Taxonomy of Data Fragment Classification Techniques

Rainer Poisel(✉), Marlies Rybnicek, and Simon Tjoa

St. Pölten University of Applied Sciences, St. Pölten, Austria
{rainer.poisel,marlies.rybnicek,simon.tjoa}@fhstp.ac.at
http://www.fhstp.ac.at

Abstract. Several fields of digital forensics (i.e. file carving, memory forensics, network forensics) require the reliable data type classification of digital fragments. Up to now, a multitude of research papers proposing new classification approaches have been published. Within this paper we comprehensively review existing classification approaches and classify them into categories. For each category, approaches are grouped based on shared commonalities. The major contribution of this paper is a novel taxonomy of existing data fragment classification approaches. We highlight progress made by previous work facilitating the identification of future research directions. Furthermore, the taxonomy can provide the foundation for future knowledge-based classification approaches.

Keywords: Digital forensics · Computer forensics · Data fragment · Classification · Taxonomy · File carving · Recovery · Collating

1 Introduction

The sources of digital fragments are manifold. Remnants of digital data can be found on all types of storage devices such as hard disks or USB sticks, in memory dumps, or in kind of packets in the case of computer networks [1]. Digital forensics deals with making sense of unstructured data in order to obtain evidence that can be used in court. Typical fields of application for data fragment classification are therefore general file recovery applications such as file carving, the analysis of memory or network dumps (e.g. detection of malware [2]).

The vast amount of data and the proliferation of file formats pose one of the major challenges which have to be solved by current and future developments in the field of digital forensics [3]. Garfinkel [3] concludes that these challenges will remain for the next 10 years. In order to overcome these issues, several strategies have been developed. In their work, Roussev et al. [4] propose to conduct partial analysis, so called "triage", in order to identify material that is relevant to an examination as quickly as possible. In that case, the analysis process takes place outside the actual forensics lab. Young et al. [5] follow a different approach to achieve the same objective. By using "sector hashing", investigators can automatically provide evidence about the existence of remnants from well-known

P. Gladyshev et al. (Eds.): ICDF2C 2013, LNICST 132, pp. 67–85, 2014.
DOI: 10.1007/978-3-319-14289-0_6

files on various types of digital storage media. In contrast to identifying fragments from known files, the topic of file fragment classification deals with the identification of the file type of known and unknown data fragments.

Up to now many research papers have been published in this field. They all aim at improving the accuracy and/or the performance of the data fragment classification process. Especially determining the file type of complex container formats such as multimedia file formats (e.g. AVI or MP4) or the Adobe PDF file format has proven difficult. Recent analysis has shown that it might be necessary to combine different approaches (e.g. determining the information entropy [6], the existence of signatures, and the fragmentation behavior) to achieve the goal of correct file type classification of digital fragments [7]. In course of this research paper we summarize our findings in the field of fragment type classification in a taxonomy.

The **main contribution** of this paper is the introduction of a novel taxonomy for approaches which can be used to classify data fragments. For each category we surveyed existing literature to clarify the structure and to facilitate the usage of the taxonomy.

In course of this paper we start with related work in Sect. 2. Section 3 describes existing approaches and classifies them into our taxonomy. Furthermore, we give a visual representation of our taxonomy. In Sect. 4 we summarize our findings and give an outlook for future developments in this field.

2 Prior and Related Work

In her publication, Beebe [8] points out that future research should address the volume and scalability issues digital forensics analysts see themselves confronted with. Only subsets of data should be selected strategically for image and further processing. This goal could be achieved by applying “Intelligent Analytical Approaches” which, exemplarily classify data feature-based without analyzing file signatures or file meta data. Furthermore, Beebe [8] mentions to apply artificial intelligence techniques to different applications (e.g. email attribution, data classification) in the field of digital forensics.

As shown later in this paper, numerous approaches that categorize input data of digital fragments into selected data type categories have been proposed. Most publications mention available solutions to perform this task in their related work section [7,9–12]. Other research papers elucidate different available data fragment type classification approaches in case, techniques are applied in order to achieve the actual research goal, e.g. recovering files from their fragments by applying the file carving approach [13,14].

Garfinkel [3] argues that some work has been conducted by the digital forensics community in order to create common schemas, file formats, and ontologies. However, to the best of our knowledge, no current research publication categorizes available fragment type classification solutions in kind of an extensible taxonomy. Several taxonomies have been published in recent digital forensics research publications. Raghavan [15] presented the current state of the art in digital forensics research in a taxonomy. The ultimate goal of his taxonomy was to summarize research directions for the future. In order to create the taxonomy, Raghavan reviewed research literature since the year 2000 and categorized developments since then into four major categories: digital forensics modelling, acquisition and modelling, examination and discovery, and digital forensics analysis. According to his findings, developments conducted in the field of fragment type classification can be found in the "examination and discovery" category. In their paper, Garfinkel et al. [16] present a taxonomy describing different types of corpora available to the digital forensics research community. By using the taxonomy the usage of available test data could e.g. be restricted to specific purposes or tailored to involved people.

3 Taxonomy of Classification Approaches

In this section, we introduce our taxonomy on data fragment classifiers. Our taxonomy aims at supporting experts from academia and industry by creating a common understanding on fragment classification within the discipline of data carving. For the development of our taxonomy we surveyed more than 100 research papers reflecting the state-of-the-art in this research area. In the following, we briefly outline the structure of our taxonomy before we present representatives of the individual classes.

In the course of our studies we identified several categories of data fragment classifiers. For our taxonomy we divided them into the following main-classes: signature-based approaches, statistical approaches, computational intelligence based approaches, approaches considering the context, and other approaches. Figure 1 schematically outlines our proposed taxonomy. The succeeding paragraphs contain more information on the structure and content of the above-mentioned main-classes.

Signature-based approaches use byte-sequences for the identification of unknown file fragments by matching typical and well known byte sequences. A wide-spread application area in the context of digital forensics is to determine header and footer fragments by file signatures (e.g. File Signature Table [17]) which are often referred to as magic number. Another common field of application, where signatures are extensively used, is the identification of known files by hash values. Inspired by file hashes, recent approaches in research (referred to as "sector hashing") identify known data fragments by their hash-value. Because of the characteristics of hash functions, these approaches cannot interpret analyzed data and therefore are only valid for a fixed block size. To overcome this weakness, "similarity hashing" approaches can be applied. More details on signature-based approaches are presented in Sect. 3.1.

Statistical approaches use quantitative analysis techniques to identify fragments of given file types. Statistical properties such as the mean value, variance, binary frequency distribution (BFD), or the rate of change (ROC) are determined from fragments contained in reference data sets to obtain a model for each data type. The actual classification is then carried out by comparing (e.g. by calculating the Mahalanobis distance [18]) the fragments in question to the precalculated model.

The goal of *computational intelligence approaches* is to transform data into information after learning from a collection of given data. For data fragment and type classification, strong classifiers have to be trained. We further refine this class into supervised (if the training set consists of labeled data) and unsupervised (if patterns and structures are derived from unlabeled data) approaches. Both supervised and unsupervised machine learning algorithms are used to meet the goal of correct classification of data fragments and file type classification.

Context-considering approaches use information gained from meta-data extracted from other fragments or the transport medium. Such approaches can provide additional information necessary for the correct classification.

The category *other approaches* contains techniques which cannot be assigned to one of the other categories. A special sub-class of class are *combining approaches.*

Based on the characterization of classification approaches, the following subsections describe available data fragment classification approaches. For each classification approach we describe its properties such as availability, fields of use, its strengths and its weaknesses.

3.1 Signature-Based Approaches

Signature-based approaches are suitable to identify data fragments or their types by matching predefined byte-sequences and are applied widespreadly by different programs such as the “file” command [19] or by signature-based file carvers such as “scalpel” [20], “foremost” [21], “ReviveIT” [22], or “Photorec” [23] to determine the file type of files or file fragments e.g. by matching header/footer byte-sequences. Pal et al. [13, 24] referred to this approach as “syntactical tests” because file types are identified by searching for signatures that are typical for specific file types (e.g. HTML tags in the case of HTML files). This approach has also been implemented for binary files by Al-Dahir et al. [25]. In order to determine whether subsequent blocks belong to a MP3 file, their contents have been searched for signatures of MP3 frame headers. Garfinkel et al. [26] extended the before mentioned approach by searching for additional signatures to be found in MP3 data fragments.

For the identification of known file content (and thus their type) and to reduce the amount of files that have to be processed in case of a digital forensics investigation, a library of hash-values from well-known files (e.g. files belonging to an operating system or device drivers) has been made publicly available on the Internet [27] by the National Institute of Standards and Technology (NIST). Using the National Software Reference Library (NSRL), well-known files can be excluded from the analysis process. Furthermore, using the NSRL investigators may find out which software applications are present on analyzed systems. In their work, Mead [28] examined whether file signatures contained in the NSRL produce unique results. The uniqueness of file identification has been analyzed both empirically as well as by conducting research in the field

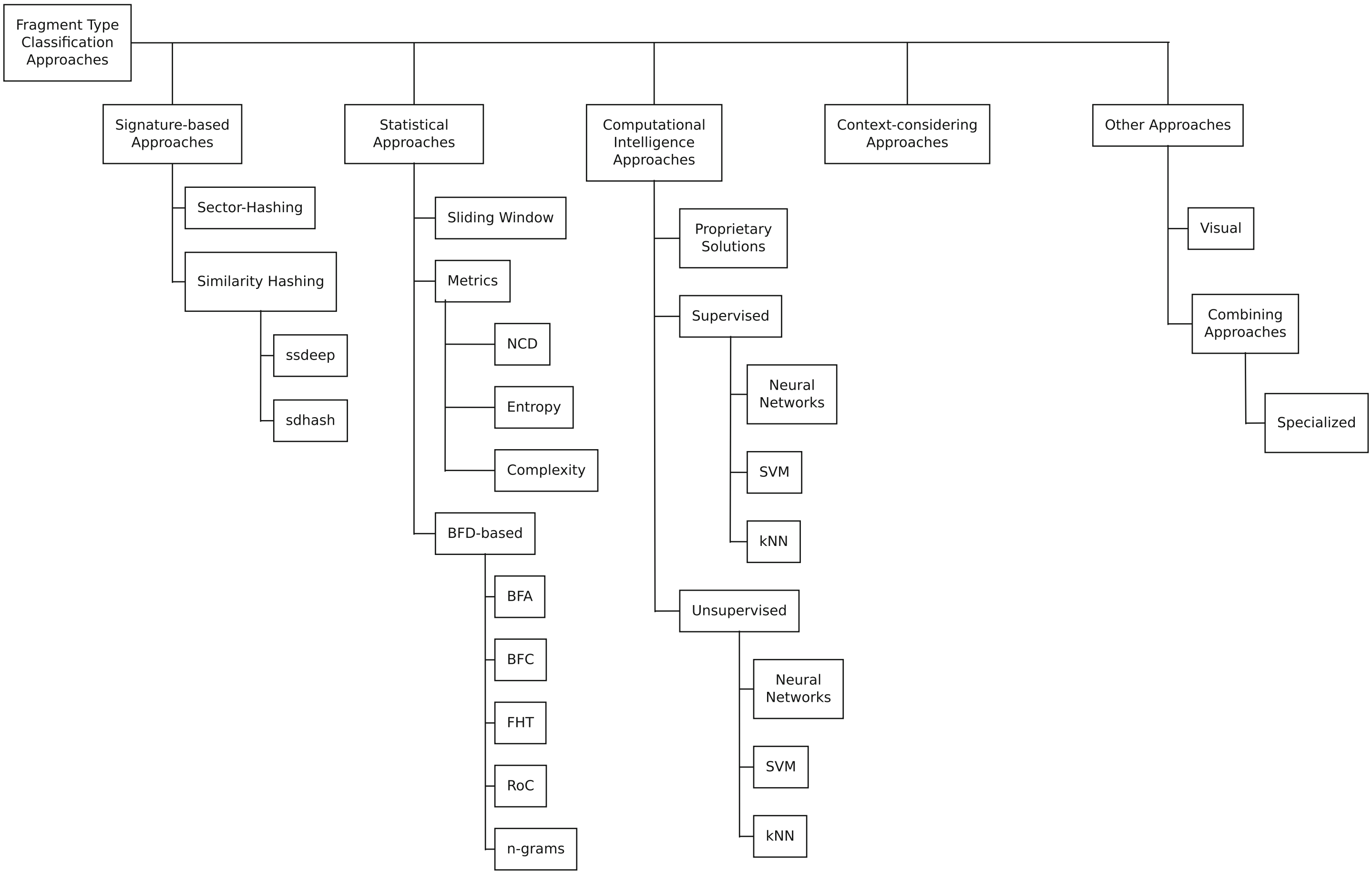

Fig. 1. Visual representation of our data fragment classification taxonomy

of attacks on hash algorithms used to generate the file signatures contained in the library. Garfinkel [29] mentions that the NSRL is part of his digital evidence research corpus. The NSRL Reference Data Set (RDS) has been extended by Kim et al. [30] to support for the exclusion of Korean software from the analysis process. Ruback [31] took this approach further by implementing data mining techniques in order to create hashsets that only contain samples from a given country or geographical region. Chawathe [32] proposes an improvement that allows for deciding on which files to consider for further analysis in the digital forensics process. His approach is based on hashing composite file signatures using a locality-sensitive hashing scheme. The more similar items are, the more it is likely that they are put into the same hash bucket [33].

A similar approach has been followed by Garfinkel [34] and Dandass et al. [35]. Instead of matching parts of data fragments hash-values of whole fragments (e.g. the sector of a storage medium) are calculated. The hash-value of data fragments' content is unique with high-probability. In case a fragment's hash-value matches a pre-calculated hash-value, its content is known and thus its file type. In [35] they conclude that both, the CRC32 and the CRC64 algorithms, produce shorter hash-values than the MD5 and the SHA1 algorithm while having a comparably low false positive rate. Collange et al. [36] proposed to speed up hash computations by using Graphical Processing Units (GPUs). Furthermore, Collange et al. [36] introduced the term "Hash-based Data Carving" for matching hash-based signatures of disk sectors with signatures of sectors from known contraband files.

Identifying data fragments by their hash-value has become a well-established technique in digital forensics research. It is referred to as "sector hashing" [5,37]. The main challenge of "sector hashing" is to store the vast amount of hashes in a suitable database. In their work, Garfinkel et al. [26] describe data structures (map-based) and an algorithm that minimize the amount of storage required to match contents of known files (master files) with files contained on an arbitrary image (image files). Experts [5,26,38] propose the usage of bloom filters before storing the hash-values in a database (a B-tree back end).

Hash algorithms are one-way functions and they work at the byte-stream level. As these functions do not attempt to interpret analyzed data, commonality can only be proven of the binary representations of digital fragments [39]. Hash values of the "sector hashing" approach are only valid for a given and fixed block size. In order to overcome this issue, Roussev [40] proposed to generate similarity fingerprints (similarity preserving hashing, SPH) of data fragments which are based on statistical improbable features. A generic, entropy-based scheme allows for the selection of features independent of the file type of data fragments. Using this approach, similarity digests finally consist of a sequence of bloom filters and their length is about 2–3 % of the input-length [39]. The "sdhash" approach proposed by Roussev [40] outperforms Kornblum's [41] "ssdeep" both in precision (94 % vs 68 %) and recall (95 % vs 55 %) [39]. Roussev and Quates [42] demonstrate the applicability of the "sdhash" approach to a large case (e.g. 1.5 TB of raw data). The purpose of using "sdhash" was to narrow down the amount of data that had to be processed from different types of media (disk images, RAM snapshots, network traces). Based on their findings on triage, the requirements for conducting real-time digital forensics and triage are discussed by Roussev et al. [4]. In order to process at a rate of approximately 120 MB/s Roussev et al. [4] state that about 120–200 computing cores are necessary. Breitinger et al. [43–47] propose advancements in the field of similarity preserving hashing (SPH). Bloom filters are used to represent fingerprints and the idea of majority voting as well as run length coding for compressing input data are applied to achieve improvements over existing approaches. As a result their new algorithm "mvHash-B" shows faster processing (20 times faster than "sdhash"), reduced hash value length (0.5 % of input length), and improvements regarding the rebustness against active manipulation of hashed contents.

3.2 Statistical Approaches

These approaches evaluate different characteristics, such as their information entropy or the binary frequency distribution (BFD) of data fragments, in order to determine their file type.

McDaniel and Heydari [48,49] introduced the concept of "fingerprints" for the file type identification of data fragments. These fingerprints contain characteristic features that are specific to each different file type. In their work [49] they propose three different algorithms: the Byte Frequency Analysis (BFA) algorithm, the Byte Frequency Cross-Correlation (BFC) algorithm and the File Header/Trailer (FHT) algorithm. The BFA algorithm determines the number of occurrences of possible byte values (0–255 inclusive) in a given sample which is referred to as the Binary Frequency Distribution (BFD). Different file formats show different characteristic patterns of BFDs which can be used to distinguish them from other file formats. For the given set of samples [49] the accuracy (true positive rate or TP-rate) of the BFA approach turned out to be 27.50 %. The BFC algorithm extends the BFA algorithm by considering differences of byte frequencies of different byte values. That way, the accuracy could be improved to 45.83 %. The FHT algorithm additionally considers signatures found in the beginning and in the end of analyzed files. That way, the accuracy could be improved to over 95 %. However, Roussev and Garfinkel [7] argue, that this approach is unsuitable for file type identification of data fragments, as usually no such signatures exist in digital artifacts. In their work, Dhanalakshmi and Chellappan [50] give an overview of various statistical measures that could be used for data fragment classification. However, the authors neither give detailed information on how to interpret these values nor do they provide the achieved classification accuracy.

Based on [48,49], Li et al. [18] propose the usage of so called "fileprints" which are based on n-grams which in turn have been derived from the BFD of block contents. Full 1-gram distributions (so called "fileprints") consist of two 256-element vectors at most: these two vectors represent the average byte frequency and their variance. The approach proposed by Li et al. [18] achieved remarkable results (nearly 100 % success rate for 20 byte fragments) for the classification of whole files. It was not intended to be used for fragments that originated from the middle of files and thus does not evaluate the classification rate of such fragments. In his thesis, Karresand [51] propose to calculate the centroid of contained 2-grams to identify the type of data fragments with high information entropy. In order to achieve good results, the centroid values are calculated for data blocks of 1 MiB and then scaled down to match fragment sizes of 4 KiB. The results for these approaches are presented in kind of "Receiver Operating Characteristic" (ROC) and in a confusion matrix. In the ROC charts, the true positive rates are lotted against the false positive rates (FP-rates) while varying the detection threshold. Different detection thresholds have been achieved by varying the centroid used. According to the confusion matrix for the 2-gram algorithm the detection rate for detecting fragments from JPEG files with no restart markers (RST) was close to perfection (99.94 % TP, 0.0573 % FP). Fragments from JPEG files containing restart markers were classified as fragments containing no restart markers with a probability of 42.668 %. In their later paper, Karresand and Shahmehri [52] build upon findings of their earlier work to reassemble fragments from JPEG files. Mayer [53] proposes approaches dealing with n-grams longer than 2 bytes. The algorithm is based on an approach to extract and summarize features as proposed by Collins [54]. In order to overcome the vast performance requirements of traditional n-gram based approaches, similar common n-grams have been collapsed into summarized n-grams where a summarized n-gram represents common features of a file type. Using this approach an overall accuracy of 66 % could be achieved for 14 file types out of 25. In order to detect executable code in network packets, Ahmed and Lhee [2] use the n-gram approach. They [2] conclude, that the order of used n-grams influences the achievable accuracy of the classification process. In their tests, 3-grams were accurate enough to identify executable contents (FP-rate: 4.69 %, false-negative or FN-rate: 2.53 %). Cao et al. [10] analyzed the influence of the number of grams evaluated on the classification accuracy. Experiments showed that best results could be achieved when selecting 300 grams per fragment classified.

Karresand and Shahmehri [55,56] propose the "Oscar" method for file fragment classification. In contrast to [18], their approach additionally considered the Rate of Change (RoC). Karresand and Shahmehri's approach [56] is therefore well suited for the identification of JPEG fragments because they contain a large number of 0xFF 0x00 byte-pairs which have the highest RoC of almost any file type. For other file types, the approach is less suitable, due to the high false positive rate (e.g. 70 % for Windows executables).

Hall and Davis [57] proposed an algorithm based on a sliding window. The entropy and compressibility measurements were averaged and standard deviation values were calculated of each sliding window from reference files of the file types in question resulting in a profile plot. During the actual file type classification, each point of files in question sliding window values were subtracted from available profile plots ("goodness of fit"). The profile plot with the smallest difference determined the according file type. Besides subtracting sliding window values from profile plots, Hall and Davis also applied Pearson's Rank Order

Correlation in order to determine how well two data sets correlate. Results of their approaches ranged between 0 to 100 % accuracy with 20 out of 25 results having an accuracy greater than 80 % for the first approach. The second approach achieved accuracy values between 12 to 100 % with the most values having an accuracy between 50 to 75 %. The test data set consisted of 73,000 files from 454 different file types.

Approaches using several different statistical measurements are categorized as "metrics based" approaches by Roussev and Garfinkel [7]. Erbacher and Mulholland [58] differentiate between file formats by observing distributions, averages, and statistical measurements of higher momentum. In their later paper, Moody and Erbacher [59] propose a classification system that is based on their previous work [58]. It is suitable for determining the overall data type (text-based data, executable data, compressed data). Their so called "SÁDI" approach, as they call it, produces mixed results in a secondary analysis to distinguish between sub-classes of the overall data types (e.g. text, csv, or html in case of text-based data fragments).

Veenman [60] proposed to evaluate three different cluster content features in order to determine the file type of data fragments: the histogram of byte values (BFD), the information entropy of the content, and the algorithmic (or Kolmogorov) complexity. Veenman worked on a dataset of notable size (training set of 35.000 clusters and test set of 70.000 clusters) with a block size (4096 bytes) typically found when analyzing established file systems such as NTFS. Results are presented in kind of a confusion matrix. Best results could be achieved for fragments from JPEG or HTML files (true positive rate higher than 97 % for both). However, the classification results for most file types with higher information entropy were rather modest (e.g. 18 % true positive rate for ZIP files or 35 % for GIF files).

Calhoun and Coles [61] focused on the classification of data fragments without the presence of meta data. In their work, they extended the work of Veenman [60] by considering additional statistical measurements (i.e. Fisher linear discriminant, longest common subsequence). Roussev and Garfinkel [7] emphasize that Calhoun [61], with exception of the sample size, provide one of the first realistic evaluations. Results ranged between a 50 % and 95 % true positive rate for distinguishing JPEG fragments from PDF fragments (512 byte size) and between 54 % and 85 % for distinguishing JPEG fragments from GIF fragments.

Axelsson proposed to calculate the Normalized Compression Distance (NCD) [62,63] between reference blocks (blocks of known file type) and blocks in question. As a classification algorithm Axelsson implemented the k-nearest-neighbor algorithm. The class (file type) of an analyzed fragment is thus assigned based on a majority vote of the k-nearest feature vectors.

Depending on the k-value, the hit rate averaged from 36.43 % for the first nearest neighbor to 32.86 % for the 10-nearest neighbors. The hit rate of the k-nearest neighbor approach was compared to the hit rate of perfect randomness which calculates to approximately 3.5 % (= 1/28) as the reference corpus consisted of data fragments from 28 different file types. In their work, Poisel et al. [14] used the NCD approach for distinguishing data fragments from files with information entropy from those with low information entropy.

Savoldi et al. [64] presented an approach to discriminate between wiped and standard data fragments. Their findings are useful to detect the usage of anti-forensics measures [65] which do not leave traces on the system level (e.g. entries in the Registry, file systems, metadata, etc.). In order to determine whether byte sequences from data fragments have been produced by either hard- or software cryptographic random or pseudorandom number generators, Savoldi et al. drew on a test suite developed by the NIST [66]. In a case study with a chunk size of 4 KiB and a corpus of 104192 fragments, 97.6 % could be classified correctly in 142 min. With a chunk size of 4 MiB, the same amount of data could be processed in 15 min with an accuracy of 94.14 %.

3.3 Artificial Intelligence Approaches

Computational intelligence approaches explicitly operate in two phases: training and test. This group of file type identifiers utilize different types of classifiers from the field of computational intelligence [67] such as k-nearest neighbor (kNN), k-means, Support Vector Machines (SVM), Neural Networks (NN), or Bayesian Networks (BN).

TrID is a closed-source tool that identifies the type of files or data fragments based on their content [68]. Training of this product is accomplished by the "TrIDScan" tool [68] which creates databases that contain relevant file type definitions. However, as the TrID developer does not mention the accuracy (e.g. true-/false-positive rate, confusion matrix) or the methodology in detail it cannot be considered a forensic tool [59].

Ahmed et al. [69, 70] analyzed the accuracy of popular classifiers (e.g. neural networks, kNN, SVMs, etc.) with high-frequency byte patterns (1-gram features). Byte patterns (features) were extracted from different samples of the same file type. Empirical tests showed, that using the union operation for extracted features of files from the same file type performed best when used together with the kNN classifier. In their tests, Ahmed et al. [70] identified the file type of segments of different size. With a test set consisting of 5,000 files from 10 different file types, the achieved classification accuracy for 4 kB fragments was 0 %, less than 20 %, and 85 % for JPEG, MP3, and EXE fragments respectively. In their later paper, Ahmed et al. [71] focus on the reduction of computational time necessary to identify the file type of digital fragments. They conclude that by randomly sampling file blocks and by only using a subset of features the computational time necessary could be reduced fifteen-fold.

Li et al. [72] elaborate on file type identification using SVMs. Selected features are similar to those selected by Ahmed et al. [70]. Instead of using only high frequency byte-patterns, Li et al. [72] used all 256 possible 1-grams as input to the SVM classifier. The accuracy achieved by this approach was highest (>89 %) when using the linear kernel for distinguishing between fragments from test sets consisting of only two different file types (JPEG from DLL, PDF, or MP3 fragments). Li et al. [72] do not address the problem of identifying fragments from one file type in a test set consisting of fragments from an arbitrary (or high) number of different file types as given in typical file type identification scenarios (e.g. file carving). Similar to Li et al. [72], Gopal et al. [73,74] utilize SVMs-classifiers (additionally to kNN-classifiers) for identifying the file type of data fragments. As features they selected 1- and 2-grams of classified fragments. They compare the accuracy of their classifiers to commercial of the shelf (COTS) applications such as libmagic [19] and TrID [68]. When applied on whole files, which had the first 512 bytes removed, both the kNN (1-gram) and the SVM (2-gram) classifiers (TP-rate >85 %) outperform COTS applications by a factor greater than 7.

In order to improve the classification accuracy of the SVM approach in dependence on searched file types, in their work, Sportiello and Zanero [75] focus on the feature selection part. Besides the BFD they identified comprehensive list of other features suitable for determining the file type of data fragments: BFD of all possible values or only part of it (e.g. ASCII range), the RoC, entropy, complexity, mean byte value, etc. Best reported results presented in a confusion matrix vary between a TP-rate of 71.1 % with a FP-rate of 21.9 % for doc files (using Entropy, and BFD and/or Complexity as features) and a 98.1 % TP-rate with a FP-rate of 3.6 % for bmp files (using the RoC as feature). Fitzgerald et al. [11] extended selected features by using additional features from the field of natural language processing (NLP): they computed the average contiguity between bytes of sequences and the longest contiguous streak of repeating bytes. Experiments conducted with a test set consisting of 512-byte long file fragments from 24 different file types showed, that a prediction accuracy of more than 40 % could be achieved (in contrast to random chance of $\frac{1}{24} \approx 4.17\,\%$). It is remarkable that file types from the same file type family (csv, html, java, txt, ... are all text-based) could be predicted with varying accuracy, thus making the approach especially suitable for specific file types (e.g. csv, ps, gif, sql, html, java).

Beebe et al. [76] identify several other features such as the Hamming weight, standardized kurtosis, standardized skewness, average contiguity, maximum byte streak, etc. which can be used as features by utilized SVMs. According to their evaluation, their DFRWS 2012 Forensics challenge winning open-source prototype "Sceadan" [77] achieved high classification accuracy (>80 %) for text-based and multimedia file formats (e.g. MP3, M4A, JPG, MP4).

As single file fragments may contain multiple types, Garfinkel et al. [26] refer to the process of determining the file type of data fragments as "fragment discrimination" rather than "fragment type identification". Container files may consist of unaltered files from other file types (e.g. JPEG files which are embedded in PDF files). However, Sportiello and Zanero [78] did not adopt this term. In their work they propose a "context-based classification architecture" [78] which takes into account that file blocks belonging to the same file are typically stored contiguously on various types of storage media. Sportiello and Zanero [78] improve the training set of their SVM approach. The training set for primitive file types (e.g. jpg, gif, bmp, mp3) was unchanged to their former paper [75], but the training sets of compound file types (doc, odt, exe, and pdf) consisted only of empty files, not containing any embedded files. Furthermore, they consider the context around currently classified fragments. By conducting experiments with varying context-sizes and by applying different context evaluation functions, they proved that the FP- and FN-rates were approximately 3–5 % lower than without considering the context. However, Sportiello and Zanero [78] only considered the file type of neighbor fragments rather than considering the actual fragmentation-behavior of involved storage-mechanisms (e.g. file-system characteristics) which could have improved the accuracy further.

In contrast to previous approaches presented in this chapter, Amirani et al. [79] followed the concept of unsupervised learning algorithms. With their approach, during the training phase, the Principal Component Analysis (PCA) projection matrix is calculated from the BFDs of the training set. After that an auto-associative neural network is trained with the output features obtained from the PCA using a back-propagation algorithm so that outputs are the same as inputs. Even though any classifier could have been used in the testing phase, the authors decided to use three layer Multi Layer Perceptron (MLP) [79] and SVM classifiers [80]. Experiments conducted with a small test set consisting of 200 files of each tested file type (doc, exe, gif, htm, jpg, pdf) showed that the approach gives promising results. The average total accuracy for 100 examined data fragments per file type calculated to 99.16 %, 85.5 %, and 82 % (TP-rate; running times all (<0.05 s) when applied to whole file contents, random file fragments with a size of 1500 bytes, and random file fragments with a size of 1000 bytes. However, Amirani et al. [79,80] do not mention details about the test set, which would have been of special interest for the chosen compound file formats (doc, exe, pdf).

Carter [12] identifies and locates fragments containing executable code using a KNN classifier which analyzes n-gram and semantics-based features of to-be classified fragments. Semantic features are derived from byte-streams by using a disassembler and n-grams are used in case the disassembler cannot extract semantic information from raw byte-streams. Carter's approach [12] is unsupervised as necessary features as well as the labels of the training material are determined in the first phase of the algorithm. Results showed, that the approach is resilient to certain obfuscation methods: in conducted experiments, 89 % of classified fragments could be associated with their original source.

Kattan et al. [81] propose a system that is based on Genetic Programming (GP). Intended fields of application for their classifier are spam filters and anti-virus software. It performs three major functions: segmentation, creation of fileprints, classification. Byte-series are segmented based on statistical features so that each segment consists of statistically uniform data. Fileprints are used to identify unique signatures that are characteristic for each file type. As fileprints consist of vectors that contain a series of abstracted numbers which describe the contents of files, they are referred to as GP-fingerprints. GP is then used to cluster the file fingerprints in a two-dimensional Euclidean space so that all fingerprints in the same cluster belong to the same file type. Classification of unseen data occurs then by applying the k-nearest neighbor approach. As a disadvantage the authors [81] mention, that their approach entails a long-lasting learning process and as an advantage they point out the high accuracy of their classifier (88.90 % for three file types, 70.77 % for five file types). However, training and test sets chosen by the authors are considerably small (less than 20 MiB per file type). Therefore it is hard to estimate the overall performance achievable with their approach.

3.4 Approaches Considering the Context

Approaches considering the context involve information from other fragments or information related to the transport medium (e.g. the fragmentation behavior in case of storage media). Sportiello and Zanero [78] extended the approach of using SVM classifiers with features extracted from the binary representation of data fragments by considering the file type of surrounding fragments when identifying the file type of data fragments.

Garcia and Holleboom [82,83] elaborated on an analytical model that describes the probability for the existence of micro-fragments in slack space in case the original containing blocks have been overwritten. Blacher [84] applied the generic analytical model of Garcia and Holleboom [82,83] to the NTFS. While the work of previously mentioned authors does not improve the accuracy or performance of data fragment classifiers directly, their analytical model could be of use for future applications such as sector hashing [5] or for determining filesystem related parameters in the recovery of classification [78] process. Xu et al. [85] proposed "... an adaptive method to identify the disk cluster size based on the content of sectors" [85]. Xu et al. [85] differentiate non-cluster boundaries from cluster boundaries by comparing entropy difference distributions. Unavailable or corrupted filesystem meta data complicates the recovery process of fragmented files [86]. As bigger fragments results in a better classification accuracy, this approach could be especially useful in the situation of missing filesystem meta data. Furthermore, this approach would support classification algorithms to consider correct block boundaries, thus making it easier to localize remaining signatures in analyzed data fragments.

3.5 Other Approaches

Conti et al. [87] proposed to visually represent binary structures in order to determine their file type. As part of their publication they developed a visual taxonomy that can be used by applying image processing operations. Conti et al. [88] applied their visualization approach to large binary fragments in order to speed up analysis activities of storage devices.

As there are many file formats with similar characteristics Roussev and Garfinkel [7] proposed to combine approaches from the fields of signature-based and statistical approaches. In their paper [7], they explain that their specialized approaches are suitable for identifying zlib, jpeg, and mp3 fragments with high accuracy (>98 % for fragments of 1500 bytes or more).

4 Conclusion and Outlook

Digital investigations are usually carried out to analyze traces left behind on information systems. Thus, the reconstruction of criminal activities often requires analysis of deleted, hidden or unallocated data. For this reason, the demand for reconstitute evidence from data fragments steadily increases.

Through the variety of different fragment type classification approaches it is difficult to maintain an overview. This paper addresses this issue by introducing a novel taxonomy which aggregates state-of-the-art fragment type classification approaches into the following five dimensions: (1) signature-based approaches, (2) statistical approaches, (3) trainable approaches, (4) content-aware approaches, and (5) other approaches. We further refined the individual categories into sub-categories, where appropriate. For each sub-category we performed a review of existing research and highlighted important representatives in order to provide insights how the taxonomy can be used and to facilitate new researchers to get an overview on the topic.

In the future we plan to further extend the taxonomy with new approaches. Ontologies go one step further than taxonomies by allowing to define (logical) restrictions between relations of elements [89]. Hoss and Carver [90] proposed to support digital forensics analysis by creating ontologies. Therefore, we are currently evaluating how the information of our taxonomy can be enriched to capture more detailed knowledge about fragment classification approaches. Our future research will focus on the development of an ontology which is capable of describing classification approaches in more detail. As the ontology-based representation can processed by information systems, new opportunities regarding the automated combination of classification approaches could arise.

References

1. Beverly, R., Garfinkel, S., Cardwell, G.: Forensic carving of network packet and associated data structures. Digtial Invest. **8**, 78–89 (2011)
2. Ahmed, I., Lhee, K.-S.: Classification of packet contents for malware detection. J. Comput. Virol. **7**(4), 279–295 (2011)
3. Garfinkel, S.L.: Digital forensics research: the next 10 years. Digital Invest. **7**(1), S64–S73 (2010). Proceedings of the Tenth Annual DFRWS Conference
4. Roussev, V., Quates, C., Martell, R.: Real-time digital forensics and triage. Digital Invest. **10**, 20–30 (2013)
5. Young, J., Foster, K., Garfinkel, S., Fairbanks, K.: Distinct sector hashes for target file detection. Computer **45**(12), 28–35 (2012)
6. Shannon, M.M.: Forensic relative strength scoring: ASCII and entropy scoring. Int. J. Digital Evid. **2**(4), 1–19 (2004)
7. Roussev, V., Garfinkel, S.L.: File fragment classification-the case for specialized approaches. In: Proceedings of the: Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE2009), Berkeley, CA, USA, IEEE, pp. 3–14 (2009)
8. Beebe, N.: Digital forensic research: the good, the bad and the unaddressed. In: Peterson, G., Shenoi, S. (eds.) Advances in Digital Forensics V. IFIP AICT, vol. 306, pp. 17–36. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04155-6_2

9. Speirs, W.R., Cole, E.B.: Methods for categorizing input data. U.S. Patent 20 070 116 267, 05 24 (2007)
10. Cao, D., Luo, J., Yin, M., Yang, H.: Feature selection based file type identification algorithm. In: IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS'10), vol. 3, pp. 58–62. IEEE (2010)
11. Fitzgerald, S., Mathews, G., Morris, C., Zhulyn, O.: Using NLP techniques for file fragment classification. Digital Invest. **9**, S44–S49 (2012)
12. Carter, J.M.: Locating executable fragments with concordia, a scalable, semantics-based architecture. In: Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop, Series of CSIIRW '13, pp. 24:1–24:4. ACM, New York (2013)
13. Pal, A., Memon, N.D.: The evolution of file carving. IEEE Sign. Process. Mag. **26**(2), 59–71 (2009)
14. Poisel, R., Tjoa, S., Tavolato, P.: Advanced file carving approaches for multimedia files. J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. (JoWUA) **2**(4), 42–58 (2011)
15. Raghavan, S.: Digital forensic research: current state of the art. CSI Trans. ICT **1**(1), 91–114 (2013)
16. Garfinkel, S.L., Farrell, P., Roussev, V., Dinolt, G.: Bringing science to digital forensics with standardized forensic corpora. Digital Invest. **6**(1), S2–S11 (2009). Proceedings of the Ninth Annual DFRWS Conference
17. Kessler, G.: File signature table, May 2013. http://www.garykessler.net/library/file_sigs.html. Accessed 17 May 2013
18. Li, W., Wang, K., Stolfo, S.J., Herzog, B.: Fileprints: identifying file types by n-gram analysis. In: Proceedings of the Sixth Systems, Man and Cybernetics: Information Assurance Workshop (IAW'05), pp. 64–71. IEEE, New York (2005)
19. file(1). ftp://ftp.astron.com/pub/file/. Accessed 15 April 2013
20. Richard, G.G., Roussev, V.: Scalpel: a frugal, high performance file carver. In: Proceedings of the Fifth Annual DFRWS Conference, New Orleans, LA, pp. 1–10, August 2005. http://www.dfrws.org/2005/proceedings/richard_scalpel.pdf
21. Foremost. http://foremost.sourceforge.net/. Accessed 21 May 2013
22. ReviveIT. https://code.google.com/p/reviveit/. Accessed 21 May 2013
23. PhotoRec. http://www.cgsecurity.org/wiki/PhotoRec. Accessed 15 April 2013
24. Pal, A., Sencar, H.T., Memon, N.D.: Detecting file fragmentation point using sequential hypothesis testing. Digital Invest. **5**(Supplement 1), S2–S13 (2008)
25. Al-Dahir, O., Hua, J., Marziale, L., Nino, J., Richard III, G.G., Roussev, V.: Mp3 scalpel. Technical report, University of New Orleans (2007). http://sandbox.dfrws.org/2007/UNO/uno-submission.doc
26. Garfinkel, S.L., Nelson, A., White, D., Roussev, V.: Using purpose-built functions and block hashes to enable small block and sub-file forensics. Digital Invest. **7**(1), S13–S23 (2010). Proceedings of the Tenth Annual DFRWS Conference
27. National Institute of Standards and Technology, National Software Reference Library (NSRL). http://www.nsrl.nist.gov/. Accessed 15 April 2013
28. Mead, S.: Unique file identification in the national software reference library. Digital Invest. **3**(3), 138–150 (2006)
29. Garfinkel, S.: Lessons learned writing digital forensics tools and managing a 30TB digital evidence corpus. Digital Invest. **9**, S80–S89 (2012)
30. Kim, K., Park, S., Chang, T., Lee, C., Baek, S.: Lessons learned from the construction of a korean software reference data set for digital forensics. Digital Invest. **6**, S108–S113 (2009)

31. Ruback, M., Hoelz, B., Ralha, C.: A new approach for creating forensic hashsets. In: Peterson, G., Shenoi, S. (eds.) Advances in Digital Forensics VIII. IFIP AICT, vol. 383, pp. 83–97. Springer, Heidelberg (2012)
32. Chawathe, S.: Effective whitelisting for filesystem forensics. In: Proceedings of International Conference on Intelligence and Security Informatics (ISI 2009), IEEE, pp. 131–136 (2009)
33. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: Proceedings of the International Conference on Very Large Data Bases, pp. 518–529 (1999)
34. Garfinkel, S.L.: Forensic feature extraction and cross-drive analysis. Digital Invest. **3**, 71–81 (2006)
35. Dandass, Y.S., Necaise, N.J., Thomas, S.R.: An empirical analysis of disk sector hashes for data carving. J. Digital Forensic Pract. **2**, 95–106 (2008). http://www.informaworld.com/10.1080/15567280802050436
36. Collange, S., Dandass, Y.S., Daumas, M., Defour, D.: Using graphics processors for parallelizing hash-based data carving. In: Proceedings of the 42nd Hawaii International Conference on System Sciences, HICSS'09. IEEE, Los Alamitos, pp. 1–10 (2009)
37. Foster, K.: Using distinct sectors in media sampling and full media analysis to detect presence of documents from a corpus. Master's thesis, Naval Postgraduate School, Monterey, California, September 2012
38. Farrell, P., Garfinkel, S., White, D.: Practical applications of bloom filters to the nist rds and hard drive triage. In: 2008 Proceedings of Annual Computer Security Applications Conference, (ACSAC 2008), pp. 13–22 (2008)
39. Roussev, V.: An evaluation of forensic similarity hashes. Digital Investl. **8**, S34–S41 (2011)
40. Roussev, V.: Data fingerprinting with similarity digests. In: Chow, K.P., Shenoi, S. (eds.) Advances in Digital Forensics VI. IFIP AICT, vol. 337, pp. 207–226. Springer, Heidelberg (2010)
41. Kornblum, J.: Identifying almost identical files using context triggered piecewise hashing. Digital Invest. **3**, 91–97 (2006)
42. Roussev, V., Quates, C.: Content triage with similarity digests: the M57 case study. Digital Invest. **9**, S60–S68 (2012)
43. Breitinger, F., Baier, H.: A Fuzzy Hashing Approach based on Random Sequences and Hamming Distance, May 2012, forthcoming issue
44. Baier, H., Breitinger, F.: Security aspects of piecewise hashing in computer forensics. In: 2011 Sixth International Conference on IT Security Incident Management and IT Forensics (IMF), pp. 21–36 (2011)
45. Breitinger, F., Stivaktakis, G., Baier, H.: FRASH: a framework to test algorithms of similarity hashing, August 2013, forthcoming issue
46. Breitinger, F., Astebøl, K.P., Baier, H., Busch, C.: mvHash-B - a new approach for similarity preserving hashing. In: 7th International Conference on IT Security Incident Management & IT Forensics (IMF), Nürnberg, March 2013
47. Breitinger, F., Petrov, K.: Reducing time cost in hashing operations. In: Proceedings of the 9th Annual IFIP WG 11.9 International Conference on Digital Forensics, Orlando, FL, USA, January 2013
48. McDaniel, M.B.: An algorithm for content-based automated file type recognition. Master's thesis, James Madison University (2001)
49. McDaniel, M., Heydari, M.H.: Content based file type detection algorithms. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9, Washington, DC, USA, IEEE CS, p. 332.1 (2003)

50. Dhanalakshmi, R., Chellappan, C.: File format identification and information extraction. In: World Congress on Nature Biologically Inspired Computing, NaBIC, pp. 1497–1501 (2009)
51. Karresand, M.: Completing the picture: fragments and back again. Master's thesis, Linkoepings universitet (2008). http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-11752. Accessed 22 January 2013
52. Karresand, M., Shahmehri, N.: Reassembly of fragmented JPEG images containing restart markers. In: Proceedings of the European Conference on Computer Network Defense (EC2ND), Dublin, Ireland, IEEE CS, pp. 25–32 (2008)
53. Mayer, R.C.: Filetype identification using long, summarized n-grams. Master's thesis, Naval Postgraduate School, Monterey, California, March 2011
54. Collins, M.: Ranking algorithms for named-entity extraction: boosting and the voted perceptron. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Series of ACL '02. Association for Computational Linguistics, Stroudsburg, pp. 489–496 (2002)
55. Karresand, M., Shahmehri, N.: Oscar - file type identification of binary data in disk clusters and RAM pages. In: Fischer-Hübner, S., Rannenberg, K., Yngström, L., Lindskog, S. (eds.) Security and Privacy in Dynamic Environments. IFIP, vol. 201, pp. 413–424. Springer, Heidelberg (2006)
56. Karresand, M., Shahmehri, N.: File type identification of data fragments by their binary structure. In: Proceedings of the IEEE Information Assurance Workshop, pp. 140–147. IEEE, New York (2006)
57. Hall, G., Davis, W.: Sliding window measurement for file type identification. Technical report, Mantech Security and Mission Assurance (2006)
58. Erbacher, R.F., Mulholland, J.: Identification and localization of data types within large-scale file systems. In: Systematic Approaches to Digital Forensic Engineering (SADFE), pp. 55–70 (2007)
59. Moody, S.J., Erbacher, R.F.: Sádi - statistical analysis for data type identification. In: Systematic Approaches to Digital Forensic Engineering (SADFE), pp. 41–54 (2008)
60. Veenman, C.J.: Statistical disk cluster classification for file carving. In: Proceedings of the International Symposium on Information Assurance and Security (IAS'07), Manchester, UK, IEEE CS, pp. 393–398 (2007)
61. Calhoun, W.C., Coles, D.: Predicting the types of file fragments. Digital Invest. **5**, 14–20 (2008)
62. Axelsson, S.: Using normalized compression distance for classifying file fragments. In: Proceedings of the International Conference on Availability, Reliability and Security (ARES 2010), Krakow, Poland, IEEE CS, pp. 641–646 (2010)
63. Axelsson, S.: The normalised compression distance as a file fragment classifier. Digital Investl. **7**, S24–S31 (2010)
64. Savoldi, A., Piccinelli, M., Gubian, P.: A statistical method for detecting on-disk wiped areas. Digital Invest. **8**(3–4), 194–214 (2012)
65. Harris, R.: Arriving at an anti-forensics consensus: examining how to define and control the anti-forensics problem. Digital Invest. **3**, 44–49 (2006)
66. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. Information for the Defense Community, Technical report, May 2001. http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA393366

67. Ariu, D., Giacinto, G., Roli, F.: Machine learning in computer forensics (and the lessons learned from machine learning in computer security). In: Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, Series of AISec '11, pp. 99–104. ACM, New York (2011)
68. Pontello, M.: TrID - File Identifier. http://mark0.net/soft-trid-e.html. Accessed 21 April 2013
69. Ahmed, I., Lhee, K.-S., Shin, H., Hong, M.: Fast file-type identification. In: 2010 Proceedings of the ACM Symposium on Applied Computing, pp. 1601–1602. ACM, New York (2010)
70. Ahmed, I., suk Lhee, K., Shin, H., Hong, M.: Content-based file-type identification using cosine similarity and a divide-and-conquer approach. IETE Tech. Rev. **27**, 465–477 (2010). http://tr.ietejournals.org/text.asp?2010/27/6/465/67149
71. Ahmed, I., Lhee, K.-S., Shin, H.-J., Hong, M.-P.: Fast content-based file type identification. In: Peterson, G.L., Shenoi, S. (eds.) Advances in Digital Forensics VII. IFIP AICT, vol. 361, pp. 65–75. Springer, Heidelberg (2011)
72. Li, Q., Ong, A., Suganthan, P., Thing, V.: A novel support vector machine approach to high entropy data fragment classification. In: Proceedings of the South African Information Security Multi-Conference (SAISMC 2010) (2010)
73. Gopal, S., Yang, Y., Salomatin, K., Carbonell, J.: File-type identification with incomplete information. In: Proceedings of the Tenth Conference on Machine Learning and Applications, Honolulu, Hawaii, IEEE, December 2011
74. Gopal, S., Yang, Y., Salomatin, K., Carbonell, J.: Statistical learning for file-type identification. In: Proceedings of the 10th International Conference on Machine Learning and Applications and Workshops (ICMLA), vol. 1, pp. 68–73 (2011)
75. Sportiello, L., Zanero, S.: File block classification by support vector machines. In: Proceedings of the 6th International Conference on Availability, Reliability and Security (ARES 2011), pp. 307–312 (2011)
76. Beebe, N.L., Maddox, L.A., Liu, L., Sun, M.: Sceadan: using concatentated n-gram vectors for improved data/file type classification (2013, forthcoming issue)
77. Digital Forensics Research Conference (DFRWS), DFRWS 2012 Forensics Challenge (2012). http://www.dfrws.org/2012/challenge/. Accessed 5 April 2013
78. Sportiello, L., Zanero, S.: Context-based file block classification. In: Peterson, G.L., Shenoi, S. (eds.) Advances in Digital Forensics VIII. IFIP AICT, vol. 383, pp. 67–82. Springer, Heidelberg (2012)
79. Amirani, M.C., Toorani, M., Beheshti, A.A.: A new approach to content-based file type detection. In: Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC'08), pp. 1103–1108 (2008)
80. Amirani, M.C., Toorani, M., Mihandoost, S.: Feature-based type identification of file fragments. Secur. Commun. Netw. **6**(1), 115–128 (2013)
81. Kattan, A., Galván-López, E., Poli, R., O'Neill, M.: GP-fileprints: file types detection using genetic programming. In: Esparcia-Alcázar, A.I., Ekárt, A., Silva, S., Dignum, S., Uyar, A.Ş. (eds.) EuroGP 2010. LNCS, vol. 6021, pp. 134–145. Springer, Heidelberg (2010)
82. Garcia, J., Holleboom, T.: Retention of micro-fragments in cluster slack - a first model. In: First IEEE International Workshop on Information Forensics and Security, WIFS 2009, December 2009, pp. 31–35 (2009)
83. Holleboom, T., Garcia, J.: Fragment retention characteristics in slack space - analysis and measurements. In: Proceedings of the 2nd International Workshop on Security and Communication Networks (IWSCN), pp. 1–6, May 2010

84. Blacher, Z.: Cluster-slack retention characteristics: a study of the NTFS filesystem. Master's thesis, Karlstad University, Faculty of Economic Sciences, Communication and IT (2010)
85. Xu, M., Yang, H.-R., Xu, J., Xu, Y., Zheng, N.: An adaptive method to identify disk cluster size based on block content. Digital Invest. **7**(1–2), 48–55 (2010)
86. Li, Q.: Searching and extracting digital image evidence. In: Sencar, H.T., Memon, N. (eds.) Digital Image Forensics, pp. 123–153. Springer, New York (2013)
87. Conti, G., Bratus, S., Shubina, A., Lichtenberg, A., Ragsdale, R., Perez-Alemany, R., Sangster, B., Supan, M.: A visual study of primitive binary fragment types. White Paper, Black Hat USA 2010, Technical report, United States Military Academy, July 2010
88. Conti, G., Bratus, S., Shubina, A., Sangster, B., Ragsdale, R., Supan, M., Lichtenberg, A., Perez-Alemany, R.: Automated mapping of large binary objects using primitive fragment type classification. Digital Invest. **7**, S3–S12 (2010)
89. Noy, N.F., McGuinness, D.L.: Ontology development 101: a guide to creating your first ontology (2001). http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html. Accessed 22 January 2013
90. Hoss, A., Carver, D.: Weaving ontologies to support digital forensic analysis. In: IEEE International Conference on Intelligence and Security Informatics, ISI '09, pp. 203–205, June 2009