvestiGator: Memory module
- searched on MIG github repository on 11-10-2017 (DD-MM-YYYY) for
  - search

Inspection Result

- Can use a target field to search for certain "agents", which are a software that interface between MIG and the remote host.
- Support for regular expressions.
  - For file search there is a option for setting a criteria that all records within a file need to match a regular expression.
  - regexes support UTF-8 encoding
- Searches are structured as a JSON object
- Have a non default option of getting masked meta data from searches.
- Does not follow directory links as this can lead to loops.
- can search for files with content that match a MD5, SHA1, SHA2 or SHA3 hash
- search filters:
  - Can filter search on files based on name, extension, size, set of permissions for the file and the modification time attribute. Multiple filters can be applied for the same search (form of boolean search). The default behaviour is that not all filters have to match, but this can be changed.
  - A option for retrieving all files that did not match the filters
  - Can limit the number of search hits and constrain the search to only process directories on certain hierarchical levels (will not process a subdirectory that is further down the hierarchy than the limit).
- RAM search:
  - Sequential search can be performed on multiple buffered memory regions. Have the option to jump over memory or terminate the search after x number of characters have been read.
  - can search for matching bytes strings

Inspection of Hachoir

Name: Hachoir
Sources: [248–251]
Positive methodology:

- Looked though documentation pages:
  - hachoir-metadata program
  - hachoir-strip program
  - hachoir-grep program
  - hachoir-subfile program
  - Hachoir3 for developers

Inspection Result

- Uses the Lucene library for fulltext search
- Can search for images in general or for specific image filetype. Searches can jump to read from a given byte point and files size can be used as a search criteria.
- The program has grep functionality that works with string data, it can print all strings in a file, search for case sensitive, or case insensitive strings in file. This functionality comes with a high memory cost.
- Have the ability to remove sensitive metadata (stripping)
- Supports the ISO-8859-1, UTF-8, UTF-16 encodings.
- Support for regular expression

2.3.4 Inspection summary search engines

Inspection of Elasticsearch

Name: Elasticsearch

Sources: [252–284]

Positive methodology:

- Looked though documentation pages for Elasticsearch version 5.6:
  - Search APIs
  - Aggregations
  - Indices APIs
  - Query DSL
- Searched google on 16-10-2017 (DD-MM-YYYY) for:
  - elasticsearch compression index
  - elasticsearch deduplication index hashing
  - Elasticsearch stemming
  - elasticsearch case search
- Searched google on 20-10-2017 (DD-MM-YYYY) for:
  - elasticsearch field boosting
  - elasticsearch export search result

Inspection Result

- Elasticsearch uses distrubuted search and can search specific indexes/shards and retrieve all documents written by a given author, or retrieve all documents of a given type. There is also a option to search all indexes. If indexes contain more important information than others, then you can elevate priority of these indexes. The indexes involved in a given search can also be printed. Indexes are compressed using the DEFLATE algorithm in order to minimize the storage requirements.
- Option to terminate a search process during execution by a command, by a elapsed time threshold or by setting a max number of documents to be retrieved.
- Can sort search results on multiple search fields. For numerical values sum, max, min, average and median can be sorted on. Sorting on latitude and longitude in some instances are also possible. Field collapsing can be combined with sorting to get only the leading sorted documents on the collapsed field.
- Support for wildcard in search for both extending and limiting the scope of the search.
- Elasticsearch allows to customize what is returned from a search (message, values, etc) with Script Fields. Can compile a summary of the search results by using aggregations in ElasticSearch.
- With post filter ElasticSearch can narrow search results based on membership status. For example post filter can narrow search results that have the same producer, product line, and colour.
- Multiple fields can be selected to highlight search hits on.
- Can customize how scrolling is done with the search results. Alternatively the set of current search results can be used to retrieve the next batch of search hits.
- For calculating the relevancy of documents ElasticSearch uses term frequency/inverse document frequency (TF/IDF). TF/IDF considers how often the search term is in the documents, gives higher weight for more uncommon search terms in the index and gives terms matching with shorter fields higher pri-

ority then matching long fields. The relevance score can be used as a threshold to exclude less relevant search hits. It is also possible to set the weight of importance for specific fields.

- With the Profile parameter the user can get debug information regarding the performance of the search
- Support for boolean quary against numerical data, matching words, type, matching prefix (similar to substring matching) on terms, and range data. The boolean quary supports AND, OR, NOT, grouping operators with '()', joining quires etc. An alternative to the not operation, is using a list of matching terms that should get reduced relevancy score. The queries can also use regular expression and wildcard matching. By setting the Levenshtein Edit Distance the boolean queries can get results based on fuzzy matching. MoreLikeThis quary identifies similar documents to those the user lists.
- Can delete some or all cache data for indexed documents.
- Multi threading support as well as support for multiple concurrent searches.
- Little support for deduplication, you can search for matching fields as indicators for duplicate files. Deduplication was not identified as a advertised feature of ElasticSearch during the inspection.
- Good support for stemming (mapping to root of a word)
- Support for case insensitive and case sensitive search by using or omitting the lowercase function.
- A feature that ElasticSearch lack is the ability to export search results

Inspection of Solr

Name: Solr
Sources: [285–301]
Positive methodology:

- Looked though pages in Solr Ref Guide 7.0
  - Searching
- Searched google on 20-10-2017 (DD-MM-YYYY) for:
  - solr deduplication
- Searched google on 21-10-2017 (DD-MM-YYYY) for:
  - solr regex
  - solr character limit
  - solr substring match
  - solr tf idf
  - solr language detection

Inspection Result

- Possible to highlight matching search terms in documents with coloured borders
- Support for clustering search results in labelled clusters.
- Just like ElasticSearch Solr also support MoreLikeThis queries.
- Search hits can be be sorted on all non-multi valued fields.
- It is possible for the user to reduce the search results by ignoring the first x documents in the result set, or by setting a maximum number of allowed results. A filter query can also be used to specify multiple conditions to reduce the size of the resulting set.
- A user can set a max elapsed time for searching, search processes that lasts longer than this threshold is terminated.
- Solr enables users to customize the information presented with the result set.
- Support for wildcard characters, fuzzy matching using Levenshtein Distance and setting a minimum of criterias that have to match.
- A user can set the importance of a given field. Unlike ElasticSearch Solr can also set a fixed relevancy score for any documents that match a search term, the presence of more than 1 matching terms in the results set would be irrelevant for the relevancy score.
- Support matching documents on numerical and text fields that fall within a specified interval.
- Solr supports the distinct boolean operators AND, OR, NOT, +(single term have to be present in document) and Grouping with parenthesis.
- With Solr you can export a sorted result set in JSON format.
- Solr has a feature for recommending key terms to user, that they can use to search.
- Solr have hash signatures and fuzzy matching to detect duplicated documents. These can be used for deduplication.
- Solr field collapsing works similar to elasticSearch
- Solr support documents encoded and indexes using UTF-8 encoding.
- Solr supports regular expressions.

- EdgeNGramFilterFactory in Solr can be used for sub-strings matching.
- Solr uses TF-IDF just as ElasticSearch
- At index time Solr can figure out the language of the documents being indexed.

Inspection of Sphinx

Name:Sphinx
Sources: [302–311]
Positive methodology:

- Looked though pages in Sphinx 2.2.11-release reference manual
  - searching
  - Additional functionality
- Searched google on 22-10-2017 (DD-MM-YYYY) for:
  - sphinx regex

Inspection Result

- For Boolean search Sphinx support AND, OR, NOT and grouping by parenthesis.
- Sphinx can use relevance ranking schemes for search hits: TF-IDF, user defined field importance (as in Solr and ElasticSearch), if the search terms are in the same order in the search string and the documents, the count of distinct matching terms, overall number of matched terms etc.
- It is possible to sort on 1 to 5 document fields, sorting on time or by a customized math function.
- Can cluster search hits on fields or by time information
- For scaling Sphinx allows users to manually setting up distributed searching.
- Support regular expressions, wildcards and sub-string matching.
- Can set the number of possible concurrent searches.
- Can process UTF-8 encoding.
- Matching terms highlighting in documents

Inspection of Dezi

Name: Dezi

Sources:

Positive methodology:

- a

Inspection Result

- a

2.3.5 Search capability comparison

Application name	test 1
a	b

3 Choice of methods

4 Experimental design

5 Performing the experiments and results

6 General discussion and conclusion

6.1 Discussion

Answering the sub question: 1.a)

is difficult as there is a lack of recent surveys of the usage of search in a Digital Forensic setting. The systematic literature review in section 2.1 attempted to answers this sub question by looking at scientific literature published in 2014-2017 that used search in some capacity in their digital forensic experimental designs, discussion etc. It covered topics like:

- Regulations and search,
- searching for evidence in junk,
- searchable hash databases,
- reference databases,
- approximate hash based matching,
- inexact search,
- de-duplication,
- searching RAM,
- visualization of search results,
- issues with keyword search,
- clustering of search results,
- search suggestion,
- searching after deleted files,
- search in file carving
- and issues with search and encodings.

The topic list covers a wide range of the usage of search in Digital Forensics, but there is undoubtedly more to cover. Future work could tackle this subquestion by looking at

- scientific articles on Digital forensics that matches the key terms "string, keyword, query" and "matching"
- Look at older scientific articles
- Use different resources such as books, whitepapers and interviews with forensic practitioners.

Answering the sub question: 1.d)

Selection of the open source forensic tools and open source search engines was done based on popularity and how well documented these applications where. The forensic tools can be divided into categories such as RAM search, Live inspection etc. Ideally 2 forensic tools of every catagory would be selected for inspection, so that forensic practioners could know what tool of which catagory would be most applicable for their needs. But this was not feasible due to time constraints. The inspection of the selected applications was thorough, comprehensive and revealed multiple differences between the search capabilities of the applications. The compiled information from the inspection shows the search features and how this information was obtained. A subset of this information is also presented in a easy accessible table view that compares the different applications. While this information is useful for forensic practitioners deciding on which search tools to use, it is not without limitations. Many of the sources are good, but some of the sources that this information is compiled from lacks context, have little content, and are potentially outdated. One issue that came up often was determining if a search feature was a built-in functionality or something that you could theoretically implement yourself in the system using the application API. A future study could verify these search capabilities by performing practical experiments. These experiments would improve the confidence of these findings.

Low level algorithms

A lot of resources have been invested in finding details about what string matching search algorithms the open source search engines and open source forensic tools was using. This was a point of interest as this could partly explain why the applications are performing the way they do, and how they can be improved. This information is not easily accessible. Searching was done to find relevant scientific papers, white papers, web pages, searching github repositories after comments and source code and manual inspection of the source code. This searching and manual review of source code did not give much results, and this process had to be terminated due to time constraints. A future paper could look into using benchmark scripts written specifically for the applications and application specific benchmark API that can output details about underlying algorithms.

Bibliography

[1] Zawoad S, Hasan R. Digital Forensics in the Age of Big Data: Challenges, Approaches, and Opportunities. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems; 2015. p. 1320–1325.

[2] Yafooz WMS, Abidin SZZ, Omar N, Idrus Z. Managing unstructured data in relational databases. In: 2013 IEEE Conference on Systems, Process Control (ICSPC); 2013. p. 198–203.

[3] Li Y, Liu Z, Zhu H. Enterprise Search in the Big Data Era: Recent Developments and Open Challenges. Proc VLDB Endow. 2014 Aug;7(13):1717–1718. Available from: http://dx.doi.org/10.14778/2733004.2733071.

[4] Palmer G, Corporation M. A Road Map for Digital Forensic Research; 2001. Accessed 29.04.17. Available from: http://dfrws.org/sites/default/files/session-files/a_road_map_for_digital_forensic_research.pdf.

[5] Armknecht F, Dewald A. Privacy-preserving email forensics. Digital Investigation. 2015;14, Supplement 1:S127 – S136. The Proceedings of the Fifteenth Annual {DFRWS} Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287615000481.

[6] Martini B, Choo KKR. Distributed filesystem forensics: XtreemFS as a case study. Digital Investigation. 2014;11(4):295 – 313. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000942.

[7] Yu S. Covert communication by means of email spam: A challenge for digital investigation. Digital Investigation. 2015;13:72 – 79. Available from: http://www.sciencedirect.com/science/article/pii/S1742287615000432.

[8] Garfinkel SL, McCarrin M. Hash-based carving: Searching media for complete files and file fragments with sector hashing and hashdb. Digital Investigation. 2015;14, Supplement 1:S95 – S105. The Proceedings of the Fifteenth Annual {DFRWS} Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287615000468.

[9] Thongjul S, Tritilanunt S. Analyzing and searching process of internet username and password stored in Random Access Memory (RAM). In: 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE); 2015. p. 257–262.

[10] Bjelland PC, Franke K, Årnes A. Practical use of Approximate Hash Based Matching in digital investigations. Digital Investigation. 2014;11, Supplement 1:S18 – S26. Proceedings of the First Annual {DFRWS} Europe. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000085.

[11] Farhadi MR, Fung BCM, Fung YB, Charland P, Preda S, Debbabi M. Scalable code clone search for malware analysis. Digital Investigation. 2015;15:46 – 60. Special Issue: Big Data and Intelligent Data Analysis. Available from: http://www.sciencedirect.com/science/article/pii/S1742287615000705.

[12] Wang WB, Huang ML, Lu L, Zhang J. Improving Performance of Forensics Investigation with Parallel Coordinates Visual Analytics. In: 2014 IEEE 17th International Conference on Computational Science and Engineering; 2014. p. 1838–1843.

[13] Sharif SA, Ali MA, Reqabi NA, Iqbal F, Baker T, Marrington A. Magec: An Image Searching Tool for Detecting Forged Images in Forensic Investigation. In: 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS); 2016. p. 1–6.

[14] Pollitt M. Chapter 2 - The key to forensic success: examination planning is a key determinant of efficient and effective digital forensics. In: Sammons J, editor. Digital Forensics. Boston: Syngress; 2016. p. 27 – 43. Available from: `http://www.sciencedirect.com/science/article/pii/B9780128045268000022`.

[15] Rogers MK. Chapter 3 - Psychological profiling as an investigative tool for digital forensics. In: Sammons J, editor. Digital Forensics. Boston: Syngress; 2016. p. 45 – 58. Available from: `http://www.sciencedirect.com/science/article/pii/B9780128045268000034`.

[16] Mascarnes S, Lopes P, Sakhare P. Search model for searching the evidence in digital forensic analysis. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT); 2015. p. 1353–1358.

[17] Pollitt MM. Triage: A practical solution or admission of failure. Digital Investigation. 2013;10(2):87 – 88. Triage in Digital Forensics. Available from: `http://www.sciencedirect.com/science/article/pii/S1742287613000030`.

[18] Overill RE, Silomon JAM, Roscoe KA. Triage template pipelines in digital forensic investigations. Digital Investigation. 2013;10(2):168 – 174. Triage in Digital Forensics. Available from: `http://www.sciencedirect.com/science/article/pii/S1742287613000261`.

[19] Moser A, Cohen MI. Hunting in the enterprise: Forensic triage and incident response. Digital Investigation. 2013;10(2):89 – 98. Triage in Digital Forensics. Available from: `http://www.sciencedirect.com/science/article/pii/S1742287613000285`.

[20] Case A, III GGR. Memory forensics: The path forward. Digital Investigation. 2017;20:23 – 33. Special Issue on Volatile Memory Analysis. Available from: `http://www.sciencedirect.com/science/article/pii/S1742287616301529`.

[21] III GGR, Case A. In lieu of swap: Analyzing compressed {RAM} in Mac {OS} X and Linux. Digital Investigation. 2014;11, Supplement 2:S3 – S12. Fourteenth Annual {DFRWS} Conference. Available from: `http://www.sciencedirect.com/science/article/pii/S1742287614000541`.

[22] Sylve JT, Marziale V, III GGR. Pool tag quick scanning for windows memory analysis. Digital Investigation. 2016;16, Supplement:S25 – S32. {DFRWS} 2016 EuropeProceedings of the Third Annual {DFRWS} Europe. Available from: `http://www.sciencedirect.com/science/article/pii/S1742287616000062`.

[23] Lapso JA, Peterson GL, Okolica JS. Whitelisting system state in windows forensic memory visualizations. Digital Investigation. 2017;20:2 – 15. Special Issue on Volatile Memory Analysis. Available from: `http://www.sciencedirect.com/science/article/pii/S1742287616301438`.

[24] Mohamed AFAL, Marrington A, Iqbal F, Baggili I. Testing the forensic soundness of forensic examination environments on bootable media. Digital Investig-

ation. 2014;11, Supplement 2:S22 – S29. Fourteenth Annual {DFRWS} Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000589.

[25] Attoe R. Chapter 6 - Digital forensics in an eDiscovery world. In: Sammons J, editor. Digital Forensics. Boston: Syngress; 2016. p. 85 – 98. Available from: http://www.sciencedirect.com/science/article/pii/B978012804526800006X.

[26] Beebe NL, Liu L. Clustering digital forensic string search output. Digital Investigation. 2014;11(4):314 – 322. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614001108.

[27] Lees C. Determining removal of forensic artefacts using the {USN} change journal. Digital Investigation. 2013;10(4):300 – 310. Available from: http://www.sciencedirect.com/science/article/pii/S1742287613001084.

[28] Leimich P, Harrison J, Buchanan WJ. A {RAM} triage methodology for Hadoop {HDFS} forensics. Digital Investigation. 2016;18:96 – 109. Available from: http://www.sciencedirect.com/science/article/pii/S1742287616300780.

[29] Wagner J, Rasin A, Grier J. Database image content explorer: Carving data that does not officially exist. Digital Investigation. 2016;18, Supplement:S97 – S107. Available from: http://www.sciencedirect.com/science/article/pii/S1742287616300500.

[30] Anwar T, Abulaish M. A social graph based text mining framework for chat log investigation. Digital Investigation. 2014;11(4):349 – 362. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614001091.

[31] Minnaard W. Out of sight, but not out of mind: Traces of nearby devices' wireless transmissions in volatile memory. Digital Investigation. 2014;11, Supplement 1:S104 – S111. Proceedings of the First Annual {DFRWS} Europe. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000188.

[32] Mathew LM, R S, Kizhakkethottam JJ. A survey on different video restoration techniques. In: 2015 International Conference on Soft-Computing and Networks Security (ICSNS); 2015. p. 1–3.

[33] Stewart J, Uckelman J. Unicode search of dirty data, or: How I learned to stop worrying and love Unicode Technical Standard number 18. Digital Investigation. 2013;10, Supplement:S116 – S125. The Proceedings of the Thirteenth Annual {DFRWS} Conference13th Annual Digital Forensics Research Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287613000595.

[34] karpet. karpet/Dezi; 2016. Accessed on 20.03.2017. Available from: https://github.com/karpet/Dezi.

[35] Karman P. Dezi::Config;. Accessed 23.04.17. Available from: https://metacpan.org/pod/Dezi::Config.

[36] Karman P. Dezi::Aggregator::DBI;. Accessed 23.04.17. Available from: https://metacpan.org/pod/Dezi::Aggregator::DBI.

[37] Apache. Index of /lucene/solr/6.4.2; 2017. Accessed on 22.03.2017. Available from: http://apache.uib.no/lucene/solr/6.4.2/.

[38] Targett C. Spatial Search; 2017. Accessed 23.04.17. Available from: https://cwiki.apache.org/confluence/display/solr/Spatial+Search.

[39] Bernstein J. Streaming Expressions; 2017. Accessed 23.04.17. Available

from: https://cwiki.apache.org/confluence/display/solr/Streaming+Expressions.

[40] Targett C. Faceting; 2017. Accessed 23.04.17. Available from: https://cwiki.apache.org/confluence/display/solr/Faceting.

[41] Sphinx. Sphinx 2.3.2-beta downloads; 2016. Accessed on 22.03.2017. Available from: http://sphinxsearch.com/downloads/beta/.

[42] Sphinx. Sphinx 2.3.2-beta reference manual; 2016. Accessed 23.04.17. Available from: http://sphinxsearch.com/docs/devel.html#searching.

[43] Lemur. The Lemur Project; 2017. Accessed on 22.03.2017. Available from: https://sourceforge.net/projects/lemur/.

[44] lemur project. Sifaka; 2016. Accessed 23.04.17. Available from: http://www.lemurproject.org/sifaka.php.

[45] Opensearchserver. Configuring facets;. Accessed 24.04.17. Available from: http://www.opensearchserver.com/documentation/clients/php_client/facets.md.

[46] emmanuel keller. OpenSearchServer; 2017. Accessed on 22.03.2017. Available from: https://github.com/jaeksoft/opensearchserver.

[47] OpenSearchServer. Downloads and documentation; 2017. Accessed on 22.03.2017. Available from: http://www.opensearchserver.com/.

[48] romseygeek. flaxsearch/luwak; 2017. Accessed on 26.03.2017. Available from: https://github.com/flaxsearch/luwak.

[49] Julien. Advanced search feature (Datafari 3.2 and above); 2017. Accessed 24.04.17. Available from: https://datafari.atlassian.net/wiki/pages/viewpage.action?pageId=61282866.

[50] julienFL. francelabs/datafari; 2017. Accessed on 26.03.2017. Available from: https://github.com/francelabs/datafari.

[51] ElasticSearch. Full text search;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/guide/current/full-text-search.html.

[52] ElasticSearch. Facets;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-facets.html.

[53] ElasticSearch. Geo Distance query;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-geo-distance-query.html.

[54] ElasticSearch. Fuzzy query;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-fuzzy-query.html.

[55] ElasticSearch. Phonetic Matching;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/guide/current/phonetic-matching.html.

[56] elasticsearch. elasticsearch; 2017. Accessed 24.04.17. Available from: https://github.com/elastic/elasticsearch.

[57] Groonga. Characteristics of Groonga; 2017. Accessed 24.04.17. Available from: http://groonga.org/docs/characteristic.html#groonga-overview.

[58] Groonga. The latest release; 2017. Accessed 24.04.17. Available from: http://groonga.org/.

[59] tantivy. tantivy; 2017. Accessed 24.04.17. Available from: https://github.com/tantivy-search/tantivy.

[60] TNTsearch. TNTsearch; 2017. Accessed 24.04.17. Available from: https://github.com/teamtnt/tntsearch.

[61] pouchdb-quick search. pouchdb-quick-search; 2017. Accessed 24.04.17. Available from: https://github.com/nolanlawson/pouchdb-quick-search.

[62] Search OS. Open Semantic Search;. Accessed 25.04.17. Available from: https://www.opensemanticsearch.org/.

[63] Search OS. Open Semantic Search; 2017. Accessed 25.04.17. Available from: https://github.com/opensemanticsearch/open-semantic-search-apps.

[64] Krellenstein M. Starting a Search Application; 2009. Accessed 25.04.17. Available from: https://whitepapers.em360tech.com/wp-content/files_mf/white_paper/lucid2.pdf.

[65] Lucidworks. Full Text Search Engines vs. DBMS (whitepaper);. Accessed 25.04.17. Available from: https://lucidworks.com/2009/09/02/full-text-search-engines-vs-dbms/.

[66] Guarino A. In: Reimer H, Pohlmann N, Schneider W, editors. Digital Forensics as a Big Data Challenge. Wiesbaden: Springer Fachmedien Wiesbaden; 2013. p. 197–203. Available from: http://dx.doi.org/10.1007/978-3-658-03371-2_17.

[67] Cleverley PH, Burnett S. Retrieving haystacks: a data driven information needs model for faceted search. Journal of Information Science. 2015;41(1):97–113. Available from: http://dx.doi.org/10.1177/0165551514554522.

[68] Li J, Wang Q, Wang C, Cao N, Ren K, Lou W. Fuzzy Keyword Search over Encrypted Data in Cloud Computing. In: 2010 Proceedings IEEE INFOCOM; 2010. p. 1–5.

[69] Ji S, Li G, Li C, Feng J. Efficient Interactive Fuzzy Keyword Search. In: Proceedings of the 18th International Conference on World Wide Web. WWW '09. New York, NY, USA: ACM; 2009. p. 371–380. Available from: http://doi.acm.org/10.1145/1526709.1526760.

[70] Solutions VI. Approximate Matching (whitepaper); 2008. Accessed 25.04.17. Available from: http://viewds.com/images/pdf/Whitepapers/approximate%20matching%202.pdf.

[71] Elmes GA, Roedl G, Conley J. Forensic GIS: The Role of Geospatial Technologies for Investigating Crime and Providing Evidence. Springer Publishing Company, Incorporated; 2014.

[72] Selvi RT, Raj EGDP. An Approach to Improve Precision and Recall for Ad-hoc Information Retrieval Using SBIR Algorithm. In: 2014 World Congress on Computing and Communication Technologies; 2014. p. 137–141.

[73] Cai J, Shao X, Ma W. Ontology Driven Semantic Search over Structure P2P Network. In: 2009 Ninth International Conference on Hybrid Intelligent Systems. vol. 3; 2009. p. 29–34.

[74] Bonino D, Corno F, Farinetti L, Bosca A. Ontology Driven Semantic Search. In: WSEAS International Journal Multimedia and Image Processing (IJMIP), Volume 2, Issues 1/2, March/June 2012 Copyright © 2012, Infonomics Society 148 Transaction on Information Science and Application, Issue 6; 2004. p. 1597–1605.

[75] Kleppmann M. real time full text search with luwak and samza; 2015. Ac-

cessed 29.04.17. Available from: https://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/.

[76] Data I. Building a Streaming Search Platform; 2016. Accessed 29.04.17. Available from: https://blog.insightdatascience.com/building-a-streaming-search-platform-61a0d5a323a8.

[77] Lillis D, Scanlon M. In: Park JJJH, Jin H, Jeong YS, Khan MK, editors. On the Benefits of Information Retrieval and Information Extraction Techniques Applied to Digital Forensics. Singapore: Springer Singapore; 2016. p. 641–647. Available from: http://dx.doi.org/10.1007/978-981-10-1536-6_83.

[78] Yannikos Y, Graner L, Steinebach M, Winter C. In: Peterson G, Shenoi S, editors. Data Corpora for Digital Forensics Education and Research. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014. p. 309–325. Available from: https://doi.org/10.1007/978-3-662-44952-3_21.

[79] Grajeda C, Breitinger F, Baggili I. Availability of datasets for digital forensics. And what is missing. Digital Investigation. 2017;22(Supplement):S94 – S105. Available from: http://www.sciencedirect.com/science/article/pii/S1742287617301913.

[80] Grajeda C, Breitinger F, Baggili I. DATASETS FOR CYBER FORENSICS; 2017. Last accessed (DD/MM/YYYY) 19/09/2017. Available from: http://datasets.fbreitinger.de/datasets/.

[81] Abt S, Baier H. Are We Missing Labels? A Study of the Availability of Ground-Truth in Network Security Research. In: Proceedings of the 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security. BADGERS '14. Washington, DC, USA: IEEE Computer Society; 2014. p. 40–55. Available from: http://dx.doi.org/10.1109/BADGERS.2014.11.

[82] Schler J, Koppel M, Argamon S, Pennebaker J. In: Effects of age and gender on blogging. vol. SS-06-03; 2006. p. 191–197. Last accessed (DD/MM/YYYY) 22/09/2017. Available from: http://u.cs.biu.ac.il/~schlerj/schler_springsymp06.pdf.

[83] u cs biu ac il. The Blog Authorship Corpus;. Last accessed (DD/MM/YYYY) 22/09/2017. Available from: http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm.

[84] Nunes E, Shakarian P, Simari GI, Ruef A. Argumentation Models for Cyber Attribution. CoRR. 2016;abs/1607.02171. Last accessed (DD/MM/YYYY) 22/09/2017. Available from: https://arxiv.org/pdf/1607.02171v1.pdf.

[85] Nunes E. CTF data;. Last accessed (DD/MM/YYYY) 22/09/2017. Available from: https://www.dropbox.com/sh/17d4eyg0cwoxg8s/AAA5g1NvQw-tUoZPvldloddRa?dl=0.

[86] Marcinczuk M, Zasko-Zielinska M, Piasecki M. Structure Annotation in the Polish Corpus of Suicide Notes. In: Text, Speech and Dialogue - 14th International Conference, TSD 2011, Pilsen, Czech Republic, September 1-5, 2011. Proceedings; 2011. p. 419–426. Available from: https://doi.org/10.1007/978-3-642-23538-2_53.

[87] Zaśko-Zielińska M, Piasecki M, Marcińczuk M. Polski korpus listów pożegnalnych samobójców; 2015. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: http://www.pcsn.uni.wroc.pl/.

[88] Brennan M, Greenstadt R. Practical Attacks Against Authorship Recognition Techniques. In: Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference, IAAI-09; 2009. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: https://www.cs.drexel.edu/~greenie/brennan_paper.pdf.

[89] psal cs drexel edu. JStylo-Anonymouth; 2013. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: https://psal.cs.drexel.edu/index.php/JStylo-Anonymouth.

[90] Luyckx K, Daelemans W. Personae: a Corpus for Author and Personality Prediction from Text. In: LREC. European Language Resources Association; 2008. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: http://www.lrec-conf.org/proceedings/lrec2008/pdf/759_paper.pdf.

[91] Luyckx K, Daelemans W. Personae Corpus; 2017. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: https://www.clips.uantwerpen.be/datasets/personae-corpus.

[92] Argamon S, Juola P. Overview of the International Authorship Identification Competition at PAN-2011. In: Petras V, Forner P, Clough P, editors. Notebook Papers of CLEF 2011 Labs and Workshops, 19-22 September, Amsterdam, Netherlands; 2011. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: http://www.uni-weimar.de/medien/webis/events/pan-11/pan11-papers-final/pan11-author-identification/argamon11-overview.pdf.

[93] pan webis de. Evaluation Data; 2016. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: http://pan.webis.de/data.html.

[94] Juola P. An Overview of the Traditional Authorship Attribution Subtask. In: Forner P, Karlgren J, Womser-Hacker C, editors. CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy; 2012. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: http://www.uni-weimar.de/medien/webis/events/pan-12/pan12-papers-final/pan12-author-identification/juola12-overview.pdf.

[95] Juola P, Stamatatos E. Overview of the Author Identification Task at PAN 2013. In: Forner P, Navigli R, Tufis D, editors. CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain; 2013. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: http://www.uni-weimar.de/medien/webis/events/pan-13/pan13-papers-final/pan13-authorship-verification/juola13-overview.pdf.

[96] Stamatatos E, Daelemans W, Verhoeven B, Potthast M, Stein B, Juola P, et al. Overview of the Author Identification Task at PAN 2014. In: Cappellato L, Ferro N, Halvey M, Kraaij W, editors. CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers, 15-18 September, Sheffield, UK. CEUR-WS.org; 2014. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: http://www.uni-weimar.de/medien/webis/events/pan-14/pan14-papers-final/pan14-authorship-verification/stamatatos14-overview.pdf.

[97] Stamatatos E, amd Ben Verhoeven WD, Juola P, López-López A, Potthast M, Stein B. Overview of the Author Identification Task at PAN 2015. In: Cappellato L, Ferro N, Jones G, San Juan E, editors. CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse, France. CEUR-WS.org; 2015. Last accessed (DD/MM/YYYY) 20/09/2017. Available

from: http://www.uni-weimar.de/medien/webis/events/pan-15/pan15-papers-final/pan15-authorship-verification/stamatatos15-overview.pdf.

[98] Halvani O, Winter C, Graner L. On the Usefulness of Compression Models for Authorship Verification. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. ARES '17. New York, NY, USA: ACM; 2017. p. 54:1–54:10. Available from: http://doi.acm.org/10.1145/3098954.3104050.

[99] Halvani O, Winter C, Graner L. ARES_WSDF2017; 2017. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: https://www.dropbox.com/sh/f2mlp6u5vervx9b/AABr_c7qrmahCqUviIu3ORz6a?dl=0.

[100] schonlau. Masquerading User Data;. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: http://www.schonlau.net/intrusion.html.

[101] Wang k, Stolfo S. One-Class Training for Masquerade Detection. 2003 01;Last accessed (DD/MM/YYYY) 01/10/2017. Available from: https://www.researchgate.net/publication/247054265_One-Class_Training_for_Masquerade_Detection.

[102] netresec. Publicly available PCAP files; 2017. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://www.netresec.com/?page=PcapFiles.

[103] malware-traffic analysis. A source for pcap files and malware samples...; 2017. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://malware-traffic-analysis.net/.

[104] pcapr. Welcome to pcapr, where pcaps come alive.;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.pcapr.net/home.

[105] evilfingers. PCAP Repository; 2010. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://www.evilfingers.com/repository/index.php.

[106] caida. CAIDA Data - Overview of Datasets, Monitors, and Reports; 2017. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://www.caida.org/data/overview/.

[107] mining group C. Datasets;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://csmining.org/index.php/data.html.

[108] azsecure data. Get Data; •. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.azsecure-data.org/get-data.html.

[109] Corpora D. Corpora;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://digitalcorpora.org/corpora.

[110] Harrison S. The Global Inteligence Files;. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://wikileaks.org/the-gifiles.html.

[111] wlstorage net. Index of /torrent/gifiles/;. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://wlstorage.net/torrent/gifiles/.

[112] Kaggle. Hillary Clinton's Emails; 2016. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://www.kaggle.com/kaggle/hillary-clinton-emails.

[113] Tatman R. Fraudulent E-mail Corpus CLAIR collection of "Nigerian" fraud emails; 2017. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://www.kaggle.com/rtatman/fraudulent-email-corpus.

[114] Ruano-Ordas D. Corpus 200 Emails. 2015 3;Available from: https://figshare.com/articles/Corpus_200_Emails/1326662.

[115] Enrondata. Data;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from:

https://enrondata.readthedocs.io/en/latest/references/data/.

[116] Dang-Nguyen DT, Pasquini C, Conotter V, Boato G. RAISE: A Raw Images Dataset for Digital Image Forensics. In: Proceedings of the 6th ACM Multimedia Systems Conference. MMSys '15. New York, NY, USA: ACM; 2015. p. 219–224. Available from: http://doi.acm.org/10.1145/2713168.2713194.

[117] Dang-Nguyen DT, Pasquini C, Conotter V, Boato G. Introducing RAISE dataset;. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: http://mmlab.science.unitn.it/RAISE/.

[118] Mirsky Y, Shabtai A, Rokach L, Shapira B, Elovici Y. SherLock vs Moriarty: A Smartphone Dataset for Cybersecurity Research. In: Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. AISec '16. New York, NY, USA: ACM; 2016. p. 1–12. Available from: http://doi.acm.org/10.1145/2996758.2996764.

[119] Mirsky Y, Shabtai A, Rokach L, Shapira B, Elovici Y. DOWNLOADS; 2016. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: http://bigdata.ise.bgu.ac.il/sherlock/#/download.

[120] Allix K, Bissyandé TF, Klein J, Traon YL. AndroZoo: Collecting Millions of Android Apps for the Research Community. In: 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR); 2016. p. 468–471.

[121] androzoo. androzoo; 2016. Last accessed (DD/MM/YYYY) 24/09/2017. Available from: https://androzoo.uni.lu/.

[122] Pozzolo AD, Caelen O, Johnson RA, Bontempi G. Calibrating Probability with Undersampling for Unbalanced Classification. In: IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, December 7-10, 2015; 2015. p. 159–166. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: https://www3.nd.edu/~dial/publications/dalpozzolo2015calibrating.pdf.

[123] Andrea. Credit Card Fraud Detection: Anonymized credit card transactions labeled as fraudulent or genuine; 2016. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: https://www.kaggle.com/dalpozz/creditcardfraud.

[124] K R S, Zareapoor M. FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining. 2014 09;2014:252797. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: https://www.researchgate.net/publication/266746615_FraudMiner_A_Novel_Credit_Card_Fraud_Detection_Model_Based_on_Frequent_Itemset_Mining.

[125] purdue. Index of /data/credit_card;. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: https://www.cs.purdue.edu/commugrate/data/credit_card/.

[126] cms. Dataset Downloads; 2017. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: https://www.cms.gov/OpenPayments/Explore-the-Data/Dataset-Downloads.html.

[127] CMS. Open Payments Public Use Files: Methodology Overview & Data Dictionary; 2017. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: https://www.cms.gov/OpenPayments/Downloads/OpenPaymentsDataDictionary.pdf.

[128] Lopez-Rojas E. Synthetic Financial Datasets For Fraud Detection - Synthetic datasets generated by the PaySim mobile money simulator; 2017. Last accessed

(DD/MM/YYYY) 02/10/2017. Available from: https://www.kaggle.com/ntnu-testimon/paysim1.

[129] Lopez-Rojas EA. Applying Simulation to the Problem of Detecting Financial Fraud. , Department of Computer Science and Engineering; 2016.

[130] Lopez-Rojas EA, Axelsson S. BankSim: A Bank Payment Simulation for Fraud Detection Research; 2014. Available from: https://www.researchgate.net/publication/265736405_BankSim_A_Bank_Payment_Simulation_for_Fraud_Detection_Research.

[131] Lopez-Rojas EA, Axelsson S. Synthetic data from a financial payment system - Synthetic datasets generated by the BankSim payments simulator; 2017. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://www.kaggle.com/ntnu-testimon/banksim1.

[132] Amerini I, Ballan L, Caldelli R, Del Bimbo A, Serra G. A SIFT-Based Forensic Method for Copy&#x2013;Move Attack Detection and Transformation Recovery. Trans Info For Sec. 2011 Sep;6(3):1099–1110. Available from: http://dx.doi.org/10.1109/TIFS.2011.2129512.

[133] lambertoballan. sift-forensic; 2015. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://github.com/lambertoballan/sift-forensic/blob/master/README.md.

[134] Carrier B. Digital Forensics Tool Testing Images; 2010. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: http://dftt.sourceforge.net/.

[135] Corpora D. Real Data Corpus; 2017. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: https://digitalcorpora.org/corpora/disk-images/real-data-corpus.

[136] NIST. The CFReDS Project; 2016. Last accessed (DD/MM/YYYY) 20/09/2017. Available from: https://www.cfreds.nist.gov/.

[137] VirusShare. VirusShare.com - Because Sharing is Caring; 2017. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://virusshare.com/.

[138] Kaggle. Microsoft Malware Classification Challenge (BIG 2015);. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://www.kaggle.com/c/malware-classification/data.

[139] Arp D, Spreitzenbarth M, Gascon H, Rieck K. Drebin: Effective and explainable detection of android malware in your pocket; 2014. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://www.tu-braunschweig.de/Medien-DB/sec/pubs/2014-ndss.pdf.

[140] Arp D, Spreitzenbarth M, Gascon H, Rieck K. The Drebin Dataset; 2016. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://www.sec.cs.tu-bs.de/~danarp/drebin/.

[141] d Costa KAP, d Silva LA, Martins GB, Rosa GH, Pereira CR, Papa JP. Malware Detection in Android-Based Mobile Environments Using Optimum-Path Forest. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA); 2015. p. 754–759.

[142] RECOVI. DroidWare; 2015. Last accessed (DD/MM/YYYY) 23/09/2017. Available from: https://github.com/RECOVI/DroidWare.

[143] Bowen T, Poylisher A, Serban C, Chadha R, Chiang CYJ, Marvel LM. Enabling reproducible cyber research - four labeled datasets. In: MILCOM 2016 - 2016

IEEE Military Communications Conference; 2016. p. 539–544.

[144] McDaniel P. Data Sets;. Last accessed (DD/MM/YYYY) 23/09/2017. Available from: https://cybervan.appcomsci.com:9000/datasets.

[145] Kiss N, Lalande JF, Leslous M, Tong VVT. Kharon Dataset: Android Malware under a Microscope. In: The LASER Workshop: Learning from Authoritative Security Experiment Results (LASER 2016). San Jose, CA: USENIX Association; 2016. p. 1–12. Available from: https://www.usenix.org/conference/laser2016/program/presentation/kiss.

[146] Kharon-project. Kharon Malware Dataset; 2016. Last accessed (DD/MM/YYYY) 23/09/2017. Available from: http://kharon.gforge.inria.fr/dataset/.

[147] Avdiienko V, Kuznetsov K, Gorla A, Zeller A, Arzt S, Rasthofer S, et al. Mining Apps for Abnormal Usage of Sensitive Data. In: Proceedings of the 37th International Conference on Software Engineering. ICSE 2015; 2015. .

[148] Avdiienko V, Kuznetsov K, Gorla A, Zeller A. About MUDFLOW;. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: https://www.st.cs.uni-saarland.de/appmining/mudflow/.

[149] Lab IR. Datasets; 2010. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: https://www.uvic.ca/engineering/ece/isot/datasets/index.php#section0-0.

[150] ISOT. ISOT Dataset Overview;. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: https://www.uvic.ca/engineering/ece/isot/assets/docs/isot-datase.pdf.

[151] Saad S, Traoré I, Ghorbani AA, Sayed B, Zhao D, Lu W, et al. Detecting P2P botnets through network behavior analysis and machine learning. In: PST. IEEE; 2011. p. 174–180. Available from: http://ieeexplore.ieee.org/abstract/document/5971980/.

[152] lirmm. Analyzing Web Traffic ECML/PKDD 2007 Discovery Challenge; 2007. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: http://www.lirmm.fr/pkdd2007-challenge/index.html#dataset.

[153] isi csic es. HTTP DATASET CSIC 2010; 2012. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: http://www.isi.csic.es/dataset/.

[154] contagiodump. Collection of Pcap files from malware analysis; 2015. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://contagiodump.blogspot.no/2013/04/collection-of-pcap-files-from-malware.html.

[155] dropbox. PCAPS_TRAFFIC_PATTERNS fra DeepEnd Research (DeepEnd Research);. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://www.dropbox.com/sh/7fo4efxhpenexqp/AACmuri_l-LDiVDUDJ3hVLqPa?dl=0.

[156] Dolan-Gavitt B. (Sys)Call Me Maybe: Exploring Malware Syscalls with PANDA; 2015. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://moyix.blogspot.no/search?q=dataset.

[157] Garcia S. Dataset; 2015. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://stratosphereips.org/category/dataset.html.

[158] Samani EBB, Jazi HH, Stakhanova N, Ghorbani AA. Towards effective feature selection in machine learning-based botnet detection approaches. In: IEEE Conference on Communications and Network Security, CNS 2014, San Francisco, CA, USA, October 29-31, 2014; 2014. p. 247–255. Last accessed (DD/MM/YYYY)

02/10/2017. Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6997492.

[159] UNB. Botnet dataset;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.unb.ca/cic/research/datasets/botnet.html.

[160] Abdul Kadir AF, Stakhanova N, Ghorbani AA. In: Qiu M, Xu S, Yung M, Zhang H, editors. Android Botnets: What URLs are Telling Us. Cham: Springer International Publishing; 2015. p. 78–91. Available from: https://doi.org/10.1007/978-3-319-25645-0_6.

[161] UNB. Android Botnet dataset;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.unb.ca/cic/research/datasets/android-botnet.html.

[162] ll mit edu. DARPA Intrusion Detection Data Sets;. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://ll.mit.edu/ideval/data/index.html.

[163] Haines JW, Lippman RP, Fried DJ, Zissman MA, Tran E, Boswell SB. 1999 DARPA INTRUSION DETECTION EVALUATION DESIGN AND PROCEDURES; 2001. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://ll.mit.edu/ideval/files/TR-1062.pdf.

[164] ll mit edu. 2000 DARPA Intrusion Detection Scenario Specific Data Sets;. Last accessed (DD/MM/YYYY) 21/09/2017. Available from: https://ll.mit.edu/ideval/data/2000data.html.

[165] Romain F, Pierre B, Patrice A, Kensuke F. MAWILab Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking. In: ACM CoNEXT 10. Philadelphia PA;. .

[166] fukuda lab. MAWILab v1.1; 2017. Last accessed (DD/MM/YYYY) 23/09/2017. Available from: http://www.fukuda-lab.org/mawilab/data.html.

[167] "kdd ics uci edu". KDD Cup 1999 Data; 1999. Last accessed (DD/MM/YYYY) 23/09/2017. Available from: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[168] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A Detailed Analysis of the KDD CUP 99 Data Set. In: Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications. CISDA'09. Piscataway, NJ, USA: IEEE Press; 2009. p. 53–58. Available from: http://dl.acm.org/citation.cfm?id=1736481.1736489.

[169] Moustafa N, Slay J. The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems. In: 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS); 2015. p. 25–31.

[170] unsw-adfa-edu au". The UNSW-NB15 data set description; 2016. Last accessed (DD/MM/YYYY) 24/09/2017. Available from: https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/.

[171] states military academy west point U. Data Sets; 2009. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: Unitedstatesmilitaryacademywestpoint.

[172] Creech EITUCU Gideon. Developing a high-accuracy cross platform Host-Based Intrusion Detection System capable of reliably detecting zero-day attacks. Awar-

ded by:University of New South Wales. Engineering and Information Technology; 2014. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://unsworks.unsw.edu.au/fapi/datastream/unsworks:11913/SOURCE02?view=true.

[173] unsw. The ADFA Intrusion Detection Datasets; 2013. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-IDS-Datasets/.

[174] takakura. Traffic Data from Kyoto University's Honeypots; 2015. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: http://www.takakura.com/Kyoto_data/.

[175] SONG J, Takakura H, Okabe Y. Description of Kyoto University Benchmark Data;. Last accessed (DD/MM/YYYY) 01/10/2017. Available from: http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf.

[176] RAWDAD. All datasets and tools: sorted by name; 2017. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://crawdad.org/all-byname.html.

[177] automayt. A collection of ICS/SCADA PCAPs; 2016. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://github.com/automayt/ICS-pcap.

[178] amazon. Common Crawl on AWS;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: https://aws.amazon.com/public-datasets/common-crawl/.

[179] UNB. NSL-KDD dataset;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.unb.ca/cic/research/datasets/nsl.html.

[180] Lashkari AH, Gil GD, Mamun MSI, Ghorbani AA. Characterization of Tor Traffic using Time based Features. In: Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,. INSTICC. SciTePress; 2017. p. 253–262.

[181] UNB. Tor-nonTor dataset;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.unb.ca/cic/research/datasets/tor.html.

[182] Draper-Gil G, Lashkari AH, Mamun MSI, Ghorbani AA. Characterization of Encrypted and VPN Traffic using Time-related Features. In: Proceedings of the 2nd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,. INSTICC. SciTePress; 2016. p. 407–414.

[183] UNB. VPN-nonVPN dataset;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.unb.ca/cic/research/datasets/vpn.html.

[184] Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security. 2012;31(3):357 – 374. Available from: http://www.sciencedirect.com/science/article/pii/S0167404811001672.

[185] UNB. Intrusion detection evaluation dataset;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.unb.ca/cic/research/datasets/ids.html.

[186] Bonneau J. Yahoo Password Frequency Corpus. 2015 12;Available from: https://figshare.com/articles/Yahoo_Password_Frequency_Corpus/2057937.

[187] Blocki J, Datta A, Bonneau J. Differentially Private Password Frequency Lists. IACR Cryptology ePrint Archive. 2016;2016:153. Available from: http://eprint

.iacr.org/2016/153.

[188] Granville V. Password and hijacked email dataset for you to test your data science skills; 2012. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://www.datasciencecentral.com/forum/topics/password-dataset-for-you-to-test-your-data-science-skills.

[189] Wood T, Tarasuk-Levin G, Shenoy P, Desnoyers P, Cecchet E, Corner MD. Memory Buddies: Exploiting Page Sharing for Smart Colocation in Virtualized Data Centers. In: Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments. VEE '09. New York, NY, USA: ACM; 2009. p. 31–40. Available from: http://doi.acm.org/10.1145/1508293.1508299.

[190] umass. Index of /traces/cpumem/memtraces; 2009. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://skuld.cs.umass.edu/traces/cpumem/memtraces/.

[191] umass. readme.txt;. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://skuld.cs.umass.edu/traces/cpumem/memtraces/readme.txt.

[192] Chen T, Kan MY. Creating a live, public short message service corpus: the NUS SMS corpus. Language Resources and Evaluation. 2013 Jun;47(2):299–335. Available from: https://doi.org/10.1007/s10579-012-9197-9.

[193] kite1988. nus-sms-corpus; 2016. Last accessed (DD/MM/YYYY) 22/09/2017. Available from: https://github.com/kite1988/nus-sms-corpus.

[194] Wang D, Irani D, Pu C. Evolutionary Study of Web Spam: Webb Spam Corpus 2011 Versus Webb Spam Corpus 2006. In: Proceedings of the 2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2012). COLLABORATECOM '12. Washington, DC, USA: IEEE Computer Society; 2012. p. 40–49. Last accessed (DD/MM/YYYY) 22/09/2017. Available from: http://de-wang.org/download/webbspamcorpus2011.pdf.

[195] Wang D, Irani D, Pu C. Webb Spam Corpus 2011;. Last accessed (DD/MM/YYYY) 22/09/2017. Available from: https://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html.

[196] Delany SJ, Buckley M, Greene D. SMS spam filtering: Methods and data. Expert Systems with Applications. 2012;39(10):9899 – 9908. Available from: http://www.sciencedirect.com/science/article/pii/S0957417412002977.

[197] dublin institute of technology. DIT SMS Spam Dataset;. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://www.dit.ie/computing/research/resources/smsdata/.

[198] NIST. Spam Track; 2017. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://trec.nist.gov/data/spam.html.

[199] UCI. Spambase Data Set; 1999. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://archive.ics.uci.edu/ml/datasets/Spambase?ref=datanews.io.

[200] WEBSPAM-UK2007. "Web Spam Collections"; 2007. Crawled by the Laboratory of Web Algorithmics, University of Milan, http://law.di.unimi.it/.Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://chato.cl/webspam/datasets/uk2007/.

[201] Jun Liu(liukeen '@' mail xjtu cn) MZJMYL Hao Chen(lechenhao '@' gmail com).

microblogPCU Data Set;. MOEKLINNS Lab, Department of Computer Science ,Xi'an Jiaotong University, China. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: https://archive.ics.uci.edu/ml/datasets/microblogPCU.

[202] NIST. Tweets2011; 2014. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://trec.nist.gov/data/tweets/.

[203] Dredze M, Gevaryahu R, Elias-Bachrach A. Learning Fast Classifiers for Image Spam.; 2007. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.8417&rep=rep1&type=pdf.

[204] jhu. Image Spam Dataset; 2007. Last accessed (DD/MM/YYYY) 02/10/2017. Available from: http://www.cs.jhu.edu/~mdredze/datasets/image_spam/.

[205] PhishTank. FAQ;. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://www.phishtank.com/faq.php#whatisphishing.

[206] millersmiles. about us; 2017. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://www.millersmiles.co.uk/aboutus.php.

[207] Mohammad R, McCluskey TL, Thabtah FA. Intelligent Rule based Phishing Websites Classification. IET Information Security. 2014 May;8(3):153–160. Available from: http://eprints.hud.ac.uk/id/eprint/17994/.

[208] UCI. Phishing Websites Data Set;. Last accessed (DD/MM/YYYY) 26/09/2017. Available from: http://archive.ics.uci.edu/ml/datasets/Phishing+Websites#.

[209] forensicswiki. Tools; 2017. Last accessed (DD/MM/YYYY) 06/10/2017. Available from: http://www.forensicswiki.org/wiki/Tools#Open_Source_Tools.

[210] forensicswiki. Tools:Data Recovery; 2017. Last accessed (DD/MM/YYYY) 06/10/2017. Available from: http://www.forensicswiki.org/wiki/Tools:Data_Recovery.

[211] forensicswiki. Tools:File Analysis; 2015. Last accessed (DD/MM/YYYY) 06/10/2017. Available from: http://www.forensicswiki.org/wiki/Tools:File_Analysis.

[212] cugu. awesome-forensics; 2016. Last accessed (DD/MM/YYYY) 06/10/2017. Available from: https://github.com/cugu/awesome-forensics#live-forensics.

[213] rshipp. awesome-malware-analysis; 2017. Last accessed (DD/MM/YYYY) 06/10/2017. Available from: https://github.com/rshipp/awesome-malware-analysis#file-carving.

[214] wikipedia. List of digital forensics tools; 2017. Last accessed (DD/MM/YYYY) 06/10/2017. Available from: https://en.wikipedia.org/wiki/List_of_digital_forensics_tools.

[215] encasefinal. Digital Forensic Analysis, EnCase; 2011. Last accessed (DD/MM/YYYY) 06/10/2017. Available from: http://encasefinal.blogspot.no/2011/07/open-source-tools.html.

[216] Carrier B. Analysis Features;. Last accessed (DD/MM/YYYY) 08/10/2017. Available from: https://www.sleuthkit.org/autopsy/features.php.

[217] Carrier B. Keyword Search and Indexing;. Last accessed (DD/MM/YYYY) 08/10/2017. Available from: https://www.sleuthkit.org/autopsy/keyword.php.

[218] Carrier B. The sluth kit and open source digital Forensic conferance - Autopsy 3.0; 2012. Last accessed (DD/MM/YYYY) 08/10/2017. Available from: https://www.osdfcon.org/presentations/2012/OSDF-2012-Autopsy-3-0-Brian-Carrier.pdf.

[219] sleuthkit. About File Search; 2015. Last accessed (DD/MM/YYYY) 08/10/2017. Available from: http://sleuthkit.org/autopsy/docs/user-docs/3.1/file_search_page.html#how_to_open_file_search.

[220] sleuthkit. Keyword Search; 2015. Last accessed (DD/MM/YYYY) 08/10/2017. Available from: http://sleuthkit.org/autopsy/docs/user-docs/3.1/keyword_search.html#keyword_search_configuration_dialog.

[221] sleuthkit. Overview;. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://www.sleuthkit.org/autopsy/help/srch_mode.html.

[222] sleuthkit. Ad Hoc Keyword Search; 2017. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://sleuthkit.org/autopsy/docs/user-docs/4.3/ad_hoc_keyword_search_page.html.

[223] Carrier B. CONTENTS;. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://www.sleuthkit.org/informer/sleuthkit-informer-15.txt.

[224] sleuthkit. Reporting; 2015. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://sleuthkit.org/autopsy/docs/user-docs/3.1/reporting_page.html.

[225] sleuthkit. Autopsy User's Guide; 2015. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://sleuthkit.org/autopsy/docs/user-docs/3.1/index.html.

[226] sleuthkit. UI Layout; 2016. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: https://sleuthkit.org/autopsy/docs/user-docs/4.0/uilayout_page.html.

[227] sleuthkit/autopsy. Different character encodings #129; 2013. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: https://github.com/sleuthkit/autopsy/issues/129.

[228] sleuthkit. Hash Database Help;. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://www.sleuthkit.org/autopsy/help/hash_db.html.

[229] sleuthkit. Keyword Search Module; 2017. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://sleuthkit.org/autopsy/docs/user-docs/4.3/keyword_search_page.html.

[230] sleuthkit. Embedded File Extraction Module; 2015. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://sleuthkit.org/autopsy/docs/user-docs/3.1/embedded_file_extractor_page.html.

[231] sleuthkit. Interesting Files Module; 2015. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: http://sleuthkit.org/autopsy/docs/user-docs/3.1/interesting_page.html.

[232] wiki sleuthkit. PTK; 2013. Last accessed (DD/MM/YYYY) 09/10/2017. Available from: https://wiki.sleuthkit.org/index.php?title=PTK.

[233] fossies. contrib.plugins.psdispscan.PSDispScan Class Reference;. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://fossies.org/dox/volatility-2.6/classcontrib_1_1plugins_1_1psdispscan_1_1PSDispScan.html.

[234] volatilityfoundation. volatility/volatility/poolscan.py; 2015. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/volatilityfoundation/volatility/blob/master/volatility/poolscan.py.

[235] volatilityfoundation. volatility/volatility/scan.py; 2014. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/volatilityfoundation/volatility/blob/master/volatility/scan.py.

[236] volatilityfoundation. Command Reference Mal; 2017. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/volatilityfoundation/volatility/wiki/Command-Reference-Mal.

[237] volatility labs. Automating Detection of Known Malware through Memory Forensics; 2016. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://volatility-labs.blogspot.no/2016/08/automating-detection-of-known-malware.html.

[238] volatilityfoundation. volatilityfoundation/volatility; 2017. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/volatilityfoundation/volatility.

[239] volatilityfoundation. volatility/volatility/win32/tasks.py; 2016. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/volatilityfoundation/volatility/blob/69142099447a5248ebdb9e3ba636738d509b7055/volatility/win32/tasks.py.

[240] volatilityfoundation. Volatility; 2015. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://www.aldeid.com/wiki/Volatility#psscan.

[241] volatilityfoundation. volatility/volatility/plugins/malware/malfind.py; 2017. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/volatilityfoundation/volatility/blob/1ace3c4e0e1b85e97f5357a3b1f35b198868ac75/volatility/plugins/malware/malfind.py.

[242] JamesHabben. JamesHabben/evolve; 2015. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/JamesHabben/evolve.

[243] volatilityfoundation. Unified Output; 2016. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/volatilityfoundation/volatility/wiki/Unified-Output.

[244] mig. mig/doc/concepts.rst; 2017. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/mozilla/mig/blob/master/doc/concepts.rst.

[245] mig. Mozilla InvestiGator: File module; 2016. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/mozilla/mig/blob/master/modules/file/doc.rst#search-paths.

[246] MIG. mig/modules/memory/doc.rst; 2015. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/mozilla/mig/blob/master/modules/memory/doc.rst.

[247] MIG. mig/conf/mig-agent.cfg.inc; 2017. Last accessed (DD/MM/YYYY) 11/10/2017. Available from: https://github.com/mozilla/mig/blob/a2fe0fed53fb75d6c1ece5f917268c79774ca0ef/conf/mig-agent.cfg.inc.

[248] hachoir. Docs/hachoir-metadata program;. Last accessed (DD/MM/YYYY) 12/10/2017. Available from: http://hachoir3.readthedocs.io/metadata.html.

[249] hachoir. hachoir-subfile program;. Last accessed (DD/MM/YYYY) 12/10/2017. Available from: http://hachoir3.readthedocs.io/subfile.html.

[250] hachoir. Hachoir3 for developers;. Last accessed (DD/MM/YYYY) 12/10/2017. Available from: http://hachoir3.readthedocs.io/developer.html#why-using-hachoir-parsers.

[251] hachoir. hachoir.regex module;. Last accessed (DD/MM/YYYY) 12/10/2017. Available from: http://hachoir3.readthedocs.io/regex.html.

[252] elastic. Search;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-search.html#search-multi-index-type.

[253] elastic. Search APIs;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search.html.

[254] elastic. URI Search;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-uri-request.html.

[255] elastic. Sort;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-sort.html.

[256] elastic. Source filtering;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-source-filtering.html.

[257] Elastic. Script Fields;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-script-fields.html.

[258] Elastic. Post filter;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-post-filter.html.

[259] Elastic. Highlighting;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-highlighting.html.

[260] Elastic. Scroll;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-scroll.html.

[261] Elastic. Elasticsearch: The Definitive Guide [2.x] / Search in Depth / Controlling Relevance / Query-Time Boosting;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/guide/current/query-time-boosting.html.

[262] Elastic. Elasticsearch Reference [5.6] / Search APIs / Request Body Search / Index Boost;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-index-boost.html.

[263] Elastic. Elasticsearch: The Definitive Guide [2.x] / Getting Started / Sorting and Relevance / What Is Relevance?;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/guide/current/relevance-intro.html.

[264] Elastic. Elasticsearch Reference [5.6] / Search APIs / Request Body Search / min_score;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-min-score.html.

[265] Elastic. Elasticsearch Reference [5.6] / Search APIs / Request Body Search / Search After;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-search-after.html.

[266] Elastic. Elasticsearch Reference [5.6] / Search APIs / Request Body Search / Field Collapsing;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-collapse.html#CO27-2.

[267] Elastic. Elasticsearch Reference [5.6] / Search APIs / Search Shards API;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-shards.html.

[268] Elastic. Elasticsearch Reference [5.6] / Search APIs / Profile API;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-profile.html.

[269] Elastic. Elasticsearch Reference [5.6] / Aggregations;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations.html.

[270] Elastic. Elasticsearch Reference [5.6] / Indices APIs / Clear Cache;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-clearcache.html.

[271] Elastic. Elasticsearch Reference [5.6] / Query DSL;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html.

[272] Elastic. Elasticsearch Reference [5.6] / API Conventions / Common options;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/common-options.html#fuzziness.

[273] Elastic. Elasticsearch Reference [5.6] / Query DSL / Full text queries / Match Query;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query.html.

[274] Elastic. Elasticsearch Reference [5.6] / Query DSL / Full text queries / Simple Query String Query;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-simple-query-string-query.html.

[275] Elastic. Elasticsearch Reference [5.6] / Query DSL / Term level queries;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/term-level-queries.html.

[276] Elastic. Elasticsearch Reference [5.6] / Query DSL / Compound queries / Boosting Query;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-boosting-query.html.

[277] Elastic. Elasticsearch Reference [5.6] / Query DSL / Joining queries;. Last accessed (DD/MM/YYYY) 15/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/joining-queries.html.

[278] Grand A. Store compression in Lucene and Elasticsearch; 2015. Last accessed (DD/MM/YYYY) 16/10/2017. Available from: https://www.elastic.co/blog/store-compression-in-lucene-and-elasticsearch.

[279] elastic. Elasticsearch Reference [5.6] / Modules / Thread Pool;. Last accessed (DD/MM/YYYY) 16/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-threadpool.html.

[280] Mohan V. Eliminating Duplicate Documents in Elasticsearch; 2015. Last accessed (DD/MM/YYYY) 16/10/2017. Available from: https://qbox.io/blog/minimizing-document-duplication-in-elasticsearch.

[281] Elastic. Elasticsearch Reference [5.6] / Analysis / Token Filters / Stemmer Token Filter;. Last accessed (DD/MM/YYYY) 16/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-stemmer-tokenfilter.html.

[282] Elastic. Elasticsearch: The Definitive Guide [2.x] / Dealing with Human Language / Normalizing Tokens / In That Case;. Last accessed (DD/MM/YYYY) 16/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/guide/current/lowercase-token-filter.html.

[283] ElasticSearch. Elasticsearch Reference [5.6] » Mapping / Mapping parameters / boost;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-boost.html.

[284] Alger S. Elastic Team Member - Export data to csv file; 2017. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: https://discuss.elastic.co/t/export-data-to-csv-file/80250.

[285] Solr. Overview of Searching in Solr;. Last accessed (DD/MM/YYYY) 16/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/overview-of-searching-in-solr.html.

[286] Solr. Common Query Parameters;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/common-query-parameters.html#fq-filter-query-parameter.

[287] Solr. The Standard Query Parser;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/the-standard-query-parser.html.

[288] Solr. The DisMax Query Parser;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/the-dismax-query-parser.html.

[289] Solr. Highlighting;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/the-dismax-query-parser.html.

[290] Solr. Transforming Result Documents;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/transforming-result-documents.html.

[291] Solr. Exporting Result Sets;. Last accessed (DD/MM/YYYY) 20/10/2017. Avail-

able from: http://lucene.apache.org/solr/guide/7_0/exporting-result-sets.html.

[292] Solr. Suggester;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/suggester.html.

[293] Solr. MoreLikeThis;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/morelikethis.html.

[294] Solr. Result Clustering;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/result-clustering.html.

[295] Solr. De-Duplication;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: https://lucene.apache.org/solr/guide/6_6/de-duplication.html.

[296] Solr. Result Grouping;. Last accessed (DD/MM/YYYY) 20/10/2017. Available from: http://lucene.apache.org/solr/guide/7_0/result-grouping.html.

[297] wikiApache. General - What is Solr?; 2016. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: https://wiki.apache.org/solr/FAQ#Why_don.27t_International_Characters_Work.3F.

[298] WikiApache. Specifying a Query Parser; 2015. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: https://wiki.apache.org/solr/SolrQuerySyntax.

[299] WikiApache. Analyzers, Tokenizers, and Token Filters; 2016. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: https://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters#solr.EdgeNGramFilterFactory.

[300] WikiApache. Solr Relevancy FAQ; 2017. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: https://wiki.apache.org/solr/SolrRelevancyFAQ.

[301] Solr. Detecting Languages During Indexing;. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: https://lucene.apache.org/solr/guide/6_6/detecting-languages-during-indexing.html.

[302] sphinx. Boolean query syntax;. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: http://sphinxsearch.com/docs/latest/boolean-syntax.html.

[303] Sphinx. Quick summary of the ranking factors;. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: http://sphinxsearch.com/docs/latest/ranking-factors.html.

[304] Sphinx. Field-level ranking factors;. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: http://sphinxsearch.com/docs/latest/field-factors.html.

[305] Sphinx. Sorting modes;. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: http://sphinxsearch.com/docs/latest/sorting-modes.html.

[306] Sphinx. Grouping (clustering) search results;. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: http://sphinxsearch.com/docs/latest/clustering.html.

[307] Sphinx. Distributed searching;. Last accessed (DD/MM/YYYY) 21/10/2017. Available from: http://sphinxsearch.com/docs/latest/distributed.html.

[308] Sphinx. regexp_filter;. Last accessed (DD/MM/YYYY) 22/10/2017. Available from: http://sphinxsearch.com/docs/current/conf-regexp-filter.html.

[309] Sphinx. 12.2.7. dict;. Last accessed (DD/MM/YYYY) 22/10/2017. Available from:

http://sphinxsearch.com/docs/current.html#conf-enable-star.

[310] Sphinx. Sphinx features;. Last accessed (DD/MM/YYYY) 22/10/2017. Available from: http://sphinxsearch.com/docs/latest/features.html.

[311] Sphinx. BuildExcerpts;. Last accessed (DD/MM/YYYY) 22/10/2017. Available from: http://sphinxsearch.com/docs/latest/api-func-buildexcerpts.html.