

Using Artificial Intelligence Approaches to Categorise Lecture Notes

Naushine Bibi Baijoo, Khusboo
Bharossa
Faculty of Engineering
University of Mauritius
Reduit, Mauritius

Somveer Kishnah
Software and Information Systems
Dept
University of Mauritius
Reduit, Mauritius

Sameerchand Pudaruth
ICT Department
University of Mauritius
Reduit, Mauritius

Abstract—Lecture materials cover a broad variety of documents ranging from e-books, lecture notes, handouts, research papers and lab reports amongst others. Downloaded from the Internet, these documents generally go in the Downloads folder or other folders specified by the students. Over a certain period of time, the folders become so messy that it becomes quite difficult to find our way through them. Sometimes files downloaded from the Internet are saved without the certainty that they will be used or revert to in the future. Documents are scattered all over the computer system, making it very troublesome and time consuming for the user to search for a particular file. Another issue that adds up to the difficulty is the improper naming conventions. Certain files bear names that are totally irrelevant to their contents. Therefore, the user has to open these documents one by one and go through them to know what the files are about. One solution to this problem is a file classifier. In this paper, a file classifier will be used to organise the lecture materials into eight different categories, thus easing the tasks of the students and helping them to organise the files and folders on their workstations. Modules each containing about 25 files were used in this study. Two machine learning techniques were used, namely, decision trees and support vector machines. For most categories, it was found that decision trees outperformed SVM.

Keywords—Classification; lecture materials; machine learning; support vector machines; decision trees

I. INTRODUCTION

The rapid advancements in IT have brought about an exponential increase in the number of electronic documents. Documents that were presented on paper in the past are today created, stored, distributed and displayed digitally [1]. This trend has captured a wide variety of fields, if not all. The education field has not been left behind in the process. It has evolved alongside with the advent of new technologies.

Students nowadays have thousands of files on their workstations, scattered in different folders, on different drives, etc. Some files have meaningful names while others do not. The easy access to information has also led to an increase in the amount of irrelevant information. Information from web pages, news articles, presentations, papers are saved on the machines without the certainty that they will be of some use in the future. This usually costs users a great deal of time looking for a particular file especially if all the files are scattered in different places on the computer system and the file in question

is not properly named. Therefore, an automatic file classification system is of utmost importance. The role of the file classifier would be to go through all the files in a given folder and determine the best fitting category for each file.

This paper proceeds as follows. Section II gives a description of the different techniques that are used for the classification process. Section III describes the methodology used and the tasks that need to be carried out to classify the documents. Section IV outlines the implementation process and critically analyses and evaluates the results of the classifiers. Finally, we conclude the study in Section V.

II. LITERATURE REVIEW

A. Text Mining

Text mining, also known as text analytics, is a hypernym used to describe the wide range of technologies in place to analyze and process unstructured and semi-structured textual data [2], [3]. These technologies are used to extract meaningful information from documents or files that would then serve particular purposes. The most common theme behind all the technologies is to turn textual information into numbers. Algorithms are then applied to the numerical format of the words, documents and eventually to full databases. The data is then handled and processed as per to one's requirements.

Text mining involves the applications of techniques from fields such as information retrieval, information extraction, natural language processing, machine learning, classification, clustering and text categorisation. Information retrieval is an area pertaining to the organisation, examination, storage and retrieval of information from different sources. It performs several tasks such as document ranking and document classification. This paper discusses two main classification techniques, namely decision trees and support vector machines.

B. Decision Trees

Decision trees are a very simple but powerful classification method. One advantage of a decision tree is that it can be very easily interpreted by humans. It is commonly used in pattern recognition problems for knowledge systems [4]. A decision tree is very similar to a flow diagram. It consists of an internal node with many attached branches and leaf nodes. A test on a particular element is designated by the internal node. The branches denote the result of that experiment and finally, the class distribution is indicated by the leaf nodes [5]. The

topmost node is known as the root node and it is denoted by an oval. Rectangles are used to symbolise the internal nodes. The leaf nodes, on the other hand, are circular in shape.

A list of attributes is made for measurement in order to create a decision tree. A target attribute is then chosen for prediction. All data is processed to know the number of times an attribute appears in each document. Decision trees use the concept of entropy for splitting attributes – reducing the number of attributes. Splitting the attributes results in a hierarchy of branches. These branches or nodes are called the decision tree. All nodes can form another branch of node. Each branch in the tree produces an observation. This observation is made using the state of one of the fields in the dataset. Another method used for splitting is called pruning. There are two types of well-known pruning namely pre-pruning and post-pruning also known as forward pruning and back-pruning respectively. In pre-pruning, the user decides when to stop adding attributes during the building process. As a result, it can lead to very biased decisions as individual attributes do not contribute much to the decision. Post-pruning is different in that the decision tree is fully built prior to pruning the elements [6].

Decision trees are efficient for new and unseen inspections. However, building a decision tree can be very time-consuming. One serious weakness of decision trees is the problem of error propagation throughout a tree. Decision trees are built by a series of local decisions. These local decisions have a carry-over effect. Therefore, if one of the local decisions goes wrong at some point in time, all successive decisions are bound to be bad as well. In such a case, the correct path of the tree might not be returned [6].

C. Support Vector Machines

SVM algorithms are a learning method introduced by Vladimir Vapnik and colleagues. They are used for pattern recognition, classification and regression. Support vector machines have been very successful in various learning areas [7], [8]. SVMs construct hyperplanes for linearly separated patterns. The basic idea in SVM is to find a mediator which separates multi-dimensional data into two classes [9]. SVMs work towards maximising predictive accuracy while avoiding over-fitting. SVMs give very significant results for applications involved in classifying text, recognizing hand-written characters, classifying images and also in bio-informatics. One of the strongest points for SVMs is that they impose no limit on the number of attributes that can be used. However, the only problem is that SVMs require a lot of memory [10].

III. METHODOLOGY

The very first step to the classification of the lecture materials is to build a dataset. A dataset in this study is simply a bulk of relevant documents. Eight categories of lecture materials amounting to 213 files were selected and were put in a common folder. Table I shows the categories and the number of files used in each category.

NLTK (Natural Language Toolkit) has been used to process the files. It is the most commonly used platform to write Python programs to interact with textual data [11]. It is open source software and is made up of a plethora of libraries to allow for the manipulation of high-level data.

TABLE I. DETAILS OF CATEGORIES

Category	Number of files
Cyberlaws	23
Database	35
Enterprise Resource Planning	25
Management Information Systems	26
Multimedia	26
Networking	33
Security	24
Software Engineering	21

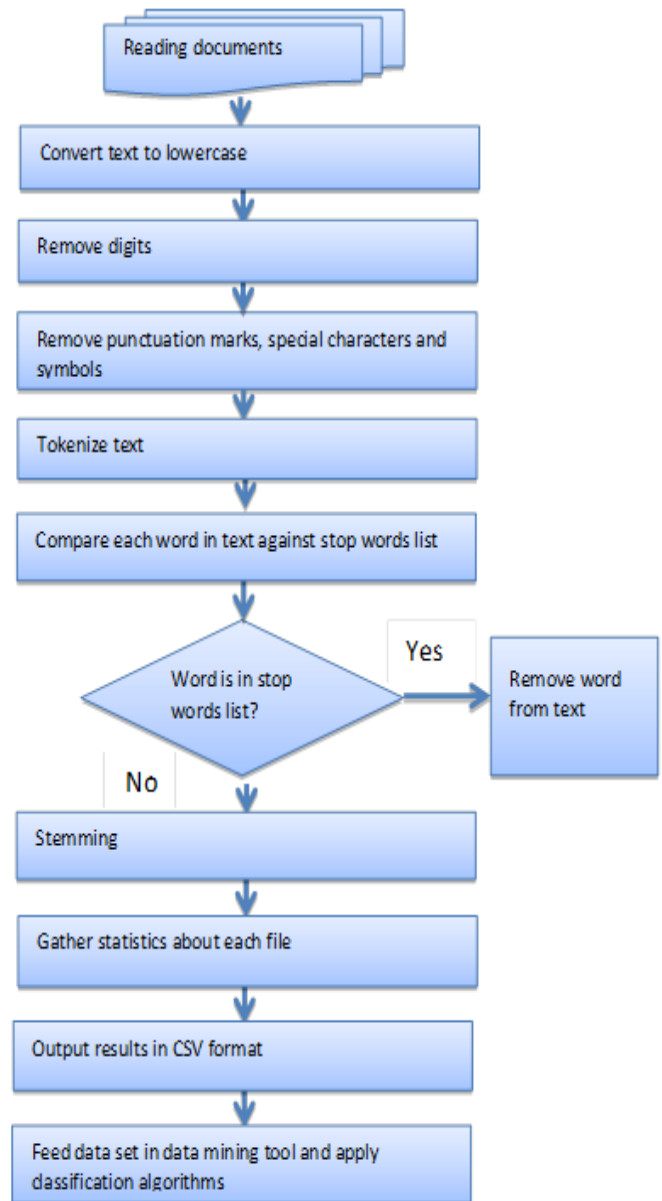


Fig. 1. Flowchart Outlining the Steps of the Implementation Process.

Firstly, the documents are converted to lowercase to avoid ambiguities at later stages. Secondly, the files are cleaned. All the punctuation marks, special symbols, digits and special characters are removed. The series of words is then subjected to the process of tokenization which breaks the documents into distinct words or tokens. Each word is then checked against NLTK's stopword list. The stopword list is a large body of text consisting of 11 languages with a total of 2,400 stop words [12]. Stop words are words like 'the', 'is', 'a', that do not carry much weight when it comes to determining the best category of a file. Thus, all stop words are eliminated from the documents leaving us with only potentially useful and meaningful words.

The last step in the cleaning process is the application of stemming to the words, as shown in Fig. 1. Stemming is a method for removing the affixes from a word in order to end up only with the stem which is also known as the root. It is a common technique used in search engines for indexing words. The search engine stores only the stems, instead of keeping all the different forms of a word. This is very helpful as it reduces the size of the index by a considerable amount, thus improving performance and retrieval accuracy. One of the most popular stemming algorithms is the Porter Stemmer Algorithm. It removes and replaces well known suffixes of English words [13]. NLTK supports a number of other stemming algorithms as well, namely the Lancaster stemmer, Regexp stemmer and the Snowball stemmer [14]. For this project, the Snowball stemmer has been used.

Once the documents are cleansed, the array of meaningful and stemmed words is further processed to get the frequencies of each word in each document. The outputs are stored in CSV files. These CSV files produced are fed into WEKA [15]. The following section gives more details about the classification process in WEKA and evaluates the classifier outputs.

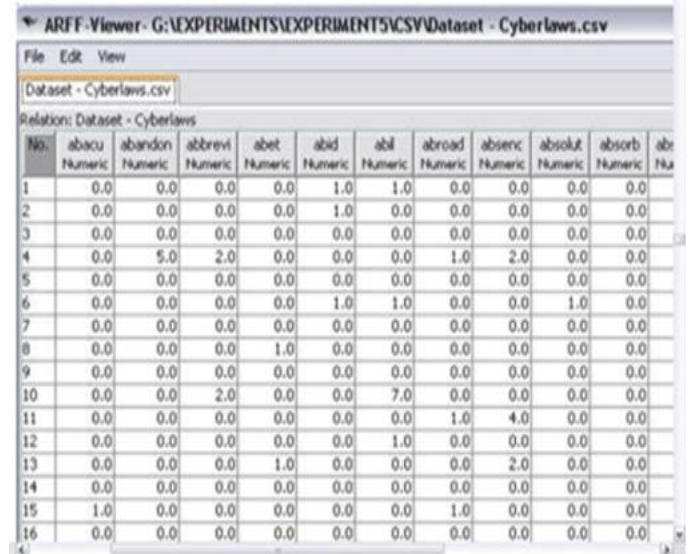
IV. IMPLEMENTATION AND EVALUATION

WEKA supports a particular file format known as the ARFF data format. ARFF stands for Attribute – Relation File Format. It is an ASCII file describing a set of samples having a number of elements in common. The ARFF-Viewer tool in WEKA allows for the conversion of CSV data files to the ARFF data format. An ARFF data file has a very particular format. It basically has two distinct sections, the header part followed by the data information. It starts with @RELATION, which gives the name of the file, followed by @ATTRIBUTE, giving a list of the file's attributes and lastly @DATA.

All the attributes in an ARFF file are of type 'numeric' since we are dealing with the frequencies of the words in the documents. The data is represented as a stream of numbers. Viewed in WEKA's ARFF-Viewer, we are presented with a tabular form of the file (Fig. 2), which is easier to interpret.

The datasets for all eight categories of lecture materials were classified using two different machine learning techniques and the outputs were compared. From existing works, we have noticed that it is a common practice to test the algorithms with a balanced number of positive and negative samples. Thus, we have used an equal number of documents to carry out the experiments. A binary approach was followed, i.e.

for each category we took 15 positive samples and 15 negative samples (which was termed as the 'Others' category).



No.	abacu Numeric	abandon Numeric	abbrevi Numeric	abet Numeric	abid Numeric	abil Numeric	abroad Numeric	absenc Numeric	absolut Numeric	absorb Numeric	ab N/A
1	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	5.0	2.0	0.0	0.0	0.0	1.0	2.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
10	0.0	0.0	2.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	
11	0.0	0.0	0.0	0.0	0.0	0.0	1.0	4.0	0.0	0.0	
12	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	
13	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0	0.0	0.0	
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
15	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Fig. 2. The ARFF-Viewer.

A. J48

The datasets were first classified using the J48 decision tree algorithm in WEKA. J48 normally selects a set of keywords in the set to base its decision on [16]. However, the selection of that keyword is not stable as a little change in the dataset may alter the results by a great amount. Also, the keyword chosen may not always reflect the intended category. An example is given in Fig. 3.

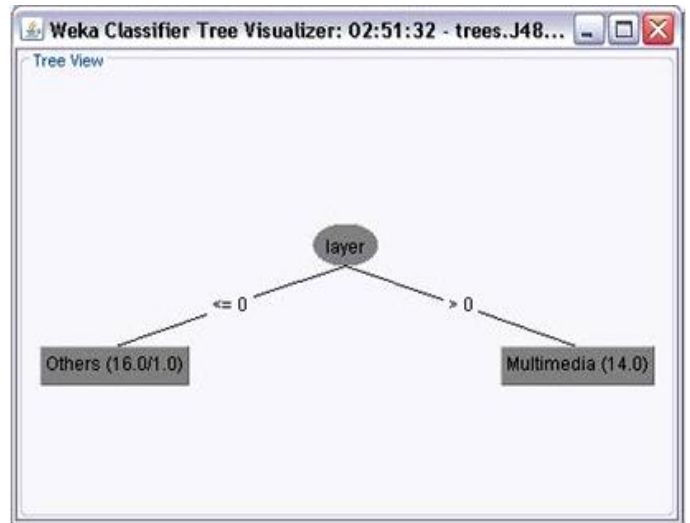


Fig. 3. Decision Tree for Multimedia.

Fig. 3 shows the classifier's tree visualizer for Multimedia. The word 'layer' has been chosen to decide between the Multimedia and the Others categories. This word however is not appropriate as it may be used in many contexts other than Multimedia. Words like 'multimedia', 'image', 'video' would have been more appropriate in this case.

B. LibSVM

The datasets were subjected to a second round of classification, this time with LibSVM [17]. The classification for the Multimedia category, for instance, yielded very good results. All of the 15 documents pertaining to this category were correctly classified.

TABLE II. CONFUSION MATRIX

Predicted Class			Actual Class
	Others	Multimedia	
Others	7	0	
Multimedia	8	15	

Table II indicates that out of 15 files that are actually from the Others category, seven of them were correctly classified while the remaining eight were not. They were classified as Multimedia files instead of Others. As for the Multimedia files, it was an error-free classification.

C. Summary of Outputs

Table III shows a summary of the classifier outputs with J48 and LibSVM for all the 8 categories of lecture materials.

A pertinent observation is the meagre percentage of correctly classification instances for the Database category. Database is a very common field in computing. It merges with many other fields in a fluid manner and it may be applied in a variety of computing contexts. Therefore, files from Enterprise Resource Planning (ERP) and Management Information Systems (MIS) files may well fall in the Database category. This is one potential reason for the downfall in the positive percentage for this particular category.

TABLE III. SUMMARY OF CLASSIFIERS OUTPUTS

Categories	Correctly classified instances	Incorrectly classified instances	Correctly classified instances	Incorrectly classified instances
	J48		LibSVM	
Cyberlaws	23	7	25	5
	76.7%	23.3%	83.3%	16.7%
Database	19	11	20	10
	63.3%	36.7%	66.7%	33.3%
Enterprise Resource Planning	24	6	22	8
	80%	20%	73.3%	26.7%
Management Information Systems	24	6	22	8
	80%	20%	73.3%	26.7%
Multimedia	26	4	22	8
	86.7%	13.3%	73.3%	26.7%
Networking	27	3	25	5
	90%	10%	83.3%	16.7%
Security	28	2	26	4
	93.3%	6.7%	86.7%	13.3%
Software Engineering	29	1	22	8
	96.7%	3.3%	73.3%	26.7%

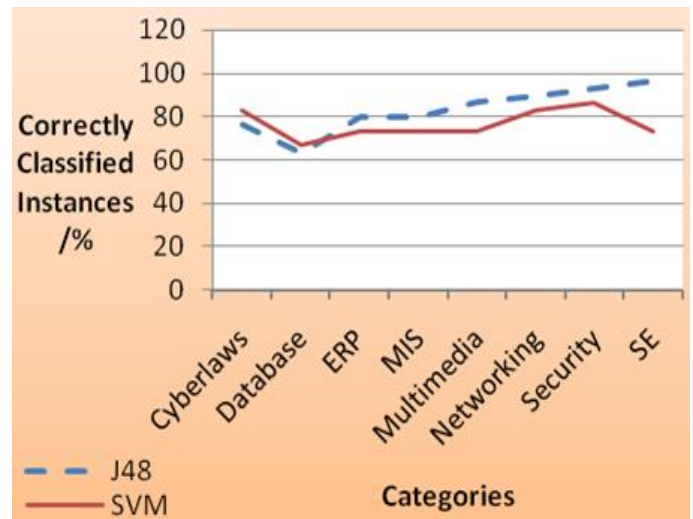


Fig. 4. Line Graph Comparing Results of J48 with SVM.

The overall accuracy for J48 is 83.3% while for SVM it was 76.7%. From these statistics and from Fig. 4, we can see that J48 has done slightly better in this scenario.

D. Accuracy of Outputs

The accuracy of the classifier outputs in WEKA is determined by some very distinct parameters. These parameters are: True Positive Rate (TP Rate or Recall), False Positive Rate (FP Rate), Precision and the F-measure.

TABLE IV. ACCURACY BY CATEGORY

Categories		TP Rate	FP Rate	Precision	F-measure
Cyberlaws	J48	0.667	0.133	0.8	0.741
	LibSVM	0.933	0.267	0.778	0.848
Database	J48	0.733	0.467	0.611	0.667
	LibSVM	0.4	0.067	0.857	0.545
ERP	J48	0.867	0.267	0.765	0.813
	LibSVM	0.6	0.133	0.818	0.692
MIS	J48	0.8	0.2	0.8	0.800
	LibSVM	0.933	0.467	0.667	0.778
Multimedia	J48	0.933	0.2	0.824	0.875
	LibSVM	0.467	0	1	0.636
Networking	J48	0.933	0.133	0.875	0.903
	LibSVM	0.8	0.133	0.857	0.828
Security	J48	1	0.133	0.882	0.938
	LibSVM	1	0.267	0.789	0.882
Software Engineering	J48	1	0.067	0.938	0.968
	LibSVM	0.667	0.2	0.769	0.714

Table IV shows the accuracy by category for both classifiers. A TP rate of one is an ideal result. It means that all or almost of the documents were correctly classified. All Security files were correctly classified, hence yielding a recall of 100% with both classifiers. Fields like Software Engineering and Cyberlaws, which are quite distinct from the rest, have also fetched high values. The recall value for Multimedia is exceptionally low for the SVM classifier. However, the explanation for this can be seen in Table II. This is because many files from the Others category were classified as being in the Multimedia category due to the presence of certain superfluous words. Nevertheless, the precision values are very high. A TP rate as low as 0.4 is an undesirable result, which is indicative of poor classification of the files. It is noticed that the TP rates for ERP and MIS are not very high too. These values point towards the confirmation of the observation that the modules ERP, MIS and Database bear a lot of similar words, hence some files were incorrectly classified. In general, the values for precision and recall were appreciably high.

V. CONCLUSIONS

This paper discussed the classification of lecture materials. Two hundred and thirteen documents from eight different university modules were selected and were classified into pre-defined sets. The documents were classified using two different machine learning techniques namely decision trees and support vector machines. A number of experiments were carried out and the results of the classification were critically analysed. The outputs' parameters and various other factors showed that J48 was a better classification technique than SVM for this particular case. The overall accuracy for J48 was found to be 83.3% while for SVM it was only at 76.7%. However, these results cannot be generalised as our data set was quite small. In the future, we intend to repeat these experiments with many more files and more classifiers such as kNN, Naïve Bayes and artificial neural networks. Document size, i.e. the number of words in each file will also be taken into consideration.

REFERENCES

- [1] H. Abdulkadhim, M. Bahari, A. Bakri, and W. Ismail, "A research framework of Electronic Document Management Systems (EDMS) implementation process in Government," *Journal of Theoretical and Applied Information Technology*, vol. 81(3), pp. 420-432, 2015.
- [2] G. Miner, D. Delen, J. Elder, A. Fast, T. Hill, and R. Nisbet, *Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications*. Oxford: Academic Press, 2012.
- [3] S. Binkheder, H. Y. Wu, S. Quinney, and L. Li, "Analyzing Patterns of Literature -Based Phenotyping Definitions for Text Mining Applications," 2018 IEEE International Conference on Healthcare Informatics (ICHI), 4-7 June 2018, New York, USA.
- [4] Y. Ben-Haim, and E. Tom-Tov, "A Streaming Parallel Decision Tree Algorithm," *The Journal of Machine Learning Research*, vol. 11, pp. 849-872, 2010.
- [5] J. Han, M. Kamber, and J. Pei, *Data Mining: Concept and Techniques*, 3rd ed. MA: Morgan Kaufmann, 2011.
- [6] M. Jaworski, P. Duda, and L. Rutkowski, "New Splitting Criteria for Decision Trees in StationaryData Streams," *IEEE Transactions on NeuralNetworks andLearningSystems*, vol. 29(6), pp. 2516-2529, 2018.
- [7] S. Tong, and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45-66, 2002.
- [8] Y. You, J. Demmel, K. Czechowski, L. Song, and R. Vuduc, "Design and Implementation of a Communication -Optimal Classifier for Distributed Kernel Support Vector Machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28(4), pp. 974-988, 2017.
- [9] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," 10th European Conference on Machine Learning (ECML), pp. 137-142, 21-23 April 1998, Chemnitz, Germany.
- [10] B. Bryant, H. Sari-Sarraf, R. Long, and S. Antani, "A Kernel Support Vector Machine Trained Using Approximate Global and Exhaustive LocalSampling," 4th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, pp. 267-268, 5-8 December 2017, Texas, USA.
- [11] NLTK 3.3 Documentation. Retrieved January 4, 2018, from: <http://www.nltk.org/>
- [12] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol: O'Reilly Media, 2009.
- [13] J. Perkins, *Python Text Processing with NLTK 2.0 Cookbook*. Birmingham, Packt Publishing Ltd, 2010.
- [14] A. Schofield, and D. Mimno, "Comparing Apples to Apple: The Effects of Stemmers on Topic Models," *Transactions of the Association for Computational Linguistics*, vol. 4, 287-300, 2016.
- [15] Weka 3: Data Mining Software in Java. Retrieved January 10, 2018, from: <http://www.cs.waikato.ac.nz/ml/weka/>
- [16] F. Borges, R. Fernandes, A. M. Lucas, and I. Silva, "Comparison Between Random Forest Algorithm and J48 Decision Trees Applied to the Classification of Power Quality Disturbances", 11th International Conference on Data Mining (DMIN), 27-30 July 2015, Las Vegas, Nevada.
- [17] F. Wang, Y. Yang, C. Liu, H. Wang, and X. Weng, "Analysis of Influencing Factors of Water Traffic Accidents based on LIBSVM", 27th International Ocean and Polar Engineering Conference, 25-30 June 2017, San Francisco, California.