THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 12 | Number 2 Article 6

6-30-2017

File Type Identification - Computational Intelligence for Digital Forensics

Konstantinos Karampidis
Technological Educational Institute of Crete

Giorgos Papadourakis
Technological Educational Institute of Crete

Recommended Citation

Karampidis, Konstantinos and Papadourakis, Giorgos (2017) "File Type Identification - Computational Intelligence for Digital Forensics," *Journal of Digital Forensics, Security and Law*: Vol. 12 : No. 2 , Article 6.
DOI: https://doi.org/10.15394/jdfsl.2017.1472
Available at: https://commons.erau.edu/jdfsl/vol12/iss2/6

FILE TYPE IDENTIFICATION – COMPUTATIONAL INTELLIGENCE FOR DIGITAL FORENSICS

Konstantinos Karampidis, Giorgos Papadourakis
Technological Educational Institute of Crete
Department of Informatics Engineering
Heraklion Crete

ABSTRACT

In modern world, the use of digital devices for leisure or professional reasons is growing quickly; nevertheless, criminals try to fool authorities and hide evidence in a computer by changing the file type. File type detection is a very demanding task for a digital forensic examiner. In this paper, a new methodology is proposed – in a digital forensics perspective- to identify altered file types with high accuracy by employing computational intelligence techniques. The proposed methodology is applied to the three most common image file types (jpg, png and gif) as well as to uncompressed tiff images. A three-stage process involving feature extraction (Byte Frequency Distribution), feature selection (genetic algorithm) and classification (neural network) is proposed. Experimental results were conducted having files altered in a digital forensics perspective and the results are presented. The proposed model shows very high and exceptional accuracy in file type identification.

Keywords: digital forensics, file type identification, computational intelligence, genetic algorithm, neural network, data integrity

1. INTRODUCTION

Digital forensics is a relatively new field in Computer Science and focuses on the acquisition, preservation and analysis of digital evidence. Palmer defined digital forensics as "the use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation, and presentation of digital evidence derived from digital sources for the purpose of facilitation or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations." (Palmer, 2001). Identification of the evidence is one of the most important and difficult stages during a forensic examination of the acquired data. File type detection methods can be categorized into three kinds: extension-based, magic bytes-based, and content-based methods (Meghanathan, Boumerdassi, Chaki, & Dhinaharan Nagamalai, 2010). Each of them has its own advantages and weaknesses, and none of them are comprehensive or infallible enough to satisfy all the requirements. The fastest and easiest method of file type detection is the extension-based method.

The main advantage of this method is the speed of file type detection. In the extension based method there is no need to open the file in order to determine the file type. Nevertheless, it has great vulnerability while it can be easily fooled by a simple extension renaming. As soon as a forensic program perceives such a deception, it will immediately highlight an extension mismatch. This method is a common concept in Windows operating systems. In other operating systems such as Linux based ones, the issue of the file command shows information about the type of a file. The second method of file type detection is based on the magic bytes. Magic bytes are predefined signatures and they can be found on file's header. There are several thousand's file types for which magic bytes are defined and listed (Kessler, 2015) and there are multiple lists of magic bytes that are not completely consistent. Checking the magic bytes of a file is indeed a much slower method than just checking its extension since the file should be opened and its magic bytes should be read and compared with the predefined ones. One major drawback of this method is the lack of a predefined standard for the developers, so the magic bytes are not used in all file types. Moreover, magic bytes only work on the binary files and predefined signatures differ in length for unlike file types. When a digital media with files of amended signature is attached, the forensic software will indicate the deception and suggest the forensic analyst the true file type. The third method of file type detection is the examination of file contents and the use of statistical modeling techniques to achieve detection. It is a new and promising research area and it is likely the only way to determine the bogus file types. McDaniel and Heydari (McDaniel, 2001), (McDaniel & Heydari, 2003) were the first who actually suggested a way for content-based file type detection. They proposed three different algorithms for the content-based file type detection. The accuracy varied from 23% to 96% depending upon the algorithm used. Li et al. (Li, Wang, Stolfo, & Herzog, 2005) made a few changes on McDaniel's and Heydari's method, in order to improve its accuracy. They proposed to compute a set of centroid models and use clustering to find a minimal set of centroids with good performance while the use of more pattern data is necessary. This approach resulted to 82% accuracy (one centroid), 89.5% accuracy (multi-centroid) and 93.8% accuracy (more exemplar files). Dunham et al. (Dunham, Sun, & Tseng, 2005) used neural networks for classification and achieved 91.3% accuracy. Amirani et al. (Amirani, Toorani, & Shirazi, 2008) used the Principal Component Analysis and unsupervised neural networks for the automatic feature extraction. The classifier they used was a neural network, achieving an accuracy of 98.33% which was the best so far. Cao et al. (Cao, Luo, Yin, & Yang, 2010) used Gram Frequency Distribution and vector space model with results of 90.34% accuracy. Ahmed et al. (Ahmed, Lhee, Shin, & Hong, 2010) proposed two very interesting methods. Primary: they used the cosine distance as a similarity metric when comparing the file content. Subsequent: they decomposed the identification procedure into two steps. They used 2000 files of 10 file types as a dataset and achieved an accuracy of 90.19%. Ahmed et al. (Ahmed, Lhee, Shin, & Hong, 2011) also proposed two new techniques to reduce the classification time. The first method was a feature selection technique and the K-nearest neighbor (KNN) classifier was used. The second method was the content sampling technique, which used a small portion of a file to obtain its byte-frequency distribution. Amirani et al. (Amirani, Toorani, & Mihandoost, 2013) then proposed an improved version of their first approach by using a

Support Vector Machine classifier and finally succeeded in raising the accuracy of the method to 99.16%. Finally, Evensen et al. (Evensen, Lindahl, & Goodwin, 2014) used an n-gram analysis with naïve Bayes classifier to a large dataset of 60000 files (6 file types) with very good results achieving 99.51% topmost. The above papers refer to identification of whole files. Moreover, methods for identifying types of fragments are also proposed by scientists and both (whole files and fragments) are documented in detail (Karampidis, Papadourakis, & Deligiannis, 2015). The above methods showed poor to good results in file type identification, but the real problem during a forensic examination relies on the modification of file's signature and its extension at the same time. When this occurs the majority – if not all- of the forensic software cannot identify correctly the file type. In this paper, a new methodology is proposed for file type identification using computational intelligence techniques in order to identify the correct file type if the file is altered, i.e. both file's extension and magic bytes are altered. The paper is organized as follows: In Section 2 the proposed methodology is described, then in Section 3 a large dataset is utilized and the experimental results are presented followed by conclusions.

2. METHODOLOGY OF THE PROPOSED METHOD

The proposed methodology uses computational intelligence techniques in order to identify the file type and to reveal the correct type if the file is altered. It is a three-stage process involving feature extraction, feature selection and classification, as illustrated in Figure 1. Initially all files from the dataset are loaded and the features are extracted. Afterwards, feature selection is accomplished using a genetic algorithm and finally a neural network performs the classification. Byte Frequency Distribution (BFD) is used as a feature extraction method. In order to create the BFD, the number of occurrences of each byte value in an input file is counted and an array with elements from 0 to 255 is created. Then each element of the array is normalized by dividing with the maximum occurrence. The final result is a file containing 256 features for each instance. The next stage is feature selection, in order to decrease the number of features. Feature selection is the procedure of finding and selecting the minimum number of the most informative relevant features.

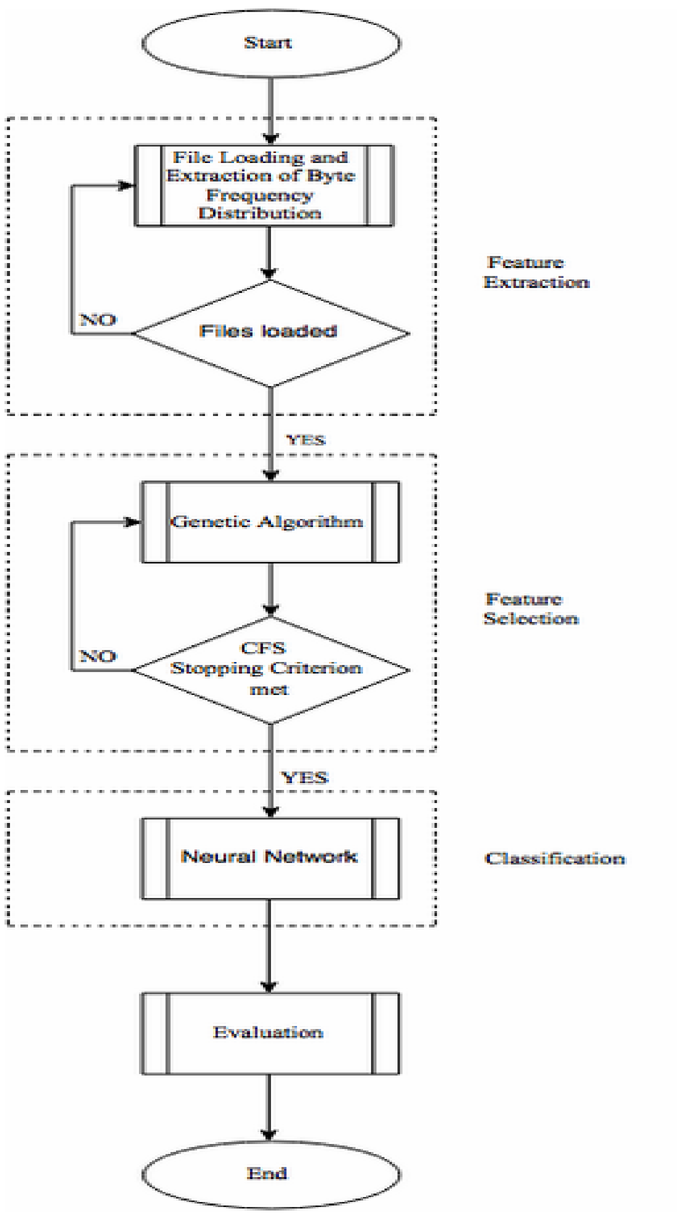

Figure 1. Flowchart of the proposed method

As a search method, a genetic algorithm was used. The idea of using a genetic algorithm, for feature extraction is not new (Vafaie & Jong, 1992), (Zhuo Li, Zheng Jing, Wang Fang, Li Xia, Ai Bin, 2008), (Jourdan, Dhaenens, & Talbi, 2001) since they can provide candidate solutions. Each candidate solution (chromosome) is represented by a binary feature vector of dimension 256, where zero (0) indicates that the respective feature is not selected, and one (1) indicates that the feature is selected. The score of each candidate solution is evaluated by a fitness function. As a fitness function the Correlation based Feature Selection (CFS) (MA Hall, 1999) algorithm is utilized. This algorithm evaluates the candidate solutions from the genetic algorithm and choses those which include features highly associated to the file type category and low correlated with each other, by calculating each candidate's solution merit. Let S be a candidate solution consisting of k features. The merit of each candidate solution is calculated by equation 1.

$$\text{Merits}_k = \frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}} \qquad (1)$$

where:

$\overline{r_{cf}}$ is the average value of all feature-classification correlations and

$\overline{r_{ff}}$ is the average value of all feature-feature correlations.

CFS stops when five consecutive fully expanded candidate solutions show no improvement (MA Hall, 1999). The utilization of the genetic algorithm as a search method and CFS as an evaluator, led to the reduction of the 256 extracted features to 44. The third and final stage is classification, which was performed with a one hidden layer neural network using the backpropagation algorithm. A neural network with one hidden layer was also used by Harris (Harris, 2007) in order to identify file types. Initially, the data are separated into a training set (70%) and a test set (30%).

Furthermore, in order to estimate the accuracy of classification during the training phase a stratified 10 fold cross validation is used (Kohavi, 1995). Subsequently, unseen instances from all categories are presented to the model for evaluation.

3. EXPERIMENTAL SETUP AND RESULTS

Due to thousands of known file types, this research has focused only in images and portable documents, because of their significance to Digital Forensics. In particular, this research only included jpeg png, gif (not animated), tiff and pdf files. Furthermore, only whole files and not fragments were examined. Caltech 101 (Fei-Fei, Fergus, & Perona, 2007) was used as dataset. It is a dataset made by Caltech University containing 9144 images in jpeg format from 101 categories. These images are in 101 subfolders each one representing one category; therefore, the images were extracted to a single folder. From this jpeg dataset, 5519 images were utilized. Afterwards, these images were renamed (image 0001 to image 5519) and one third of them were converted to png format and a similar number to gif format. Total Image Converter (CoolUtils, 2017) -free trial version- was used, in order to convert those files. There were no alterations to converted files regarding size, rotation, crop, further compression, filtering, transparency or watermark embedded. Table 1 shows the conversion parameters used.

The dataset was divided into a training set (70%) and a test set (30%) and the exact numbers of any file type used from Caltech dataset in both sets, are shown in Table 2. Additionally, 1840 pdf files were added, which were open access undergraduate theses found online from the library of the Technological Educational Institute of Crete (T.E.I of Crete, 2015). The final dataset is uniformly distributed and its exact numbers are indicated in Table 3. In order to examine if the proposed methodology identifies the correct file type when the file is altered, one third of the testing pdf files (168) were replaced by image files and their extension and signature was changed to pdf.

Table 1
Conversion Parameters

From jpg	To png	To gif
Image size changed	No	No
Rotation of the image	No	No
The image was cropped	No	No
Image was compressed	No	-
Any filter applied	No	-
Transparency of source file used	Yes	Yes
Watermark embedded	No	No

Table 2
Images utilized from Caltech Dataset

Type	**Training Set**		**Test Set**	
	Number of images	Image Number	Number of images	Image Number
jpg	1288	0001-1288	552	1289-1840
png	1288	1841-3128	552	3129-3680
gif	1287	3681-4967	552	4968-5519
Total	**3863**		**1656**	

Table 3
The final dataset

Dataset			
Total files		**Training**	**Testing**
jpeg	1840	1288	552
png	1840	1288	552
gif	1839	1287	552
pdf	1840	1288	552
Total	**7359**	**5151**	**2208**

More specifically, the extension of 168 jpeg images was changed from .jpg to .pdf. Also with a hex editor the signature of each jpeg image was also changed. The same procedure was performed to png and gif images and therefore three new test sets were created. The first contained 168 altered files of jpeg format, the second contained 168 files of png format and the third contained 168 files of gif format. Table 4 shows the changes made to the 168 files in each new test set.

A script written in MATLAB® (The MathWorks Inc., 2016) was implemented to create the BFD containing 256 features. Waikato Environment for Knowledge Analysis (Weka) (Mark Hall et al., 2009), a popular machine learning software developed at the University of Waikato, New Zealand was used for all the experiments. Weka uses Goldberg's Genetic Algorithm (Goldberg, 1989). The population size was 256, the number of generations 100, crossover was set to 0.8 and mutation probability to 0.033. CFS was the fitness function, roulette wheel selection was used to probabilistically select individuals and the single-point crossover operator was selected. The use of CFS as a filter selection evaluator and the genetic algorithm as a search strategy resulted to the selection of 44 features (82.81% reduction). A multilayer neural network using the backpropagation algorithm was implemented as a classifier in Weka. The neural network consisted of one hidden layer with 3 nodes. The number of inputs was the 44 selected features and the number of outputs the four possible categories; namely jpeg, png, gif, and pdf. The learning rate was set to 0.3 and in order to avoid local minimum and to accelerate the learning process, the momentum parameter was set to 0.2. The training time (epochs) after experimentation was set to 500. When the training of the neural network was completed the three test sets described previously were evaluated and the results are shown in Tables 5, 6, and 7.

Table 4
Changes made to image files

From		**To**	
Extension	Signature	Extension	Signature
jpg	FF D8 FF E0 xx xx 4A 46 49 46 00	pdf	25 50 44 46
png	89 50 4E 47 0D 0A 1A 0A	pdf	25 50 44 46
gif	47 49 46 38 37 61	pdf	25 50 44 46

Table 5
Confusion matrix – Identifying forged jpg images

Test set	**Classified as**			
Image Type	**jpg**	**pdf**	**png**	**gif**
jpg	552	0	0	0
pdf (168 jpg)	171	377	2	2
png	0	3	548	1
gif	0	1	7	544

Table 6
Confusion matrix – Identifying forged png images

Test set	**Classified as**			
Image Type	**jpg**	**pdf**	**png**	**gif**
jpg	552	0	0	0
pdf (168 png)	3	379	168	2
png	0	3	548	1
gif	0	1	7	544

Table 7
Confusion matrix – Identifying forged gif images

Test set	**Classified as**			
Image Type	**jpg**	**pdf**	**png**	**gif**
jpg	552	0	0	0
pdf (168 gif)	3	377	2	170
png	0	3	548	1
gif	0	1	7	544

Table 5 shows the confusion matrix when the neural network tried to identify forged jpg images (168). When the output of the neural network were compared to the testing dataset, the "misclassified" files were the altered jpg images. The accuracy of the proposed method to altered jpg images was 100%. Table 6 shows the confusion matrix when the neural network tried to identify forged png images (168) and 166 out of 168 images were detected. Two png images were wrongly identified as pdf files. In the two misclassified png images there were large areas of a specific color or small variations of a color. Small variations of a color can be found also on pdf files, which led to misclassification of the images. Therefore 2 out of 168 png altered files were not predicted correctly. The accuracy of the proposed method to altered png images was 98.81%. Table 7 shows the confusion matrix when the neural network tried to identify forged gif images (168). The "misclassified" files were the altered gif images, thus the accuracy of the proposed method in this case was 100%. The accuracy results for the altered images (jpg, png, gif) of the proposed method are summarized in Table 8.

Table 8
Final Confusion Matrix of the proposed method

168 Forged Files	**Classified as**			
Actual Type	**jpg**	**pdf**	**png**	**gif**
jpg	168	0	0	0
png	0	2	166	0
gif	0	0	0	168

The above results showed that a very simple neural network achieved excellent results so a traditional clustering method such as the k-means algorithm was implemented in order to examine its accuracy. The k-means algorithm was implemented in Weka and the three test sets were clustered into four categories. The algorithm first computed randomly the initial centers of the four clusters, then assigned every instance of the testing file to the cluster whose center was the closest to that instance, by calculating the Euclidean distance. This was repeated until the assignment of the instances has not been changed during one iteration. The output of the clustering algorithm was compared to altered files of the three testing sets and the predictions of this method are summarized in Table 9.

Table 9
Results of the k-means algorithm

Altered Type (168 altered files)	**Clustered as**			
	jpg	**pdf**	**png**	**gif**
jpg	1	99	0	68
png	4	84	72	8
gif	30	3	1	134

The accuracy of the k-means algorithm to altered jpg images was 0,006%, to altered png images 42,85% and to gif images 79,76%. The clustering method failed to identify correctly the exact type of the altered files.

Moreover, in order to ensure that the proposed model had not learned to detect the software used (Total Image Converter) when converting jpg images to gif and png format, we repeated the experiment using this time Pixillion (NCH Software, 2017) (free version) another popular converter. Again, no alterations made to the converted images regarding size, crop, filtering or embedding watermark. The same procedure was repeated to create the final dataset, a new classification model was trained again and the resulted confusion matrix when the unseen instances from the three test sets with the forged files was presented to the model is shown in Table 10.

Table 10
Confusion Matrix of the proposed method

168 Forged Files	**Classified as**			
Actual Type	**jpg**	**pdf**	**png**	**gif**
jpg	168	0	0	0
png	0	8	160	0
gif	0	0	0	168

Comparing the two confusion, matrices i.e. Table 8 and Table 10, it is obvious that the proposed model shows extremely high accuracy, regardless the software used to convert jpg images to png and gif format.

The proposed method identified with high accuracy images (jpg,png,gif) which are the most common in digital mediums (e.g. computer, tablet, smartphones etc.). Although tiff images are not widely used and by a digital forensics viewpoint are not frequently met, it was examined if the proposed method worked as well as for uncompressed tiff images and whether the proposed model depends on file compression. For this, a new dataset was created replacing the gif images with uncompressed tiff images and the proposed methodology was applied. Once more, 30% of the pdf files in the testing set were tiff images with altered extension and signature (in a digital forensics viewpoint). Table 11 shows the confusion matrix when the neural network tried to identify forged tiff images (168).

Table 11

Confusion matrix – Identifying forged tiff images

Test set	**Classified as**			
Image Type	**jpg**	**pdf**	**png**	**tiff**
jpg	552	0	0	0
pdf (168 tiff)	3	367	9	173
png	0	3	545	4
tiff	4	4	4	540

Only three altered tiff images were misclassified. Two of them were classified as png images and one as pdf file. Images which misclassified as png, had high color depth and this led to misclassification. The image which misclassified as pdf, had large areas of a specific color, something also found on pdf files. Thus, the accuracy of the proposed method in uncompressed tiff images was 98.21%.

4. CONCLUSIONS

In this paper, a new methodology was proposed – in a digital forensics perspective- to identify altered file types with high accuracy by employing computational intelligence techniques. The proposed methodology was applied to the three most common image file types (jpg, png and gif) as well as to uncompressed tiff images. A three-stage process involving feature extraction (BFD), feature selection (genetic algorithm), and classification (neural network) was proposed. Experimental results were conducted having files altered in a digital forensics perspective. The accuracy of the proposed method to altered jpg images and to gif images was 100%, to altered png images was 98,81% and to altered tiff images was 98,21%. An attempt to replace the neural network with the traditional k-means clustering algorithm in the proposed methodology gave poor results. Experiments also were conducted regarding the scenario the proposed model learned to detect specific converter and the results were promising again. The proposed model showed extremely high accuracy regardless the conversion tool used.

REFERENCES

Ahmed, I., Lhee, K., Shin, H., & Hong, M. (2010). Content-based File-type Identification Using Cosine Similarity and a Divide-and-Conquer Approach. *IETE Technical Review*, *27*(6), 465. https://doi.org/10.4103/0256-4602.67149

Ahmed, I., Lhee, K., Shin, H., & Hong, M. (2011). Fast content-based file-type identification. In *7th Annual IFIP WG 11.9 International Conference on Digital Forensics* (pp. 65–75). Springer Boston. https://doi.org/10.1007/978-3-642-24212-0_5

Amirani, M. C., Toorani, M., & Mihandoost, S. (2013). Feature-based Type Identification of File Fragments. *Security and Communication Networks*, *6*(1), 115–128. https://doi.org/10.1002/sec.553

Amirani, M. C., Toorani, M., & Shirazi, a. a B. (2008). A new approach to content-based file type detection. In *IEEE Symposium on Computers and Communications* (pp. 1103–1108). https://doi.org/10.1109/ISCC.2008.4625611

Cao, D., Luo, J., Yin, M., & Yang, H. (2010). Feature selection based file type identification algorithm. In *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems* (Vol. 3, pp. 58–62). IEEE. https://doi.org/10.1109/ICICISYS.2010.5658559

CoolUtils. (2017). Powerful Image Converter Yet Easy-to-use. Retrieved from https://www.coolutils.com/TotalImageConverter

Dunham, J., Sun, M., & Tseng, J. (2005). Classifying file type of stream ciphers in depth using neural networks. In *The 3rd ACS/IEEE International Conference on Computer Systems and Applications.* Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1387088

Evensen, J. D., Lindahl, S., & Goodwin, M. (2014). File-type Detection Using Naïve Bayes and n-gram Analysis. *Norwegian Information Security Conference, NISK.* Fredrikstad. Retrieved from http://ojs.bibsys.no/index.php/NISK/article/view/99

Fei-Fei, L., Fergus, R., & Perona, P. (2007). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, *106*(1), 59–70. https://doi.org/10.1016/j.cviu.2005.09.012

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Retrieved from http://dl.acm.org/citation.cfm?id=534133

Hall, M. (1999). *Correlation-based feature selection for machine learning.* The University of Waicato. Retrieved from http://www.cs.waikato.ac.nz/~mhall/thesis.pdf

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10. https://doi.org/10.1145/1656274.1656278

Harris, R. (2007). Using artificial neural networks for forensic file type identification. *Master's Thesis, Purdue University.* Retrieved from https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/bibtex_archive/2007-19.ps

Jourdan, L., Dhaenens, C., & Talbi, E. (2001). A genetic algorithm for feature selection in data-mining for genetics. In *Proceedings of the 4th Metaheuristics International Conference*. Retrieved from ftp://155.253.6.100/acalabria/PhD/Materiale/MachineLearning/Jourdan 2001 - A Genetic Algorithm for Feature Selection in Data-Mining for Genetics.pdf

Karampidis, K., Papadourakis, G., & Deligiannis, I. (2015). File Type Identification -A Literature Review. In *9th International Conference on New Horizons in Industry Business and Education, NHIBE 2015* (p. 141). Skiathos, Greece: 9th International Conference on New Horizons in Industry Business and Education. Retrieved from http://nhibe2015.vs-net.eu/proceedings/papers/3_15_[P]0076.pdf

Kessler, G. (2015). File Signatures. Retrieved October 26, 2015, from http://www.garykessler.net/library/file_sigs.html

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence*, *14*(12), 1137–1143. https://doi.org/10.1067/mod.2000.109031

Li, W. J., Wang, K., Stolfo, S. J., & Herzog, B. (2005). Fileprints: Identifying file types by n-gram analysis. *Proceedings from the 6th Annual IEEE System, Man and Cybernetics Information Assurance Workshop, SMC 2005*, *2005*(June), 64–71. https://doi.org/10.1109/IAW.2005.1495935

McDaniel, M. (2001). *Automatic File Type Detection Algorithm.* James Madison University.

McDaniel, M., & Heydari, M. H. (2003). Content based file type detection algorithms. *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the.* https://doi.org/10.1109/HICSS.2003.1174905

Meghanathan, Boumerdassi, S., Chaki, N., & Dhinaharan Nagamalai. (2010). Recent Trends in Network Security and Applications (Vol. 89, pp. 253–262). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-14478-3

NCH Software. (2017). Convert Between All Popular Image Formats with Pixillion. Retrieved from http://www.nchsoftware.com/imageconverter/

Palmer, G. (2001). *A Road Map for Digital Forensic Research. Proceedings of the 2001 Digital Forensics Research Workshop (DFRWS 2004).* https://doi.org/10.1111/j.1365-2656.2005.01025.x

T.E.I of Crete. (2015). E-Thesis. Retrieved October 26, 2015, from http://nefeli.lib.teicrete.gr/search/

The MathWorks Inc. (2016). MATLAB. Natick, Massachusetts: The MathWorks Inc. Retrieved from http://www.mathworks.com/

Vafaie, H., & Jong, K. De. (1992). Genetic Algorithms as a Tool for Feature Selection in Machine Learning. In *International Conference on Tools with AI* (pp. 200–203). https://doi.org/10.1109/TAI.1992.246402

Zhuo Li, Zheng Jing, Wang Fang, Li Xia, Ai Bin, Q. J. (2008). A genetic algorithm based wrapper feature selection method for classification of hyper spectral data using support vector machine.

GEOGRAPHICAL RESEARCH, *27*(3), 493–501. https://doi.org/10.11821/yj2008030002