

An Improved Data Mining Tools Classifier for Digital Forensic Analysis

¹Ukwueze Frederick, N. and ²Okezie Christiana, C.

¹Department of Computer and Robotic Education, University of Nigeria, Nsukka.

²Department of Electronic and Computer Engineering Nnamdi Azikiwe University, Awka.

frederick.ukwueze@unn.edu.ng,
cc.okezie@unizik.edu.ng

Abstract- Due to prevalent issues of crime, enhanced by modern digital technologies, it has become necessary to find better ways of improving the methods used in digital investigations. Digital forensics addresses the specific need to be able to extract legally admissible evidence from computer systems, networks and other computing devices that can be used to successfully prosecute digital criminals. This process has become more time consuming and complex as the volumes of data requiring analysis continue to grow. In this study, an evaluation is made of data mining algorithms that can best establish the relevance of a digital device to a criminal case without the need for in-depth forensic examination. Thus, an improvement was affected on classification accuracies of two different data mining algorithms, namely K-Nearest Neighbours and Neural Networks. This was achieved by mitigating the effects of imbalance and overfitting in the dataset; and obtained results compared with a specified benchmark. The results show that when the pre-processing data techniques, of synthetic oversampling, training exemplars randomization, and better feature selection techniques were applied, the two selected algorithms outperformed some others.

Index Terms—data mining, digital forensics, classification algorithms, triage process.

1.0 Introduction

The enormous volume and complexity of data handled in the process of digital forensic investigation has become a prominent challenge in this field. Kryder's Law states that "the density of information on hard disks has been growing at an even faster rate, increasing by a factor of 1000 in 10.5 years, which corresponds to a doubling roughly every 13 months"[1]. Professionals involved in digital investigation are faced with greater challenges in the use of current tools to locate vital evidence within such massive volumes of data. As a result, it is difficult for digital investigators to conduct the forensic analysis process in a cost effective and timely manner. It is also known that the quantity of data which can satisfactorily incriminate a suspect is usually small in relation to the entire mass of data divulged from seized digital storage devices. This not only will delay the process of investigation, but will also interfere with the suspects privacy rights. It is expedient in this situation to have a way of setting priorities for the analysis to dramatically cut the response time. Part of the solution would be to evaluate improvement to data mining algorithms to see which can best

establish the relevance of a digital device to a criminal case without the need for in-depth forensic examination.

The digital forensics triage process begins with initial acquisition and imaging of any media as evidence. Triage results are comprised of summary information about a collection of media being investigated and are used to prioritize media for further investigation. Reports generated for triage analysis should allow for immediate exoneration of innocent suspects by isolating and removing of unnecessary media.

2.0 Literature Review

In this area, much work has been done seeking to create offence related templates by categorizing offences according to common features. Gomez in [2] already identified the need to train Machine Learning classifiers on the basis of offence-related features which is very similar to the idea of searching devices following an offence-specific template. Hong in [3], has recently addressed the problem of implementing a comprehensive Digital Triage methodology capable of meeting Korean legal system's demand for suspect's privacy protection during a criminal investigation. The cited

author claims that the seizure of a storage drive and consequent creation of forensic images have led to gross violation of suspects privacy rights as the incriminatory information that may be found represent only a minimal part of it. The work of [4] in finding a solution to multi-user data ascription problem, showed, after tests, that C4.5 was preferable to KNN as a classifier in such criminal case. The choice of the two algorithms was taken because of the ease with which C4.5, particularly, could be explained to non technical legal persons in court rooms. In the field of bulk data analysis, as an alternative solution to data extraction and analysis with digital forensic tools currently in use, [4] developed bulk extractor, a carving and feature extraction tool which allows a raid triage process [5]. In the works of [6], the authors evaluated the performance of Bayesian Networks, Decision Tree, Locally Weighted Learning, and Support Vector Machines. On average, it was observed that the Bayes Network algorithm resulted in best performance according to selected relevant offence-related features.

The conflict of the obtained results from earlier studies on the same subject informed a closer examination of the possible effects of over-fitting and imbalanced dataset problems on the performance of classification algorithms, and to see if mitigating these effects can bring any harmony in the results.

2.1 Choice of Algorithms

Here, we present an analytical description of four selected classification algorithms which would be compared to each other after employing techniques that can mitigate the two common problems of over-fitting and dataset imbalance. The algorithms were selected based on conflicting performance results obtained by different authors viz: in the earlier work of [7], the performance of four classification algorithms to a set of data for predicting students' performance across 4 technical trade areas was evaluated and the results were comparably accurate. But decision tree proved to be the most accurate classifier. In [4], as was noted earlier, the author tested and compared two Machine Learning algorithms for classification tasks, where decision tree (C4.5) also was reported to have produced better results; However, In [5], the author evaluated the performance of four algorithms, and observed that the Bayes Network algorithm resulted in best performance according to selected offence-related

features; while, the authors in [8] found that Multilayer Perceptron was a better algorithm across five different data sets. Concerning the instance based learning algorithm, (KNN), it was also noted in the previous work of [7] that building the model is cheap, and also that given our type of data set, where the number of attributes are fairly small (less than 20), and also a lot of training data to use, KNN is a preferred algorithm. Thus, we choose to apply improvement measures to K-Nearest Neighbour and Multi-Layer Perceptron along with other algorithms and evaluate their performance against a baseline result obtained in [7] for the selected algorithms. The following is analytical description of the selected algorithms.

A. Decision Tree Algorithms

Tree classification algorithms; appear to be most useful in understanding critical data distribution, [9]. A decision tree algorithm typically optimizes some information theoretic measure, like information gain, on a training set. The generation of the tree is done recursively by splitting the data set on the independent variables. Each possible split is evaluated by calculating the *purity gain* it would result in if it was used to divide the data set D into the new subsets $S = \{D_1, D_2, \dots, D_n\}$. The purity gain is the difference in purity between the original data set and the subsets as defined in equation 1 below, where $P(D_i)$ is the proportion of D that is placed in D_i . The split resulting in the highest purity gain is selected, and the procedure is then repeated recursively for each subset in this split.

$$\text{gain}(D, S) = \text{purity}(D) \sum_{i=1}^n P(D_i) * \text{purity}(D_i) \quad (1)$$

One of the algorithms that implement this process C4.5 seeks to optimize the measure of data disorder resulting from the splits. This measure is the entropy E . The binary entropy function, denoted by $H_2(x)$ or $H_b(p)$, is defined by [21] as "the entropy of a Bernoulli process with probability of success $P(x=1) = p$. Mathematically, the Bernoulli trial is modeled as a random variable x that can take on only two values: 0 and 1. The event $x = 1$ is considered a success and the event $x = 0$ is considered a failure"

If $P(x = 1)$, then $P(x = 0) = 1 - P$ and the entropy of x is given by

$$H_2(x) = x \log \frac{1}{x} + (1 - x) \log \frac{1}{1-x} \quad (2)$$

J48 is a Java implementation of the C4.5 algorithms in the WEKA data mining tool.

B. K-Nearest Neighbour (KNN)

KNN is a supervised learning algorithm, with k training samples which generate the rules by themselves without input from any external source. Test samples are categorized, and a test sample's allocation to a category is determined by the probability level of that category measured by its nearness to the test sample, in relation to other samples in the neighbourhood. The classification process of a typical sample X according to [10] in [11] is as follows:

- Suppose there are j training categories $C_1, C_2, \dots, C_j$ and the sum of the training samples is N after feature reduction, they become m -dimension feature vector. Make sample X to be the same feature vector of the form $(X_1, X_2, \dots, X_m)$, as all training samples.
- Calculate the similarities between all training samples and X . Taking the i th sample d_i ($d_{i1}, d_{i2}, \dots, d_{im}$) as an example, the similarity $SIM(X, d_i)$ is as following:

$$SIM(X, d_i) = \frac{\sum_{j=1}^m x_j d_{ij}}{\sqrt{(\sum_{j=1}^m x_j^2)} \sqrt{(\sum_{j=1}^m d_{ij}^2)}} \quad (3)$$

- Choose k samples which are larger from N similarities of $SIM(X, d_i)$, ($i = 1, 2, 3, \dots, N$), and treat

them as a KNN collection of X . Then, calculate the probability of X belonging to each category

respectively with the following formula:

$$P(X, C_j) = \sum_d SIM(X, d_j) \cdot y(d_j, C_j) \quad (4)$$

Where $y(d_i, C_j)$ is a category attribute function, which satisfied

$$y(d_i, C_j) = \begin{cases} 1, & d_i \in C_j \\ 0, & d_i \notin C_j \end{cases} \quad (5)$$

Judge sample X to be the category which has the largest $P(X, C_j)$.

The degree of the nearness of training samples to an unknown sample is defined in terms of various distance measures, one of which is the Euclidean

distance. This distance between two points, $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$ according to [12] is

$$d(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (6)$$

The unknown sample is assigned the most common class among its k nearest neighbours.

C. Neural Network Classifiers

Multilayer Perceptron:

A Multi-Layer Perceptron (MLP) according to [13] is a feed-forward neural network with one or more layers between input and output layer. Basically there are three layers: input layer, hidden layer and output layer. Hidden layer may be more than one. Each neuron (node) in each layer is connected to every neuron (node) in the adjacent layers. The training or testing vectors are connected to the input layer, and further processed by the hidden and output layers. The idea behind a Neural Network (NN) is that a very complex behaviour can arise from relatively simple units, which are referred to as artificial neurons, acting in concert. Used in legal domains such as money laundering and credit card fraud detection, a network of artificial neurons is designed to mimic the human ability to learn from experience [14].

A Multilayer Feed-Forward Neural Network

According to [15] "a multilayer feed-forward neural network is the type of neural network on which the back-propagation algorithm performs. The inputs correspond to the attributes measured for each training sample. The inputs are in the input layer.

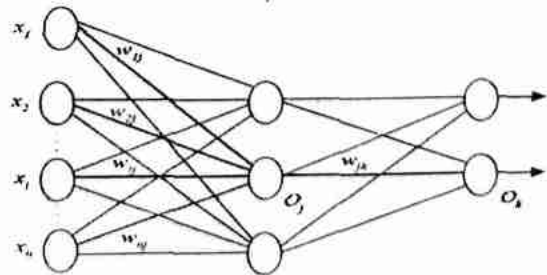


Fig 1: A multilayer feed-forward neural network

The weighted outputs of these units are fed simultaneously to a hidden layer. The hidden layer's weighted outputs can be input to another hidden

layer etc" Usually only one hidden layer is used (see Fig.1) above.

The output layer, which emits the network's prediction for given samples, is made from last hidden layer output. The network is feed-forward, because none of the weights cycles back to an input unit or to an output unit of a previous layer.

Back-propagation

Back-propagation learns by iteratively processing a set of training samples. It compares the network's prediction for each sample with the actual known class label. The weights are modified to minimize the mean squared error between the network's prediction and the actual class for each training sample. These modifications are made from the output layer through each hidden layer down to the first hidden layer. The algorithm will be described in four steps:

1. Initialize the weights:

The weights and the biases are initialized to small random numbers in interval -1.0 to 1.0 .

2. Propagate the inputs forward

The net input and output of each unit in the hidden and output layers are computed. The training sample is fed to the input layer of the network. For unit j in the input layer its output is equal to its input for input unit j , $O_j = I_j$. The net input to each unit in the hidden and output layers is computed as a linear combination of its inputs. To compute the net input to the unit, each input connected to the unit is multiplied by its corresponding weight, and summed: [16],

$$I_j = \sum_i w_{ij} O_i + \theta_j \quad (7)$$

Where w_{ij} is the weight of the connection from unit i in the previous layer to unit j O_i is the output of unit i from the previous layer. θ_j is the bias of the unit. Each unit in the hidden and output layers takes its net input and then applies an activation function to it. The function symbolizes the activation of the neuron represented by the unit.

$$O_j = \frac{1}{1 + e^{-I_j}} \quad (8)$$

3. Back-propagate the error

The error is propagated backwards by updating the weights and biases to reflect the error of the network's prediction. For a unit j in the output layer:

$$Err_j = O_j(i - O_j)(T_j - O_j) \quad (9)$$

where O_j is the actual output of unit j , T_j is the true output and $O_j(i - O_j)$ is the derivative of the logistic function. The error of a hidden layer unit j is:

$$Err_j = O_j(i - O_j) \sum_k Err_k w_{jk}, \quad (10)$$

where w_{jk} is the weight of the connection from unit j to a unit k in the next higher layer, Err_k is the error of unit k . The weights and biases are updated to reflect the propagated errors:

$$\Delta W_{ij} = (i) Err_j O_i$$

$$W_{ij} = W_{ij} + \Delta W_{ij} \quad (11)$$

where ΔW_{ij} is the change in weight w_{ij} and i is the learning rate. A constant i has typically a value between 0.0 and 1.0 . A rule of thumb is to set the learning rate to i/t , where t is the number of iterations through the training set so far. Biases are updated by following equations:

$$\Delta O_j = (i) Err_j$$

$$\theta_j = \theta_j + \Delta \theta_j \quad (12)$$

where $\Delta \theta_j$ is the change in bias θ_j .

4. Terminating condition

Training stops when:

- all Δw_{ij} in the previous epoch were so small as to be below some specified threshold, or
- the percentage of samples misclassified in the previous epoch is below some threshold, or
- a pre-specified number of epochs has expired.

D. Bayesian Classification

Bayesian classifiers are statistical classifiers that can predict class membership probabilities, such as the probability that some sample belongs to a particular class. This classification is based on Bayes theorem; it has high accuracy and speed when applied to large databases.

Bayes Theorem

Let X be a data sample whose class label is unknown and H some hypothesis, that the X belongs to class C . The probability that H holds given the X is $P(H|X)$. $P(H|X)$ is the posterior probability of H

conditioned on X . $P(H)$ is the prior probability of H . $P(X|H)$ is the posterior probability of X conditioned on H . $P(X)$ is the prior probability of X . Bayes theorem calculates the posterior probability $P(H|X)$ from $P(H)$, $P(X)$ and $P(X|H)$ and is expressed by [16] as:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (13)$$

Naive Bayesian Classification

Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This is called class conditional independence. Simplifying the computations is the main reason why this technique is used. The naive Bayesian classification algorithm work according to [16] should be divided into five steps as follows:

1. Each data sample is represented by an n -dimensional feature vector $X = (x_1, x_2, \dots, x_n)$ where

n measurements are made on sample from n attributes.

2. Suppose that there are m classes $C_1, C_2, \dots, C_m$. Given an unknown data sample X , the classifier

will predict that X belongs to the class having the highest posterior probability, conditioned on X . The naive Bayesian classifier assigns an unknown sample X to the class C_i iff

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m, j \neq i$$

Then we maximize $P(C_i|X)$:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (14)$$

3. As $P(X)$ is constant for all classes, only $P(X|C_i)P(C_i)$ need be maximized. The class prior probabilities may be estimated by $P(C_i) = \frac{s_i}{s}$, where s_i is the number of training samples of class C_i and s is the total number of training samples.

4. Because of computation reduction, the naive assumption of class condition independence is made. There are no dependence relationships among the attributes.

$$P(C_i|X) = \prod_{k=1}^n P(x_k|C_i) \quad (15)$$

The probabilities $P(x_1|C_i), P(x_2|C_i), \dots, P(x_n|C_i)$ are evaluated from the training set:

• If A_k is categorical, then:

$$P(x_k|C_i) = \frac{s_{ik}}{s_i} \quad (15)$$

where s_{ik} is the number of training samples of class C_i having the value x_k for A_k and s_i is the number of training samples belonging to C_i .

• If A_k is continuous-valued then the attribute is assumed to have a Gaussian distribution:

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi}\sigma_{C_i}} e^{-\frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}^2}} \quad (17)$$

where $g(x_k, \mu_{C_i}, \sigma_{C_i})$ is the Gaussian density function for attribute A_k while μ_{C_i} and σ_{C_i} are the mean and standard deviations.

5. $P(X|C_i)P(C_i)$ is evaluated for each class C_i . Sample X is then assigned to the class C_i iff

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \text{ for } 1 \leq j \leq m, j \neq i.$$

2.2 Handling Dataset Problems

The larger class advantage is a common problem during mining of skewed datasets. Models generated tend to be more "predictive for the larger class, while delivering performance which is poorly predictive for the minority classes [17]. The case arises naturally as classifiers will produce results that characterize the entire available data under processing. Thus, a less relevant class may obtain an undue recognition in the process. There are 5 types of problem that occur when learning from skewed datasets [17]: Incorrect evaluation measures, absolute scarcity of data, the number of items that belong to the minority class being small when compared to the total number of samples; this can make it difficult to observe patterns within the minority class and data fragmentation, this can be a problem as most classifiers utilize a divide and conquer approach. It causes a continual division of the learning space and results in situation where patterns that are found in the data as a whole cannot be found in the resulting partitions formed by this divide and conquer strategy. There may be also inappropriate inductive bias and noise. To overcome these problems, [18] also collated a number of approaches to overcome these problems. These include, using data mining algorithms that can cope with the class imbalance, use of more appropriate evaluation metrics (F-measure, G-measure, AUC values), (Metacost procedure), over sampling, cost sensitive learning, boosting etc. These methods according to the author can be divided into two

general approaches: These are data methods and algorithmic methods. For the algorithmic method, cost sensitive learning is popular. The data centric solutions generally involve re-sampling i.e., use of over/under sampling methods along with methods for creating artificial samples. On the other hand; the limited number of training samples has a way of causing the overfitting problem during the process of validation. Following are some strategies that mitigate dataset problems

E. Data Pre-processing

In feature selection, the main dataset is converted to a new dataset and at the same time, forcing a reduction of the dimensionality. This is done by extracting the most relevant features. Conversion and dimensionality reduction will achieve two results; one is that the data set pattern will become clearer and more understandable. Two, more reliable classification becomes tenable by concentrating on the most important data which keeps the maximum properties of the main data.

A popular way to achieve this is to either over-sample the minority class or under-sample the majority class. One popular versions of over-sampling, SMOTE, (Synthetic Minority over-sampling Technique) is achieved by creating new synthetic examples.

Synthetic examples are generated by operating in the "feature space" rather than the "data space" [19]. The synthetic examples cause the classifier to create larger and less specific decision regions rather than smaller and more specific regions. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours. Depending upon the amount of over-sampling required, samples from the k-nearest neighbours are randomly chosen. This approach electively forces the decision region of the minority class to become more general.

F. Attribute selection

In the attribute selection process, there are two main approaches; the wrapper, and filter approach. The wrapper approaches the actual data mining algorithm in its search for the attribute subsets while in the filter approach, undesirable attributes are filtered out of the data before classification begins.

Filter Method

The filter attribute selection method is independent of the classification algorithm. Filter method is further categorized into two types: attribute evaluation algorithms and subset evaluation algorithms

The algorithms are categorized based on whether they rate the relevance of individual features or feature subsets. Attribute evaluation algorithms rank the features individually and assign a weight to each feature according to each feature's degree of relevance to the target feature. The attribute evaluation methods are likely to yield subsets with redundant features since these methods do not measure the correlation between features. Subset evaluation methods, in contrast, select feature subsets and rank them based on certain evaluation criteria and hence are more efficient in removing redundant features. The main disadvantage of the filter method is it ignores the dependencies among the features and treats the features individually.

Wrapper Method

The wrapper method is slower and more expensive than the filter method. The interaction between the features subsets in maintaining the dependencies between the features are the main advantages of wrapper method. Wrapper methods are better in defining optimal features rather than simply relevant features. They do this by allowing for the specific biases and heuristics of the learning algorithm and the training set. This method uses the backward elimination to remove the insignificant features from the subset. It needs some predefined learning algorithm to identify the relevant feature. It has interaction with classification algorithm. The over fitting of feature is avoided using the cross validation.

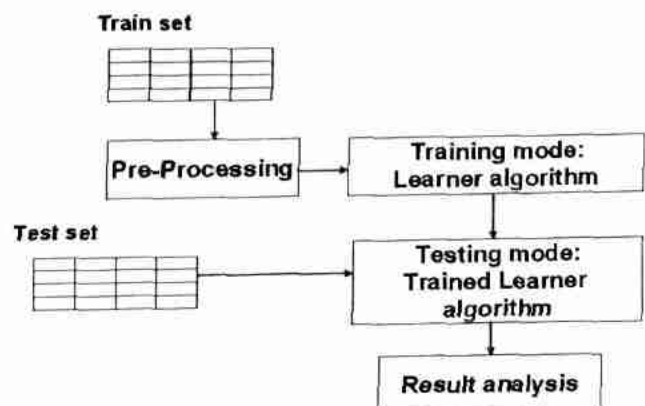


Figure 2: Model for Machine Learning System

3.0 Methodology

A Model for this Machine Learning processes presented as captured in Fig. 2 include,

Training set creation.

Exemplars of known class (through previous observations) are collected and used to populate the training input matrix. This constitutes the input to selected classifiers.

Pre-processing

The pre-processing module essentially aims at the feature extraction/selection. It finds the appropriate features for representing the input patterns by segmenting the pattern of interest from the background, removing noise, normalizing the pattern, and any other operation which will contribute in defining a compact representation of the pattern.

Training Mode

Selected algorithms process the training input matrix which has two major categories: multiclass and binary categorization methods. In multiclass categorization, instances are classified in accordance with the closest offence category in the training set. Binary categorization is where a classifier is trained per each offence category. This will be adopted in our analysis.

Learning accuracy evaluation.

The model is validated using a Confusion Matrix, with the following performance indicators: Precision, Recall, and F-measure.

Test set creation.

After training the classifiers, and learning accuracy has been determined, the model if the accuracy is satisfactory, begins to classify new test instances. This has been populated with data extracted from the digital devices provided from law enforcement agencies, either seized directly from suspects or collected from the crime scene using a search a warrant. A test instance is created per each device and the test input matrix is populated as a collection of all the available test instances.

Test set classification and performance evaluation.

Selected learners are fed with the created input matrix and test instances are classified accordingly. Testing accuracy is computed according to the three selected performance indicators.

3.1 Experiments

The following experiments were used to explore how setting the three parameters of the dataset, namely order of training instances, number of training instances, and number of attributes, in validation folds, would improve classification accuracy and solve the two inherent issues of the dataset.

WEKA, an open source machine learning suite was used in the analysis. In WEKA, datasets have to be formatted with inbuilt tools to the ARFF format. Experiments are then carried out as many times as is needful to produce a consistent output. The average accuracy is recorded.

WEKA has inbuilt data pre-processing facilities four of which was used in this study. The first seeks to modify the distribution of Exemplars, by changing the order of training instances within the validation folds. This processing filter is called RANDOMIZE in WEKA explorer. It is used to correct the effects of over-fitting during validation and improve learning accuracy. The next analysis used SMOTE (Synthetic Minority Oversampling Technique), which was employed to beef up the number of minority training samples thereby solving the problem of data imbalance. Finally, WRAPPER and FILTER methods were used for attribute reduction, by removing redundant or useless features in the dataset and also achieve automatic attribute selection to verify possible improvement in learning accuracy.

Model Performance Measures (The Confusion Matrix)

In an imbalanced classification problem, the minority class is often referred to as the positive class and the other one as the negative class. Samples can be categorized into four groups after a classification process, which is denoted in the confusion matrix Table 1, presented below.

Table 1: The Confusion Matrix

[55] [56] Actual Class	[57] Predicted Class	
	[58] Positive	[59] Negative

[60] Positive	[61] True Positive (TP)	[62] False Negative (FN)
[63] Negative	[64] False Positive (FP)	[65] True Negative (TN)

Samples in the positive class are usually regarded high identification importance; therefore, we only evaluate our approach based on the performance of the positive class. In general, there are two well-accepted measures [20] True Positive Rate (a measure of Precision) and Positive Predictive Value, (Recall, a measure of sensitivity). True Positive Rate (TPR) is defined [19] as:

$$\text{Precision} = \frac{TP}{TP + FN} \quad (18)$$

Positive Predictive Value is defined as

$$\text{Recall} = \frac{TP}{TP + FP} \quad (19)$$

To balance these two measures, F-measure is suggested in Wan *et al.* (2014) which is defined as

$$F_measure = \frac{(1 + \beta_2) \times TPR \times PPV}{\beta_2 \times TPR + PPV} \quad (20)$$

Where corresponds to the relative importance of TPR versus PPV and it is typically set to 1. The F-measure incorporates TPR and PPV into a single number. It basically represents a harmonic mean between them. It follows that the F-measure is high when both TPR and PPV are high. This indicates that F-measure is able to evaluate the performance of a learning algorithm on the class of our interest.

We analyzed thirty different digital disks, to make a case study concerning copyright infringement offence in the Nigerian legal system environment. Disk images were created from the storage disks which were used as the primary dataset. The disk images were processed for file, system information using extraction scripts based on *Sleuth Kit* and the *Bulk Extractor*. The extracted features were analyzed to obtain the number of its appearance per training instance. The data was then classified in WEKA. In this study, we adopt the results obtained in our earlier work [7] to serve as the reference baseline. Those results were first, reprocessed to obtain values consistent with our chosen performance measures. We then compared it with results of current analysis. Classifiers' learning accuracy was computed according to the selected performance measures i.e., Precision, Recall and F-measure. In the training input matrix, we had 17 attributes, and 30 available training exemplars (digital storage drives).

4.0 Results

In the work used as the reference baseline, we evaluated the performance of K-Nearest Neighbour, Decision Tree, Neural Networks and Bayesian Networks, on the training input matrix, using 10-fold cross validation. In this regards, Table 2 illustrates related validation output representing the reference baseline against which the results of the present study have been compared. Subsequent results follow.

Table 2: Reference Baseline Results

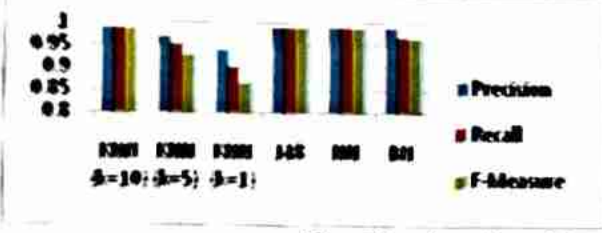
Classification Algorithm	KNN			J48	NN	BN
	k=1 0	k=5	k=1			
Precision	0.85	0.8	0.78	0.88	0.79	0.78
Recall	0.82	0.76	0.74	0.83	0.74	0.75
F-Measure	0.79	0.78	0.63	0.82	0.72	0.74

The Base classifiers were duly pre-processed before the tests and 10-fold cross validation was employed in testing classifier accuracy.

A. Effect of randomizing the Exemplars

To deal with the probable effect of overfitting, Exemplars have to be randomized in advance before testing. An estimate of the training accuracy showed remarkable improvement of the base classifiers, as shown in Table 3.

Table 3: Accuracy performance of randomized base classifiers

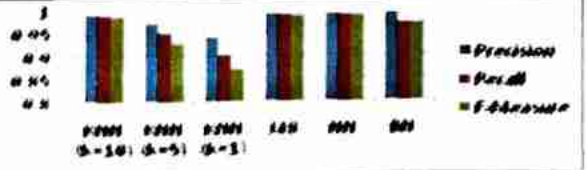
 Classification Algorithm	KNN					
	k=10	k=5	k=1	J48	NN	BN
Precision	0.88	0.83	0.81	0.9	0.81	0.8
Recall	0.87	0.69	0.76	0.85	0.78	0.78
F-Measure	0.84	0.82	0.69	0.84	0.79	0.76

It is clear from the result that the skewed distribution of Exemplars adjudged to be culpable (4) and those thought could be exonerated (11), affected the accuracy. The process of randomization balanced the training set and improved the accuracy of the base classifiers.

B. Effect of Oversampling the Exemplars.

This was done by setting the minority and majority class elements to 15 each. I.e., the minority class was oversampled from 11 to 16. This in addition to randomization (necessary to avoid the possible reverse effect of oversampling during validation.) brought a significant improvement on classification accuracy as observable in Table 4.

Table 4: Performance of classifiers under combined effect of randomization and oversampling

 Classification Algorithm	KNN					
	k=10	k=5	k=1	J48	NN	BN
Precision	0.98	0.9	0.87	0.98	0.92	0.85
Recall	0.98	0.88	0.81	0.96	0.90	0.86
F-Measure	0.97	0.87	0.80	0.93	0.88	0.84

As would be expected, KNN improved better with higher values of k , and NN.

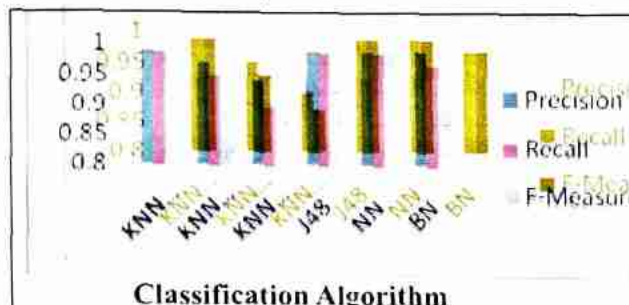
C. Effect of Attribute number reduction and automatic feature selection.

In this test we still need to keep the minority class exemplars oversampled, as in previous test, as well as randomizing training exemplars; then we reduce the number of attributes by reducing

the feature space dimensionality (*Wrapping*). We could do with a (*filtered*) feature subset of 6 instead of 30.

The combined effect of *wrapper* and *filter* methods applied brought maximum improvement to all the base classifiers as shown in Table 5.

Table 5: Improved performance of Base classifiers through wrapper and filter methods



Classification Algorithm	KNN			J48	NN	BN
	k=1	k=5	k=10			
Precision	0.99	0.97	0.94	0.99	0.99	0.99
Recall	0.99	0.95	0.90	0.99	0.99	0.97
F-Measure	0.99	0.93	0.87	0.99	0.99	0.97

5.0 Conclusion

The results achieved in these experiment show that all the selected learners, duly pre-processed, performed better than the baseline when the data is balanced. KNN (with $k = 3: 5$ equivalent to the baseline achievement of Bayesian Networks, reducing the feature space dimensionality to one dimension only resulted in the best achievable classification accuracy, which is equivalent to the Bayesian Network baseline. A feature subset consisting of 6 out of 30 attributes has been selected, scoring an information gain above 0.9, and resulting in better classification accuracy with regards to the baseline. We conclude, therefore, that Neural Network classifiers can deliver a better performance than Bayesian and Tree based classifiers if appropriate measures are undertaken to mitigate the effect of data over-fitting and imbalance. It can achieve maximum obtainable classification accuracies than Bayes Net and other algorithms.

References

- [1] Walter C. —Kryders Law insights", in *Scientific American Inc August 2005*, pp32-33.
- [2] Gomez, L. S. M. "Triage in-lab: case backlog reduction with forensic digital profiling" in *Simposio Argentino de Informtica y Derecho*, 2012.
- [3] Hong, L., Yu, H., Lee, S., and Lee, K. "A new triage model conforming to the needs of selective search and seizure of electronic evidence" in *Digital Investigation*, 10(2), 175–192, 2013.
- [4] Garfinkel, S. L. "Digital forensics research: The next 10years" in *Digital Investigation*, 7, S64–S73, 2010
- [5] Garfinkel, S. L. "Digital media triage with bulk analysis and bulk extractor" in *Computers & Security*, 32, 56–72, 2013
- [6] Marturana, F., Berte', R., Me, G., and Tacconi, S. "Triage-based automated analysis of evidence in court cases of copyright infringement" in *IEEE International Workshop on Security and Forensics in Communication Systems*, 2012.

- [7] Ukwueze F. N. and Okezie C. C. "Evaluation of Data Mining Classification Algorithms for Predicting Students Performance in Technical Trades" in International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 5 Issues 8 Aug 2016, Page No. 17593-17601
- [8] Suman, Rohit Arora "Comparative Analysis of Classification Algorithms on Different Datasets using WEKA" in International Journal of Computer Applications (0975 – 8887) Volume 54– No.13, September 2012
- [9] Nadali, A; Kakhky, E.N.; Nosratabadi, H.E., "Evaluating the success level of data mining projects based on CRISP-DM methodology by a Fuzzy expert system" in (ICECT), 2011 3rd International Conference on Electronics Computer Technology, vol.6, no., pp.161, 165, 8-10 April 2011.
- [10] Lihua Y., QiD. and Yanjun, G. "Study on KNN Text Categorization Algorithm" in Micro Computer Information, 21, pp. 269-271, 2006.
- [11] Suguma N, and Thanushkodi K, "An Improved K-Nearest Neighbor Classification Using Genetic Algorithm" in *International Journal of Computer Science Issues* 7, issue 4, No2, July 2010 ISSN(online) 1694-0784 pp 18-21
- [12] A. S Galathiya, A. P. Ganatra and C. K. Bhensdadia "Improved Decision Tree Induction Algorithm with feature selection, cross validation, model complexity and reduced error pruning" in *International Journal of Computer Science and Information technologies*, Vol. 3(2), 2012, 3427-3431.
- [13] V. Vaithiyanathan 1, K. Rajeswari2 , Kapil Tajane3 , Rahul Pitale3 "Comparison Of Different Classification Techniques Using Different Datasets" in International Journal of Advances in Engineering & Technology, May 2013, Vol. 6, Issue 2, pp. 764-768
- [14] Linoff, G. S. and Berry, M. J. Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. JohnWiley & Sons, 3rd edition, 2011
- [15] Wei Wu, Jiuping Xu. "Fundamental analysis of stock price by artificial neural networks model based on rough set theory. in *World Journal of Modeling and Simulation* Vol. 2 (2006) No. 1, pp 36-44
- [16] Kunc Michael —Selected Topics of Information Systems: Data Classification! Semester Project, Bruno University of Technology Faculty Of Information Technology 2007.
- [17] Peter Brennan "A Comprehensive Survey of Methods for Overcoming the Class Imbalance Problem In Fraud Detection! Unpublished M.Sc Project Report. Institute Of Technology Blanchardstown Dublin, Ireland June 2012.
- [18] Weiss, G. M. "Mining with rarity: a unifying framework" SIGKDD Explorations and Newsletters, 6, 7–19, 2004
- [19] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. "SMOTE: Synthetic minority over-sampling technique", *Journal of Artificial intelligence*, 16, 321–357, 2002.
- [20] Xiang Wan, Jiming Liu, William K Cheung and Tiejun Tong "Learning To Improve Medical Decision Making From Imbalanced Data Without A Priori Cost! Bmc Medical Informatics and Decision Making 2014
- [21] —Binary Entropy Function!, Wikipedia (2016), http://en.wikipedia.org/wiki/Binary_function Accessed 4th April, 2017.

Authors' Profiles



Ukwueze Frederick Nwabueze, is a Lecturer and former Coordinator, Computer Education at University of Nigeria, Nsukka. He holds a bachelors' degree in Electronic Engineering and a Masters degree in Computer Science of the University of Nigeria. He joined the academic faculty in 2007 where he is actively engaged in teaching of undergraduate and postgraduate courses. Prior to his University career, Engr. Ukwueze had worked in various companies involved in computer, communication and project management consultancy. His research interests include Data Mining, Machine Learning and Digital Forensics.



Okezie Christiana C. is a Professor and current Head, Department of Electronic and computer Engineering Nnamdi Azikiwe University Awka Nigeria. She holds a Bachelor of Engineering (B.Eng) degree in Electronic/Electronic Engineering and also a masters (M Eng) and doctorate (PhD) degrees in Computer Engineering. Starting a career as a Maintenance Engineer with Shell Petroleum Development Company (SPDC), she joined the academic faculty in 1998 where she has been engaged in teaching and research, having published over 70 academic journal articles and several conference papers, textbooks and monographs. She has served as editorial board member of many international journals. She specializes in the areas of Computer Engineering, Microprocessors, Data Communication Engineering and Database systems.