Carvml: Classification Of File Fragments To Predict And Analyze File Types

K.Lokesh[1], Dr. S.A.Sahaaya Arul Mary[2], T.ManojKumar[3],B.Jayaraj[4] , M.GopalaKrishnan[5]
[1] Dept of CSE
[2]HOD, Dept of CSE
[1, 2] Saranathan College of Engineering, Trichy, Tamil Nadu, India

Abstract- *We present with a modern cognitive algorithmic approach to solve the never-ending problem of identifying the file types of file fragments. The main purpose of using the "File Carving" technique is to reconstruct partially erased file fragments on disk into whole files. This is effectively done with the use of all the possible calculated features of an input fragment. The algorithmic analysis models of J48 and Random Forest are applied to our problem to see which produces the most efficient method of classification for predicting file types.*

Keywords- File Carving technique, prediction of file types, file fragment classification

I. INTRODUCTION

Forensic department investigators nowadays face a challenging work of retrieving back useful information together from a disk image which the perpetrators have deleted a long time back. Classification is an important data mining technique wherein huge data are classified into clusters and is used for retrieving relevant and useful information. Classification leads to clustering of dataset with rules. These rules are mined normally based on the features which are extracted from datasets. Unfortunately, the features may be relevant or irrelevant based on the users or upon the nature of the request made. This leads to some of the features being left out and might be required at some other level. This leaving out of features from datasets is usually referred to as the dimensionality curse or problem. Such occurrences are also known as data uncertainty.

File carving is a modern day technology used in computer forensics to extract data from a disk image or other storage devices without the help of the file system that originally created the file. It is a statistical method that recovers files at unallocated spaces without using any file information and is used to recover data type of the particular file fragments. A "file carver", is an application which makes use of the input fragments and write them back onto the disk.

The existing systems use the predictive accuracy of a predetermined learning algorithm to determine the goodness of the selected subsets, the accuracy of the learning algorithms is usually high. Most existing models depend on header and footer information specific to certain file types.
So in order to look for a picture file a header-footer based carver will look for fragments. Such fragments contains the pair of bytes near their start to determine that this fragment belongs to a JPEG file. Such models will fail for classifying the internal fragments. There is not much consistency between formats. The research is to provide a detailed view of the dataset used and to classify each and every algorithmic model and also to find the effective model among them which has the best outcome. We provide the result by predicting the most accurate file type for the given possible input fragment.

II. ABOUT THE DATA SET

The Data Set is driven from the govdocs1, a open source website that consists of a 1 million file dataset created by Garfinkel et al. This dataset is available in the public domain and freely accessible. The training thread that is taken into use is composed of 983 files with 28 different labels. This was primarily done with the goal in mind of creating a number of systems that would do the job of recognizing whole files, and then provide the general assumption that the fragments which are given are subjected to their respective file type.

III. FILE FRAGMENT FEATURES:

The file fragment features are those which are calculated for each training and testing input to represent each file to our learning models. The measure that is used for pursuing this calculation process is "Shannon entropy". The formula used for calculating the Shannon entropy H(X) of any input fragment is given by:

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$

Where P_i is the probability of any individual byte value in the fragment with respect to i.:

```
for (int i=0; i<frequency_array.length; i++)
{
        if (frequency_array[i] !=0)
        {
        Double probabilityOfByte = (double)frequency_array[i]/(double)fileContentLength;
        Double value = probabilityOfByte *(Math.log(probabilityOfByte)/ Math.log(2));
          entropy = entropy + value;
}
}
```

In general Shannon entropy does not directly refers to the "information per bit", but it corresponds to the file type. It was found from our research that the entropy value for all the text files were somewhere from 1.0 to 1.8, whereas for other file types including the picture file formats the entropy was so much close to that of 8.0. These range of values for each file type were considered as a major proportion for our research work. All the input values which are given to the system are normally treated as bytes instead of words. Actually the system is said to provide more accurate values for text files when compared with the values calculated for other file types.

IV. ALGORITHMIC MODELS USED:

Since the main goal of the system is to use data mining techniques for classification and prediction, we underwent a deep study on the list of classification models for our research work. In this paper we only take into consideration the use of J48, Least Absolute Deviation (LAD) and Random Forest algorithmic models. Naïve Bayes decision model is extensively used in the problem of judging documents as belonging to one category or the other with the number of words repeated in a document as the parameter. Naïve Bayes classifiers are scalable to a larger extent making them to require a number of parameters linear to the number of variables used in a classification problem.

Naïve Bayes models are also known as simple Bayes and independence Bayes. These names refer to the use of Bayes' theorem in the classifier's decision rule. The j48 algorithm that is used here is a genetic j48 modified algorithm. The j48 algorithm is used to generate decision trees that can be used for classification, and for this reason, J48 is often also referred to as a statistical classifier. The genetic j48 builds decision trees using the concept of information entropy. Here in this algorithm the attribute from the dataset with the highest normalized information gain is chosen to make the effective decision.

The second algorithmic model which is taken into account is the Least Absolute Deviation (LAD) model. This model is also called as Least Absolute Errors (LAE) or the Least Absolute Value (LAV) model. It is more similar to the famous least squares technique which attempts to find the function which closely approximates a set of data. In the case of (X, Y) data set, the approximation function uses a simple "trend line" in the form of two-dimensional Cartesian coordinates. This model also minimizes the Sum of Absolute Errors (SAE), which is the sum of the absolute values of the "vertical residuals" between points generated by the function and corresponding points of the data.

The Random Forest algorithm is an algorithm which builds multiple decision trees and merges them together to get a more efficient and stable prediction and classification. It provides a natural way of distinguishing clusters and to implement the requirements of clustering process as high within the similarity of clusters and their dissimilarities. Hence all these classification and prediction models will pave the way for a most accurate and statistical result set.

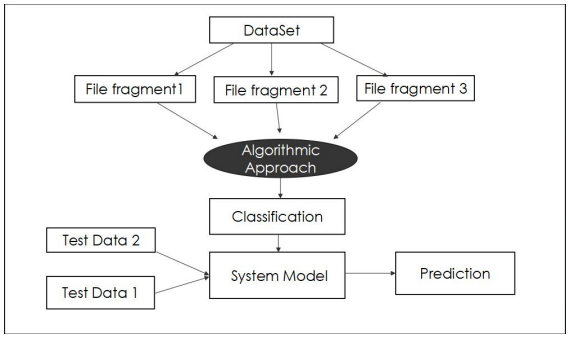

Figure 1. Design Model

V. DISCUSSION OF RESULTS

After training our system against a variety of different input fragments there are different result set values for different classification accuracies and there are values where the corresponding classification of the Decision tree is the best and there are values in which the best classifation and prediction are easily obtained. In most of the cases, the default values which are recommended are not always the optimal ones. Specifically in few cases the classification accuracies are too small. It has been found by considering the best classification features which are not that the Modified-Genetic J48 Algorithmic model is the most accurate one of all the algorithms.

Figure 2. Measuring Algorithm Accuracy

ALGORITHMS	ACCURACY
Modified genetic J48	95.8%
LAD	88%
Random Forest	92%

VI. CONCLUSION

This research work has the proposal which is primarily focused on the classification of the three decision tree algorithms based on the various parameters and also implementing the clustering approach to identify the file formats of distinct input fragments. The modified genetic J48 algorithm is found to be the most efficient one when compared with the other two models in terms of time, accuracy and features. So for scenario based training the corresponding algorithms may be used.

VII. FUTURE WORKS

There are a multiple number of future research directions to extend and improve our work. The process of extracting the features based on the reason stated will be investigated in the future work. One possible direction that this work might continue on is to improve the accuracy of similarity calculation between documents by employing different similarity calculation strategies. Although our current scheme is said to be proved more accurate than traditional methods, there are still rooms for improvement and more datasets may be accommodated in the future.

REFERENCES

[1] Pal, A., Memon, N., (2009). The Evolution of File Carving: The benefits and problems of forensics recovery. IEEE Signal Processing Magazine, page:59-71.
[2] B. Carrier File System Forensic Analysis MA Boston: Pearson Education Addison-Wesley Professional 2005.
[3] Suguna, Nandhini, "Literature review on data mining techniques", International journal of computer technology and applications, vol. 6, no. 4, pp. 583-585, 2015.

WEBSITES

[4] https://www.sans.org/reading-room/whitepapers/forensics/data-carving-concepts-32969
[5] https://www.youtube.com/playlist?list=PLea0WJq13cnCS4LLMeUuZmTxqsqlhwUoe
[6] https://www.cs.waikato.ac.nz/ml/weka/documentation.html