File Type Identification for Digital Forensics

Konstantinos Karampidis(✉) and Giorgos Papadourakis

Department of Informatics Engineering, Technological Educational
Institute of Crete, Heraklion, Crete, Greece
karampidis@outlook.com, papadour@cs.teicrete.gr

Abstract. In modern world the use of digital devices for leisure or professional reasons (computers, tablets and smartphones etc.) is growing quickly. Nevertheless, criminals try to fool authorities and hide evidence in a computer or any other digital device, 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 in the four most common types of files (jpg, png and gif). 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 · Forensic examiner · Computational intelligence · Genetic algorithm

1 Introduction

Digital forensics is a relatively new field in Computer Science and focuses on the acquisition, preservation and analysis of digital evidence. Palmer [1] 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”. 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 [2]. Each of them has its own advantages and weaknesses, and none of them are comprehensive or infallible enough to satisfy all the requirements.

The original version of this chapter was revised.
An erratum to this chapter can be found at 10.1007/978-3-319-39564-7_26

J. Krogstie et al. (Eds.): CAiSE 2016, LNBIP 249, pp. 266–274, 2016.
DOI: 10.1007/978-3-319-39564-7_25

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.

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 [3] and there are multiple lists of magic bytes that are not completely consistent. Checking the magic bytes of a file is indeed 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 to 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 [4, 5] 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. [6] 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. [7] used neural networks for classification and achieved 91.3 % accuracy. Amirani et al. [8] 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. [9] used Gram Frequency Distribution and vector space model with results of 90.34 % accuracy. Ahmed et al. [10] 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. [11] 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. [12] 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. [13] 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 [14].

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 Sect. 2 the proposed methodology is described, then in Sect. 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 Fig. 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. As a search method a genetic algorithm was used. The idea of using a genetic algorithm, for feature extraction is not new [15–17] 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) [18] 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 as shown in Eq. 1.

$$\mathrm{Merits}_{\mathrm{k}} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \tag{1}$$

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.

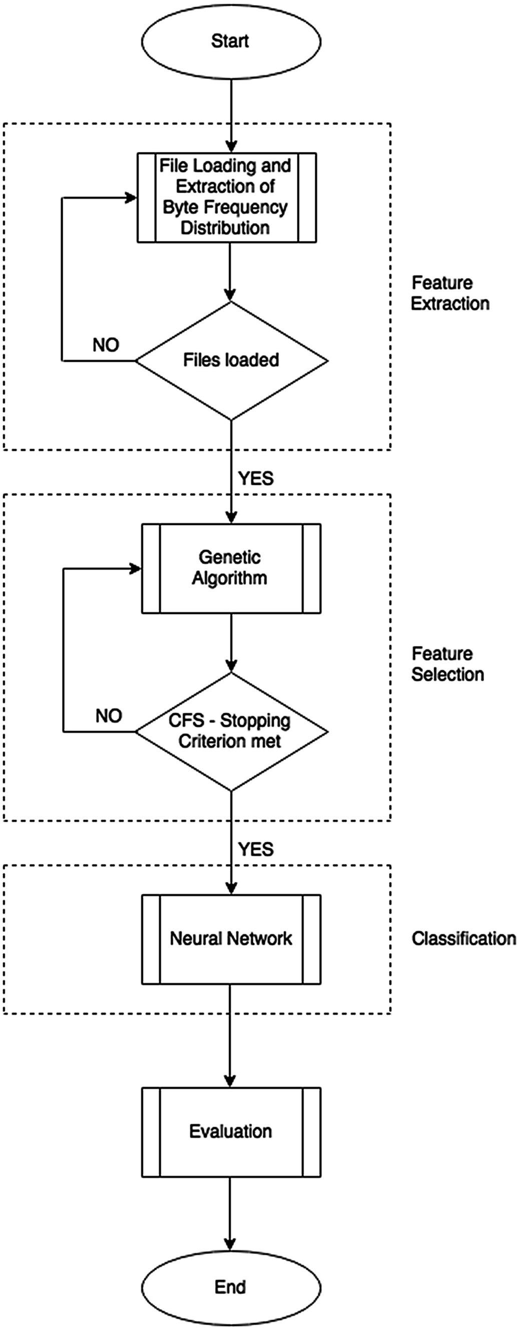

Fig. 1. Flowchart of the proposed method

CFS stops when five consecutive fully expanded candidate solutions show no improvement [18]. 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, performed with a one hidden layer neural network using the backpropagation algorithm. A neural network with one hidden layer was also used by Harris [19] 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 [20]. 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 have 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) and pdf files. Furthermore, only whole files and not fragments were examined. Caltech 101 [21] was used as dataset. It is a dataset made by Caltech University and contains 9144 images in jpeg format from 101 categories. From this jpeg dataset, 5519 images were utilized. One third of these files were converted to png format and a similar number to gif format. The dataset was divided into a training set (70 %) and a test set (30 %). Additionally, 1840 pdf files were added, which were open access undergraduate theses found online from the library of the Technological Educational Institute of Crete [22]. The created dataset is uniformly distributed and its exact numbers are indicated in Table 1. In order to examine if the proposed methodology identifies the correct file type if 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. Three new test sets were created where 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 1. The 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

A script written in MATLAB® [23] was implemented to create the BFD containing 256 features. Waikato Environment for Knowledge Analysis (Weka) [24], 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 [25]. 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 2, 3 and 4.

Table 2. Confusion matrix – identifying forged jpg images

Test set	Classified as			
Image type	jpg	pdf	png	gif
jpg	552	0	0	0
pdf	3	377	2	2
forged pdf (actual jpg)	168	0	0	0
png	0	3	548	1
gif	0	1	7	544

Table 2 shows the confusion matrix when the neural network tried to identify forged jpeg 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 3. Confusion matrix – identifying forged png images

Test set	Classified as			
Image type	jpg	pdf	png	gif
jpg	552	0	0	0
pdf	3	377	2	2
forged pdf (actual png)	0	2	166	0
png	0	3	548	1
gif	0	1	7	544

Table 3 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 4. Confusion matrix – identifying forged gif images

Test set	Classified as			
Image type	jpg	pdf	png	gif
jpg	552	0	0	0
pdf	3	377	2	2
forged pdf (actual gif)	0	0	0	168
png	0	3	548	1
gif	0	1	7	544

Table 4 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 5.

Table 5. Final confusion matrix of the proposed method

168 Forged images	Classified as			
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. In order to examine how well the proposed model identifies the other file types and not only the forged ones, the 168 forged pdf files were replaced by other normal ones (not forged pdf files). The same dataset was utilized and the same methodology was applied. The resulted confusion matrix and the detailed accuracy for every class (True Positive Rate, False Positive Rate, Precision and Recall) are shown on Tables 6 and 7.

Table 6. Confusion matrix of the classifier

	Classified as			
Actual file type	jpg	pdf	png	gif
jpg	552	0	0	0
pdf	3	545	2	2
png	0	3	548	1
gif	0	1	7	544

Table 7. Detailed accuracy by class

Class	True positive rate	False positive rate	Precision	Recall
jpg	1	0.002	0.995	1
pdf	0.987	0.002	0.993	0.987
png	0.993	0.005	0.984	0.993
gif	0.986	0.002	0.995	0.986

The results showed that the proposed model identified very well the other file types, as well as the forged ones.

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 in the four most common types of files (jpg, png and gif). 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 % and to altered png images was 98,81 %. Furthermore, the proposed methodology was also applied to the other file types with excellent results as well.

References

1. Palmer, G.: A road map for digital forensic research. In: Proceedings of the 2001 Digital Forensic Research Workshop (DFRWS 2004), pp. 1–42 (2001)
2. Meghanathan, N., Boumerdassi, S., Chaki, N., Nagamalai, D. (eds.): Recent Trends in Network Security and Applications. Springer, Heidelberg (2010)
3. Kessler, G.: File Signatures. http://www.garykessler.net/library/file_sigs.html
4. McDaniel, M.: Automatic File Type Detection Algorithm (2001)
5. McDaniel, M., Heydari, M.H.: Content based file type detection algorithms. In: 2003 Proceedings of the 36th Annual Hawaii International Conference System Sciences (2003)
6. Li, W.J., Wang, K., Stolfo, S.J., Herzog, B.: Fileprints: identifying file types by n-gram analysis. In: Proceedings from the 6th Annual IEEE Systems, Man, and Cybernetics Information Assurance Workshop SMC 2005, pp. 64–71 (2005)
7. Dunham, J., Sun, M., Tseng, J.: Classifying file type of stream ciphers in depth using neural networks. In: The 3rd ACS/IEEE International Conference on Computer Systems and Applications (2005)
8. Amirani, M.C., Toorani, M., Shirazi, A.A.B.: A new approach to content-based file type detection. In: Proceedings of the IEEE Symposium on Computers and Communications, pp. 1103–1108 (2008)

9. Cao, D., Luo, J., Yin, M., Yang, H.: Feature selection based file type identification algorithm. In: 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, pp. 58–62. IEEE (2010)
10. Ahmed, I., 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 (2010)
11. Ahmed, I., Lhee, K.-S., Shin, H.-J., Hong, M.-P.: Fast content-based file type identification. In: Peterson, G., Shenoi, S. (eds.) Advances in Digital Forensics VII. IFIP AICT, vol. 361, pp. 65–75. Springer, Heidelberg (2015)
12. Amirani, M.C., Toorani, M., Mihandoost, S.: Feature-based type identification of file fragments. Secur. Commun. Netw. **6**, 115–128 (2013)
13. Evensen, J.D., Lindahl, S., Goodwin, M.: File-Type Detection Using Naïve Bayes and n-gram Analysis (2014). http://ojs.bibsys.no/index.php/NISK/article/view/99
14. Karampidis, K., Papadourakis, G., Deligiannis, I.: File type identification – a literature review. In: Proceedings of 9th International Conference on New Horizons in Industry Business and Education, NHIBE 2015, p. 141, Skiathos, Greece (2015)
15. Vafaie, H., De Jong, K.: Genetic algorithms as a tool for feature selection in machine learning. In: International Conference on Tools with AI, pp. 200–203 (1992)
16. Zhuo, L., Zheng, J., Wang, F., Li, X., Ai, B., Qian, J.: A genetic algorithm based wrapper feature selection method for classification of hyper spectral data using support vector machine. Geogr. Res. **27**, 493–501 (2008)
17. Jourdan, L., Dhaenens, C., Talbi, E.: A genetic algorithm for feature selection in data-mining for genetics. In: Proceedings of the 4th Metaheuristics International Conference (2001)
18. Hall, M.: Correlation-based feature selection for machine learning (1999). http://www.cs.waikato.ac.nz/~mhall/thesis.pdf
19. Harris, R.: Using artificial neural networks for forensic file type identification. Master's thesis, Purdue University (2007)
20. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: 14th International Joint Conference on Artificial Intelligence, pp. 1137–1143 (1995)
21. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Generative Model Based Vision 2004, p.178 (2004)
22. T.E.I of Crete: E-Thesis. http://nefeli.lib.teicrete.gr/search/
23. The MathWorks Inc.: MATLAB. http://www.mathworks.com/
24. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software. ACM SIGKDD Explor. Newsl. **11**, 10 (2009)
25. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Boston (1989)