Machine Learning Algorithms for Document Classification: Comparative Analysis

Faizur Rashid[1]
Department of Computer Science
Haramaya University
Almaya, Ethiopia

Abdulkadir H. Aden[3]
Department of Computer Science
Bule Hora University
Ethiopia

Suleiman M. A. Gargaare[2]
Department of Computer Science
University of Hargeisa
Somaliland

Afendi Abdi[4]
Department of Software Engineering
Haramaya University
Almaya, Ethiopia

***Abstract*—Automated document classification is the machine learning fundamental that refers to assigning automatic categories among scanned images of the documents. It reached the state-of-art stage but it needs to verify the performance and efficiency of the algorithm by comparing. The objective was to get the most efficient classification algorithms according to the usage of the fundamentals of science. Experimental methods were used by collecting data from a sum of 1080 students and researchers from Ethiopian universities and a meta-data set of Banknotes, Crowdsourced Mapping, and VxHeaven provided by UC Irvine. 25% of the respondents felt that KNN is better than the other models. The overall analysis of performance accuracies through various parameters namely accuracy percentage of 99.85%, the precision performance of 0.996, recall ratio of 100%, F-Score 0.997, classification time, and running time of KNN, SVM, Perceptron and Gaussian NB was observed. KNN performed better than the other classification algorithms with a fewer error rate of 0.0002 including the efficiency of the least classification time and running time with ~413 and 3.6978 microseconds consecutively. It is concluded by looking at all the parameters that KNN classifiers have been recognized as the best algorithm.**

***Keywords*—Document classification; machine learning algorithms; text classification; analysis**

I. Introduction

Document classification is a vital research area or topic since the establishment of digital documents used widely [1]. This is one of the major parts of the manual effort. More tech companies outsource the job and business processes people have to sort the documents of the packages manually. There is the availability of the hundreds of documents types. Nowadays, text classification is an important task because of the very large amount of text documents required to deal with day-to-day activities. In general, document classification can be classified as topic-based document classification and document genre-based classification. Topic-based document categorization can be classified documents according to their topics [2]. Also, texts can be written in many different genres, for example, academic papers, advertisement updates, political news, and movie reviews. Genre referred to the way a text is made, the way it was modified, the identification of language used, and the type of listeners to whom it is addressed. Existing studies on genre classification found that task differs from the categorization of topic-based [3]. Commonly, most data based on genre classification were collected from the newspaper, web, noticeboards, and live broadcasts.

The classification is an information retrieval from the metadata, manually classified, or via an automatic classifier retrieving information from the content. As manually classifying documents can be a time-consuming and inconsistent task. It is usually not beneficial on a larger scale. Instead, automatic Document Classification is suggested to solve the categorization of retrieved information, because of the automatic process for larger systems [4].

Although some degree of automation is achieved that helps to search through keywords and expressions the accuracy and efficiency of such solutions are questionable and not satisfactory. An approval of efficiency matters to show the user's satisfaction that needs assessing and analyzing machine learning algorithms. It increases the satisfaction of automated document classification. The document classification contains many concepts as Fig. 1 shows.

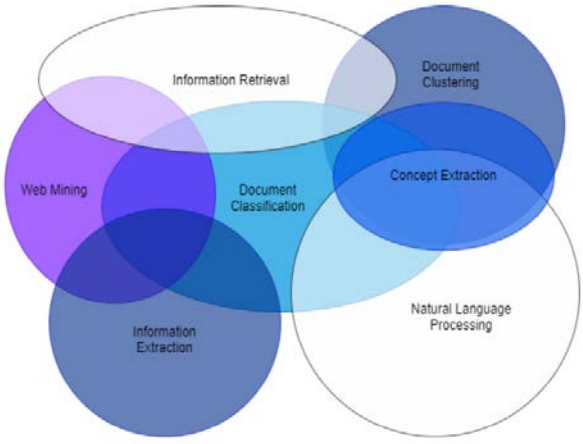

Fig. 1. By [5] Venn Diagram of the Text Mining Area.

This study is organized as follows, In Section 1, Introduction is presented. Section 2 presents different types of Classification Algorithms. Section 3 describes the related works. Section 4 introduces the Methodology. Section 5 shows the Results and Discussions of the study. Conclusions and recommendations are discussed in Section 6 and Reference is cited in Section 7.

II. CLASSIFCATION OF ALGORITHM

A. Logistic Regression

According to [6], the binary outcomes either something happens or nothing happens, yes or no, pass or fail, living or dead are calculated using logistic regression. In addition, two factors were described: independent and dependent variables, which were examined to determine the binary outcome based on whether the outcomes were numeric or categorical. The independent variables might be categorical or numeric, but the dependent variable was categorical all the time and stated in equation (1):

$$P(A=1|B) \text{ or } P(A=0|B) \quad (1)$$

Whereas, A and B calculate the probability of the dependent variable and independent variable consequently.

Positive or negative connotation {0, 1} word or tree, grass and flower which was common object contained in a photo calculated by probability of each object between 0 and 1 R. Wolf, 2021.

B. K-Nearest Neighbor (KNN)

The objects in KNN were categorized as [7], defined, and filled by selecting numerous labeled training examples with the shortest distance from each other. The k-nearest neighbor classification method stood out with its simplicity and commonly used techniques for text categorization, Even if classifying items took longer when a large number of training samples were provided. Even with multi-categorized documents, this strategy works well for regulating categorization jobs. KNN should be chosen manually, and some of them can be determined by calculating the distance between each test object and all of the training samples.

C. Decision Trees

A decision tree is a tree in which internal nodes are labeled with terms, according to [8]. The branch was labeled by numerical data, while the leaf was labeled by categories. The "divide and conquer" method was employed in decision tree concepts. [8], further stated that each node in a tree corresponded to a collection of cases. Terms should check whether these were under the same label or not, according to this entire training example. Then, if not the same label, select partitioning terms from the pooled classes of documents with similar values for the term and place each of them in a distinct subtree.

D. Random Forest

The random forest algorithm was the enlargement of the decision tree, as mentioned in [6]; constructing the actual world an axis of decision tree from training data in the real world. It essentially normalizes data so that it bonds to a nearby tree on the data scale. Random forest prototypes are important because they solve the decision tree's problem of unnecessarily "pushing" data points into a category.

E. Naive Bayes Algorithm

A Naive Bayes classifier [9] is a "simple probabilistic classifier based on Bayes' Theorem and strong independence assumptions." It calculates the document's subsequent probability of being assigned to multiple classes and assigns the document to the class with the highest subsequent probability, employing the autonomous feature as the probability model. As a result, the existence of one feature in a classification task has no bearing on the existence of other features.

F. Perceptron

A threshold function serves as an activation function in a perceptron, which is an artificial neuron. Assume an artificial neuron with input signals x_1, $x_2 \ldots$, x_n and associated weights w_1, $w_2 \ldots$, w_n with w_0 for constant, [10]. If the output of a neuron is assumed to be a perceptron, the equation is as follows in (2).

$$O(x1, x2, \ldots, xn) = \begin{cases} 1, \text{ if } Wo + W1X1 + \ldots + WnXn > 0 \\ -1, if\ W0 + W1x1 + \cdots + WnXn \leq 0 \end{cases} \quad (2)$$

G. Support Vector Machines

Support Vector Machines (SVM) are utilized for test classification which is a supervised classification algorithm. Text classifiers must cope with a large number of features [11] with high-dimensional input space. It is also capable of handling vast feature spaces and employed in place of adequate protection that is not based on the number of features. Furthermore, the majority of text categorization issues are linearly separable. Therefore, SVMs were developed to locate and search for linear separators.

H. Gaussian Naive Bayes

Gaussian Naïve Bayes is also known as Gaussian NB which is based on the Gaussian Normal Distribution and supports continuous data, as specified by [12]. It's a Naive Bayes variation to compute continuous data, continuous values were often associated with each class and distributed according to the normal distribution.

III. RELATED RESEARCH

Automated document classification is the machine learning fundamental that refers to assigning automatic categories among scanned images of the documents. KNN is the most researched topic by most researchers, with the most important aspect being to perform a detailed study over survey applications that were performed by implementing introductory data mining books and survey reports that were documented by [13], which proposed many improvements of KNN algorithms for implementing data classification. Another noteworthy study by [14] was on the weighted KNN classification method based on various symbolic characteristics, in which the distance was measured and calculated before being depicted in the form of tables to produce real-valued distances from symbolic domains that

also represent features. The authors claim that the suggested method outperforms existing algorithms such as KNN because it was tested on three different application areas, with the key advantage being training speed and ease of implementation.

[15], an optimization function based on the "leave-out-out cross-validation" technique and "greedy hill-climbing technique" and introduced three major "decision tree algorithms" was published that focused on an adjustment of weight while implementing KNN for identifying optimum weighted vector by using an optimization function that was based on the "leave-out-out cross-validation" technique and "greedy hill-climbing technique" and introduced three major "decision tree Many other studies conducted extensive surveys of applications based on various decision tree algorithms in the fields of machine learning and data mining technologies [16].

SVMs are a type of classification algorithm that works by examining a feature space and attempting to build a hyperplane to divide data points belonging to distinct classes [16][17]. It worked by employing a kernel function to translate data onto a higher-dimensional space and defining the hyperplane. Although SVMs are binary classifiers by nature, they can be adjusted for multiclass issues by employing pairwise classification, which treats a problem as a series of binary problems. The illustration in support vector machines is comprised of self-learning kernels, where the survey was demonstrated in SVMs and their mathematical foundation. Major parts of SVMs were applied to the implementation of text categorization in other works [18].

Many scholars used Decision Trees to solve the challenge of automatic affect identification. This was a simple classifier that used data observations and map observations to make class ownership decisions [16] [17]. It works by repeatedly querying a test instance for more information about the classes to which it may belong using a set of if-then rules.

IV. Methodology

A. Data Collection

The interviews were taken as a data collection tool from nine first-generation Ethiopian universities either face-to-face or via the medium of wire. We focused to question from research students, senior teachers, and individual researchers.

Our objective was to get the most efficient classification algorithms according to usage and level of understanding of the fundamentals of science. 720 research students, 225 teachers, and 135 individuals of computing domain as the sum of 1080 were questioned using the Question: “which algorithm is better for document classification, in terms of simplicity and accuracy?” We gave them eight different classification algorithms including; KNN, SVM, Naive Bayes,

Gaussian NB, Perceptron, Random Forest, Logistic Regression, and Decision Tree.

All of the interviewees did not answer the question, which makes a response rate of 92% (appx). 25% of the respondents felt that KNN is the most used algorithm and 20% of the respondents said that SVM is better in their experience. 15% agreed that Perceptron is the most popular and used, another 15% suggested that Gaussian NB is easy for them and the rest have recommended as described in Fig. 2.

Therefore, the study was compared with four different classification algorithms that the respondents felt were the most used classifiers with a response rate of 15% or above are, KNN, SVM, Perceptron, and Gaussian NB which were included in the analysis.

B. Experiments

This study was used on a whole meta-data set called Banknotes, Crowdsourced Mapping, and VxHeaven provided by UC Irvine [19] holds information about the different categories of data. We used the Irvine dataset to get the exact performance of every algorithm. Overall, functions and commands were used in the platform of Python 3.7.

The detail of the data is mentioned in Table I which is employed to perform the k-Fold cross-validation technique for training and testing the datasets were used. Here, the value is assumed as k=10 which shows the unbiased result of the datasets. KFold() Scikit-learn class used with an argument of the number of splits whether to shuffle the sample. We created an instance that splits a dataset into k folds split returned each group of train and test sets after calling the split() function. Index of a way returned into the original data samples of observation to use for train and test sets on each repetition.

Fig. 3 illustrates the first 10 rows in the banknotes dataset that were used to train and test the algorithm. Similarly, all the datasets were trained and tested.

1) Datasets: All the files are images in which banknotes are pictures of various banknotes and measuring different properties of currency and in particular, they categorized each of these banknotes as either counterfeit banknotes or not counterfeit (authentic).

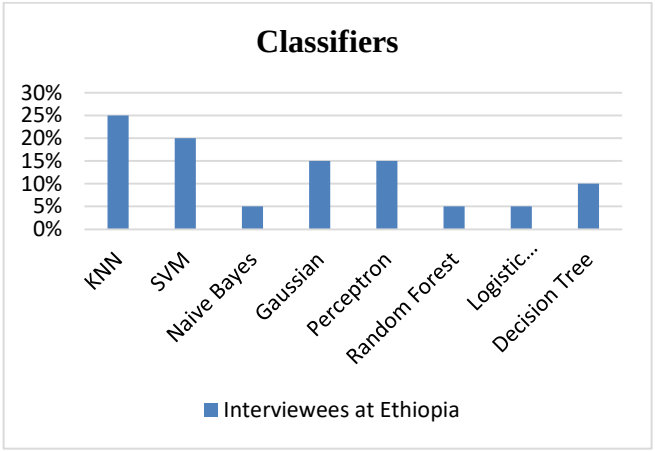

Fig. 2. Classifiers for Interviewees in Ethiopia.

TABLE I. Datasets And Data Size

Datasets	Data Size	Text Size
Banknotes	1372	10-fold Cross-Validation
Crowdsourced Mapping	10546	10-fold Cross-Validation
VxHeaven	2598	10-fold Cross-Validation

Crowdsourced Mapping Images were taken through satellites of different land cover classes of the farm, forest, orchard, and water being categorized. VxHeaven is the detection of malware observed through various features of the text in the Operating System.

2) Machine learning algorithms were evaluated using a resampling procedure called k-Fold cross-validation. A dataset is split into o number of groups known as k-Fold. The score with high variance may change the idea based on data to fit the algorithm or overestimate the skill of the algorithm due to biased data if the value of k is selected poorly. Here, k=10, which is less biased of the algorithm than another method to split the train or test result.

Each row of the dataset represents the banknote and has four different input values. These inputs have an output value of 0 or 1. 0 means a genuine (authentic) bill and 1 means a counterfeit bill.

So, this study used supervised learning to begin to predict some sort of function that can take four values as input and predict the output would be the algorithm was built using Python language, Jupyter Notebook as a compiler, and libraries including, Pandas, Scikit-Learn, and Matplotlib.

V. Result And Discussion

In evaluations, data were divided into two different sets, training and testing datasets using the k-Fold model. Training data sets were used to build the algorithm (classifier) and then tested the classifier for the prediction of the goal.

A. Confusion Matrix

A confusion Matrix is also known as an Error Matrix that is used to define the analyses of classification algorithms on a set of test data for which the true values are well known. It is a table that has two dimensions; actual value and predicted value as Table II illustrates.

TABLE II. Truth Table of Confusion-Matrix

		Predicted	
		No	***Yes***
Actual	**No**	*TN*	*FP*
	Yes	*FN*	*TP*

TN: True-Negative, TP: True-Positive, FN: False-Negative, FP: False-Positive.

```
df = pd.read_csv("banknotes.csv")
print(df.head(10))
```

```
   variance  skewness  curtosis  entropy  class
0  -0.89569   3.00250 -3.606700 -3.44570      1
1   3.47690  -0.15314  2.530000  2.44950      0
2   3.91020   6.06500 -2.453400 -0.68234      0
3   0.60731   3.95440 -4.772000 -4.48530      1
4   2.37180   7.49080  0.015989 -1.74140      0
5  -2.21530  11.96250  0.078538 -7.78530      0
6   3.94330   2.50170  1.521500  0.90300      0
7   3.93100   1.85410 -0.023425  1.23140      0
8   3.97190   1.03670  0.759730  1.00130      0
9   0.55298  -3.46190  1.704800  1.10080      1
```

Fig. 3. Sample of First 10 Rows of Banknotes.

So, the confusion matrixes of mentioned classification algorithms are illustrated in the below matrixes.

Confusion Matrix 1.1 KNN

$$\begin{matrix} 363 & 1 \\ 0 & 322 \end{matrix}$$

Confusion Matrix 1.2 SVM

$$\begin{matrix} 389 & 7 \\ 0 & 290 \end{matrix}$$

Confusion Matrix 1.3 Perceptron

$$\begin{matrix} 382 & 1 \\ 9 & 294 \end{matrix}$$

Confusion Matrix 1.4 Gaussian NB

$$\begin{matrix} 348 & 39 \\ 60 & 239 \end{matrix}$$

The evaluation metrics are classified in Accuracy, Precision, Recall, Error Rate, and F-Score which are described in paragraphs (1), (2), (3), (4), and (5) consecutively below that what variable function for their parameters calculated. The summary of all the evaluations of the analyzed algorithm is shown in Table III.

1) Accuracy: It is close to the measured value to the actual (true) value.

$$\text{Accuracy} = \left(\frac{\text{TP+TN}}{\text{Total Tuples in Test Dataset}}\right) \quad (3)$$

The accuracy of these four algorithms; KNN, SVM, Perceptron, and Gaussian NB were scored 99.85%, 98.98%, 98.54%, and 85.57% respectively.

2) Precision: It is an evaluation analysis technique that finds the closeness of the measured values to each other.

$$\text{Precision} = \left(\frac{\text{TP}}{\text{Predicted Yes}}\right) \quad (4)$$

Performance of precisions of algorithms KNN, SVM, Perceptron, and Gaussian NB were 0.996, 0.976, 0.996, and 0.856 respectively.

3) Recall: The ratio of all correctly predicted positive predictions was measured using Recall.

$$\text{Recall} = \left(\frac{\text{TP}}{\text{Actual Yes}}\right) \quad (5)$$

The algorithms, KNN, SVM, Perceptron, and Gaussian NB reached 100%, 100%, 97%, and 79% respectively.

4) Error rate: It is an evaluation analysis technique that calculates the number of all incorrect predictions divided by the total number of the datasets. The worst error rate is 1.0 and the best error rate is 0.0.

$$\text{Error Rate} = 1 - \text{Accuracy} \quad (6)$$

KNN, SVM, Perceptron, and Gaussian NB algorithm errored 0.002, 0.011, 0.015, and 0.145 respectively of error rate declares.

5) F-Score: It is an evaluation analysis technique that calculates the harmonic mean of precision and recall.

$$\text{F-Score} = \left(\frac{2(P*R)}{P+R}\right) \quad (7)$$

Where *P* is Precision and *R* is Recall.

TABLE III. EVALUATED TECHNIQUES WITH PARAMETERS FOR THE ALGORITHM

Evaluation Technique	Analyzed Algorithm			
	KNN	SVM	Perceptron	Gaussian NB
Accuracy	99.85%	98.98%	98.54%	85.57%
Precision	0.996	0.976	0.996	0.856
Recall	100%	100%	97%	79.90%
Error Rate	0.002	0.011	0.015	0.145
F-Score	0.997	0.987	0.982	0.826

All algorithms: KNN, SVM, Perceptron, and Gaussian NB were marked 0.997, 0.987, 0.982, and 0.826 respectively in the F-Score evaluation shows.

B. Evaluation Time

Time is one of the important parameters to check the efficiency of any algorithm. It helps to test a comparative algorithm. Timing of classification of document and execution were tested and analyzed using datasets as shown in Table IV and Table V. Performance of execution was measured in python using the timeit() function. Although python is slower than C++ libraries.

The optimization Table IV shows the least classification time and running time of KNN is ~413 and 3.6978 consecutively which is the lowest of another algorithm.

TABLE IV. EXECUTION TIME OF CLASSIFICATION TIME OF THE ALGORITHM

File Size (MB) (Datasets)	Classification Time			
	KNN	*SVM*	*Perceptron*	*Gaussian-NB*
0.22	~400	~409	~413	~411
0.4	~422	~428	~437	~438
0.32	~417	~419	~433	~434
Average	~413	~418.67	~427.6667	~427.6667

TABLE V. EXECUTION TIME OF RUNNING TIME OF THE ALGORITHM

File Size (MB) (Datasets)	Running Time			
	KNN	*SVM*	*Perceptron*	*Gaussian-NB*
0.22	3.6827	3.6828	3.7055	3.7076
0.4	3.7155	3.716	3.7188	3.7189
0.32	3.6952	3.6954	3.6967	3.6966
Average	3.6978	3.69807	3.707	3.7077

VI. CONCLUSION

In this comparative study, we compared the performance and efficiency of various machine learning classification algorithms. KNN, SVM, Perceptron, and Gaussian NB using a meta-data set created by UC Irvine as an experiment. Authors estimate that the used data set is the meta-size of the data whose training and testing procedure was taken using k-Fold methods and experiments. K-Fold cross-validation technique for training and testing of the datasets was used where the value of k=10 was assumed. In addition, the study focused on identifying a better algorithm for document classification that executed well on different meta-data sets. However, it was assessed that the accuracies of the tools depend on the data set used. Also, noted that the classifiers of a special group did not perform with equal accuracies in terms of the overall performance accuracies algorithm. KNN performed better than the other classification algorithms with a fewer error rate of 0.0002 the efficiency of the least classification time and running time with ~413 and 3.6978 microseconds consecutively. It is concluded that KNN classifiers have been recognized as the best algorithm for document classification with a percentage accuracy of 99.85%, recall value of 100%, and f-Score of 0.997. There would be platform variations of the algorithm that might be the case of study in the future.

ACKNOWLEDGMENT

This paper is written by us using meta-data. We thank each other for the collaboration, collection, and execution of data that we passed. Authors are working in the area of Artificial Intelligence and substitute areas like machine learning, and deep learning including data sciences. These sciences are heavily dependent on the tools and algorithms used and tested in the data of different fields. Authors considered carrying out more into the domain. This work is not supported by any funder or our affiliated organization.

REFERENCES

[1] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, “Text Classification using Machine Learning Tanique”, WSEAS TRANSACTION on COMPUTERS, vol. 4, no. 8, pp. 966-974, 2005.

[2] Y. Yang, “An Evaluation of Statistical Approaches to Text Categorization”, Journal of Information Retrieval, vol. 1, pp. 67-88, 1999.

[3] R.Johnson, and T Jhang, “Supervised and Semi-Supervised test Categorization using lstm for region Embedding”, preprint, arXiv:1602.02373, 2016.

[4] Goller, Loning, Will, and Wolf, “Automatic Document Classification”, International Symposium for Information, pp. 145-162, 2000.

[5] M.T.U.U. Hugo Moriz, “A Comparative Study of Machine Learning Algorithm for Document Classification”, Uppsala Unpublished. 2020.

[6] K. Kowsari, K Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text Classification Algorithm: A Survey Information” vol. 10, no 4, pp. 150, 2019.

[7] Tam Santoso, and R. Setiono, “A Comparative Study of Centroid Based, Neighborhood-based and Statistical Approaches for Effective Document Categorization”, ICPR 02 Proceedings of the 16th International Conference on Pattern Recognition (ICPR’02), vol. 4, no. 4, 2002, pp. 235-238, 2002.

[8] Mou R. Men, G. Li, Y. Xu, L.Zhang, R. Yan, and Z.Jin, “Natural Language inference by tree-based convolutional and heuristic matching”, arXiv:1512.08422, 2015.

[9] I. Rish, “An Empirical Study of the Naïve Bayes Classifier”, Proc. Of the IJCAI-01, Workshop in Empirical Methods in Artificial Intelligence, CiteULike-article-id-352583, 2001.

[10] V. N. Krishnachandran, “Machine Learning”, Vidya Center for Artificial Intelli9gence Research, India, 2018.

[11] Pratiksha P. Pawar, and S.H. Gawande, “A Comparative Study on Different Types of Approaches to Text Categorization”, International Journal of Machine Learning and Computing, vol. 2, pp. 423-426, 2021.

[12] S. Agarwal, B. Jha, T. Kumar, M.Kumar, and P. Ranjan, “Hybrid of Naïve Bayes and Gaussian Naïve Bayes for Classifiaction: A Map Reduace Approach”, vol. 8, no. 6S3, pp. 266-268, 2019.

[13] R. Agrawal, T. Imielinski, and A. N. Swami, "Database Mining: A Performance Perspective", IEEE Trans, Knowledge and Data Engineering, vol. 6, pp. 914-925, 1993.

[14] Jiawei Han, Jian Pei, and Micheline Kamber, "Data Mining, Concepts and Techniques", 2nd Edition: Morgan Kaufmann, 2006.

[15] H.Z., "A Sort Introduction to Data Mining and its Application", IEEE, 2011.

[16] B. Rama, P. Parveen, H. Sinha, and T. Chaudhary, "A Study on Causal Rule Discovery with PC Algorithm", International Conference on Infocom Technologies and Unmanned System (ICTUS), Dubai 2017.

[17] S. Nagaparameshwara Chary, and B. Rama, "A research Travelogue on Classification Algorithms using R. Programming", International Journal of Recent Technologies and Engineering (IJRTE), vol. 8, no. 4, pp. 9155-9158, 2019.

[18] P. Praveen, B. Rama, and Uma N. Dulhare, "A Study on monthelic Divisive Hierarchical Clustering Method", International Journal of Advanced Scientific Technologies Engineering and Management Sciences, vol. 3, 2017.

[19] L. Volker, "Bank Notes Database- University of Applied Sciences", Ostwestfalen-Lippe: Lemgo: UC Irvine, 2012.