Classification and Clustering for Neuroinformatics: Assessing the Efficacy on Reverse-Mapped NeuroNLP Data using Standard ML Techniques

Nidheesh Melethadathil
Amrita School of Biotechnology
Amrita VishwaVidyapeetham (Amrita University), Kollam, Kerala, India
nidheesh@am.amrita.edu

Priya Chellaiah
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham (Amrita University), Kollam, Kerala, India
priyac@am.amrita.edu

Bipin Nair
Amrita School of Biotechnology
Amrita VishwaVidyapeetham (Amrita University), Kollam, Kerala, India
bipin@amrita.edu

Shyam Diwakar, *SM*IEEE
Amrita School of Biotechnology
Amrita VishwaVidyapeetham (Amrita University), Kollam, Kerala, India
shyam@amrita.edu

Abstract*—NeuroinformaticsNatural Language Processing (NeuroNLP) relies on clustering and classification for information categorization of biologically relevant extraction targets and for interconnections to knowledge-related patterns in event and text mined datasets. The accuracy of machine learning algorithms depended on quality of text-mined data while efficacy relied on the context of the choice of techniques. Although developments of automated keyword extraction methods have made differences in the quality of data selection, the efficacy of the Natural Language Processing (NLP) methods using verified keywords remain a challenge. In this paper, we studied the role of text classification and document clustering algorithms on datasets, where features were obtained by mapping to manually verified MESH terms published by National Library of Medicine (NLM). In this study, NLP data classification involved comparing 8techniques and unsupervised learning was performed with 6 clustering algorithms. Most classification techniques except meta-based algorithms namely stacking and vote, allowed 90% or higher training accuracy. Test accuracy was high (=>95%) probably due to limited test dataset. Logistic Model Trees had 30-fold higher runtime compared to other classification algorithms including Naïve Bayes, AdaBoost, Hoeffding Tree. Grouped error rate in clustering was 0-4%. Runtime-wise, clustering was faster than classification algorithms on MESH-mapped NLP data suggesting clustering methods as adequate towards Medline-related datasets and text-mining big data analytic systems.

Keywords—NeuroNLP, classification, clustering, Neuroinformatics, accuracy.

I. INTRODUCTION

Extraction of meaningful information from huge online data repository via computationally efficient text mining methods has been a challenge [1]. With an important role in identifying and representing the contents, the relevance of keywords has become significant in document and text mining. In research articles, there are limited numbers of user-supplied keywords. Even though there are automatic keyword selection tools, which help to identify larger number of keyword, ensuring the quality of the selected keywords is still an issue.

To understand text categorization via efficient mining, we used a structured knowledge approach to process NLP datasets. We developed a neuro informatics platform called ANeuroNLP (manuscript under preparation) with a machine learning framework including document clustering to improve the efficacy of categorization of search process based on terms from text mined research articles. In many reported studies, the efficacy of data mining depended on the reliability of keywords used as features in the extracted datasets.

In this study, we present results of document data clustering and text classification where input dataset was pre-processed to have keywords based on a validated approach of reverse mapping mesh words from MetaMap[2]. ANeuroNLP uses MetaMap, allowing bioNLP datasets mapped to the UML metathesaurus. Features like author-defined acronyms and abbreviations retrieve the metathesaurus for concepts which were related to the input and word sense disambiguation makes MetaMap an authentic validator for the biomedical texts. Data mining experiments were carried out with datasets which were originally derived from PubMed later pre-processed using MetaMap APIs showed improvements in both classification and clustering.

II. PREVIOUS STUDIES

With several classification and clustering algorithms in data mining, it was contextual to explore the role of the algorithms in NeuroNLP and NLP data classification. Studies on machine learning algorithms on microarray datasets have been reported

978-1-4799-8792-4/15/$31.00

with varying efficacies[3]. Stream data mining using Hoeffding Tree, Naive Bayesian, Very Fast Decision Tree (VFDT) and Concept Adapting Very Fast Decision Tree (CVFDT) was proposed in [4]. Concept Adapting Very Fast Decision Tree performed better than VFDT and Naïve Bayesian techniques. A study of classification algorithms on predicting salary classes of employees by taking the data from public database like UCI census dataset indicated decision tree and Naive belief network performed well for such datasets[5]. Performance of ADTree, Simple Cart, ZeroR, J48 &Naïve Bayes for a course recommendation system were analyzed ADTree was reported to have performed better[6]. The efficacy of classification algorithms highly depended on feature selection mechanisms[7][8] . Clustering on narrow domain abstracts posed challenges due to low data size. To circumvent data related issues, feature selection techniques like transition point were used for improving accuracy[9].A study on different clustering algorithms such as K-means, graph-based approach and SVD-based method for short text documents was performed and reported that the graph based approach performed better with minimal cluster error [10].The performance of various classification and clustering algorithms [11][12] was evaluated to identify the appropriateness of such tools. With raw datasets without MESH-based remapping, accuracy was low and unreliable. Therefore, we evaluated the accuracy of classification and clustering on pre-processed data sets with mapped key terms as features.

III. METHODS

We used the ANeuroNLP tool (manuscript in preparation, see Fig. 1) to generate a pre-processed dataset. Here, the dataset was a collection of keywords related to neuroscience-based search terms. Two different search terms, namely "*cerebellum function ataxia*" and "*cerebellum physiology ataxia*" were given to the tool in order to generate the datasets, which were then extracted as reverse mapped dataset from MetaMap. While processing the search query ANeuroNLP created a database table which contains the details about the given search query such as pubmed_id, title of the paper, author list, mesh_head etc**.** Relevant keywords were loaded from the MetaMap after cross-validating the keywords extracted from the search query. This database table was then exported and used as the dataset for pre-processing. These query-related data points created for two search terms were combined and used as training dataset with two classes (oncology and neuroscience respectively). Implementation of machine learning algorithms in WEKA [13] was used to perform the data mining tasks. In order to test the classification algorithms we used 10-fold cross validation method as it allowed higher accuracy than percentage split for our NLP data set. Test data randomly chosen from the dataset before training showed 100% test accuracy and hence testing was not set as an objective. To make sure that there is no rote learning involved, we attentionally assigned wrong class labels

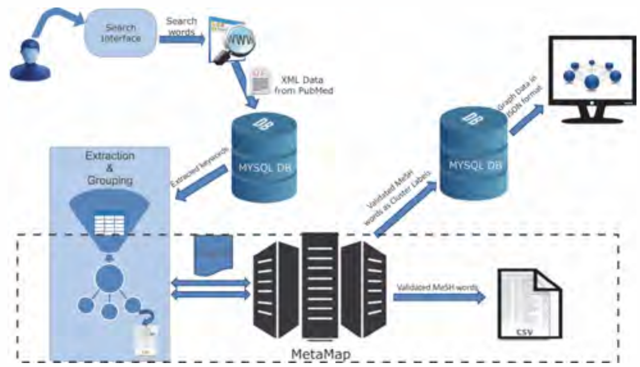

Fig. 1. Architecture of ANeuroNLP frame work. Dataset for this study was generated using the modules included in the rectangular box.

for 50% of instances and tested the accuracy of the same classifiers. We found out there only 50% accuracy of the classifiers.

Classification. There were total of 2000 instances for training, out of which 1000 for the search term "cerebellum function ataxia" and 1000 for "cerebellum physiology ataxia". The training example consisted of 10 attributes including the class label. Sample data points and features based on PubMed provided for the study are shown in Table I. We used 8classification algorithms to evaluate the candidacy role of such techniques.

AdaBoost. The NLP training examples were given to many weak learners and combined to a weighted sum that formed the output of the high performance prediction rule. Every weak learner that took individual instances as input and produced an output represented a class of the instance [14].

Naive Bayes calculated the probability for each attribute in the NLP data and predicted an unknown class label for the instances in the NLP data which had no class label based on Bayes' theorem $P(c|e) = P(c)P(e|c)/ P(e)$, where c is the class and e represents the given example [15][16].

LMT combined the logistic regression and tree structure to produce Model tree for the NLP data that contained linear regression functions at the leaves[17].

Vote algorithm defined many classification rules to classify the NLP data and combined them in such a way that the resulting classifier was superior to any of the individual rules. Each instance in the NLP data was then classified in to a class that receives majority of votes [18].

Stacking was the process of combining the individual decisions of many classifiers and used the resulted classifier to classify the new instance of the NLP data [19].

Random Forest algorithm combined the idea of bagging and random feature selection to construct a collection of decision

TABLE I
SAMPLE DATA SET

Pubmed_ID	MWordl	MWord2	MWord3	MWord4	MWord5	MWord6	MWord7	MWord8	Class label
15826993	Adult	Brain Stem	Physiopat hology	Cerebellar Diseases	diagnosis	Eye Movements	physiology	Humans	yes
16644229	Adult	Animals	Ataxia	Metabolism	Blotting	Pathology	Cerebellum	Blood Supply	no

trees with the mesh words. The resulting combination of decision trees was used to classify the NLP data[20].

Hoeffding Tree *u*sed Hoeffding bound for tree induction in order to replace confident attribute to split the tree. Trees for the NLP data were constructed by recursively replacing the leaves with decision nodes [21].

Input Mapped Classifier algorithm is a wrapper classifier that deals with the incompatibility in the NLP test and training data.

Clustering. Two datasets were employed for clustering. First data set was generated based on search term "cerebellum function ataxia". There were a total of 1000 instances with 9 attributes. Second dataset was generated with search terms "cerebellum function ataxia" and "cerebellum physiology ataxia". It consisted of 2000 instances and 9 different attributes. We measured the processing time, number of clusters and clustered instances for each algorithm.

EM Algorithm used the vector of extracted mesh words as the data input. EM assigned each NLP vector data to a cluster based on a weight representing the probability of membership[22].

K-Means algorithm divided the data points to K (K=2) partitions [23]. It iteratively took the mesh words obtained from MetaMapas 2 clusters and measured the distance between the mesh words and centroids to associate the mesh words with a centroid with maximum proximity. Centroids consisting of the mesh words had higher priority. Centroids were computed in every iteration until it converged minimizing the objective function.

FarthestFirst. Like K-Means, the first centroids was selected randomly but in farthest first method, consecutive centroids were selected as the farthest from the previous set of centroids. Remaining data points were assigned to different clusters with minimum distance between the obtained mesh words and the centroids [24].

FilteredClusterer applied different filters as clusters converged to optimal output[25].

Hierarchical Clusterer used partition clustering recursively on the selected mesh words. Different clusters were obtained by grouping connected nodes of the dendrogram after it had been cut at the desired level[26].

LVQ described a cluster based on parameters like center, size and shape. The distribution of NLP dataset was achieved with k reference vectors. These reference vectors were stored as weights in neural networks, which were updated for competitive learning [27]

Make Density Based Clusterer wrapped clusters to return a distribution and density. The method fit dataset as normal and discrete distributions within each cluster produced by a wrapped clusterer.

SOM consisted of layers of neurons that adapted to form clusters of NLP data. This adaption for the NLP dataset was achieved iteratively, determining the closest neuron for a specific input pattern and updating the neuronal weights. After several iterations, all neurons moved into the feature space where high concentrations of input patterns were observed[28].

In this paper, we also report the performance of these classification and clustering algorithms as part of evaluating the choice of NeuroNLP analytic process.

In order to test the robustness of the classifiers, we also performed the classification and clustering analysis by reducing the dataset size to 153.

IV. RESULTS

The accuracy of clustering and classification of neuroNLP data through different algorithms was estimated. The study involved classification of training data generated using two search terms in ANeuroNLP tool namely, "cerebellum function ataxia" and "cerebellum physiology ataxia", clustering of dataset generated by using the search term, "cerebellum physiology ataxia" and clustering of dataset generated using two search terms "cerebellum function ataxia" and "cerebellum physiology ataxia".

A. Classification Accuracy

With different classification algorithms, we recorded error rate, processing time and correctly and wrongly classified instances. By analyzing the accuracy of different classification algorithms for NeuroNLP datasets, we found that Naïve Bayes classifier, Hoeffding Tree and Random Forest techniques performed equally with high classification accuracy which classified 100% of the instances correctly. LMT classified 98.4% of the instances correctly and AdaBoost classified 75.75% of the instances correctly. Stacking, Vote and Input Mapped Classifier classified only 50% of the instances correctly with high root mean squared error (0.5). However the root mean squared error for Naïve Bayes and Hoeffding Tree was 0.0063 and for AdaBoost, LMT and Random forest, it was 0.36, 0.12 and 0.14

respectively. LMT algorithms took 109.07 seconds for processing the NLP data whereas Naïve Bayes, AdaBoost, Hoeffding Tree and Random Forest took 0.01, 0.16,0.46 and 1.22 seconds respectively. It was found that the Hoeffding Tree, Random Forest and Naïve Bayes classifier provided high accuracy than the remaining algorithms for the NLP data (See Table II).

Analysis shows that Naïve Bayes, Hoeffding Tree and Random Forest performed better in terms of accuracy than AdaBoost and LMT. Input Mapped Classifier, Stacking and Vote algorithms showed lesser accuracy (see Fig. 2).

TABLE II
CLASSIFICATION RESULT

Measures → Algorithm ↓	Time Taken (s)	Correctly classified Instances	Incorrectly classified Instances	Mean absolute Error	Root Mean Squared Error
AdaBoost	0.16	1515	485	0.2967	0.3644
Naïve Bayes	0.01	2000	0	0.0004	0.0063
InputMapped Classifier	0	1000	1000	0.5	0.5
LMT	109.07	1968	32	0.0356	0.1223
Stacking	0.02	1000	1000	0.5	0.5
Vote	0	1000	1000	0.5	0.5
HoeffdingTree	0.46	2000	0	0.0004	0.0063
RandomForest	1.22	2000	0	0.1085	0.1497

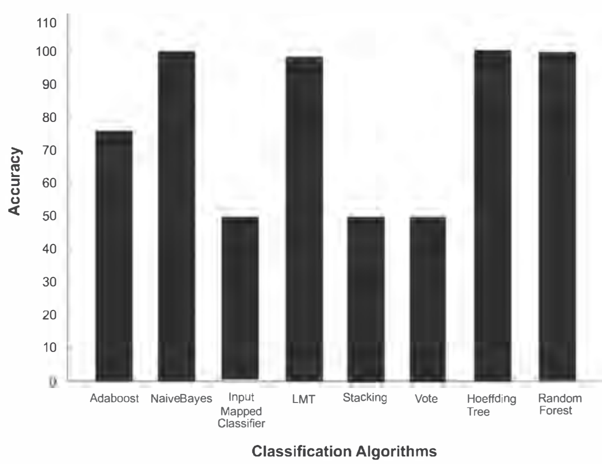

Fig. 2. Accuracy of classification algorithms based on correctness in classification.

B. Clustering Output

Initial clustering test was performed on a data set for the single query term "cerebellum function ataxia". Result shows that all algorithms mapped the NLP data into two clusters. EM algorithm clustered the instances with maximum accuracy by mapping 55% of instances in one cluster and 45% of instances in other class. Filtered Clusterer, K-means and MakeDensityBasedClusterer performed equally and it grouped the NLP data into two groups with 49% of data in one group and 51% of data in other group. Farthest First organized the data by grouping 60% of data into one cluster and 40% of the data in other cluster and took 0.01 second for process the data. SOM produces two clusters with 82% of data into one group and 18% into other group. K-means, Filtered Clusterer and Farthest First took 0.01 seconds for processing the data. EM and Make Density Based Clusterer took 3.52 and 0.03 seconds respectively whereas Self Organizing Map took 5.4 seconds. Two other clustering algorithms namely Hierarchical Clusterer and LVQ when used with same data sets gave high errors(see Table III).

To evaluate if clustering was efficient, another clustering was performed on the combined dataset generated for two search terms "cerebellum function ataxia" and "cerebellum physiology ataxia". It was found that all the algorithms produced two clusters even when the size of the data set was increased. To see the expected, we combined the dataset with 1000 instances for each search term. EM technique produced two clusters with exact 50% of instances. Filtered Clusterer, K-means and Make density Based Clusterer performed equally by clustering 46 % of the instances in one cluster and 54% of instances in other cluster by taking 0.03, 0.02 and 0.03 seconds respectively. Farthest First

TABLE III
CLUSTERING OUTPUT(FOR DATA SET GENERATED FOR SEARCH TERM 'CEREBELLUM FUNCTIONAL ATAXIA')

Measures → Algorithm ↓	Time Taken (s)	No. of Clusters	Clustered Instances
EM	3.52	2	0 554 (55%) 1 446 (45%)
FarthestFirst	0.01	2	0 598 (60%)1 402 (40%)
FilteredClusterer	0.01	2	0 490 (49%) 1 510 (51%)
K-means	0.01	2	0 490 (49%) 1 510 (51%)
MakeDensityBased Clusterer	0.03	2	0 490 (49%) 1 510 (51%)
SelfOrganizingMap	5.4	2	0 822 (82%) 1 178 (18%)

algorithm grouped 41% of instances in one cluster and 51% in other cluster with processing time 0.01 seconds. Self Organizing Map clustered the NLP data by grouping 84% of instances as one cluster and 16% as other cluster and its processing time was 10.81 seconds. For this dataset also, Hierarchical Clusterer and LVQ produced insignificant results (See Table IV).The difference in performance of clustering algorithms for the datasets generated for single and multiple search terms is shown in Fig. 3.

C. Method Validation

Since results indicated high classification accuracy we also tested effect of small datasets which was derived by extracting it from ANeuroNLP with two different search terms. Out of 153 instances in this dataset 78 instances were related to "cancer oncovirus fibroblast" and 75 to "cerebellum granule neuron ataxia". These experiments also have shown similar results with 14 % classification error and 2% to 16% grouping error. In order to see the influence of rote learning in the classification processes, we manually assigned wrong class labels for 1000 randomly selected training instances which showed proportional decline of 50% accuracy in the results.

V. DISCUSSION

In this study, we used post-processed NeuroNLP data by reverse mapping the extracted keywords with the mesh words obtained from MetaMap. The study, in order to improve better mining of NeuroNLP datasets, was conducted in two phases as classification and clustering of NLP data.

TABLE IV
CLUSTERING OUTPUT(FOR DATA SET GENERATED FOR SEARCH TERMS 'CEREBELLUM FUNCTION ATAXIA' AND 'CEREBELLUM PHYSIOLOGY ATAXIA')

Measures → Algorithm ↓	Time Taken (s)	No.of Clusters		Clustered Instances
EM	5.77	2	0	1002 (50%)
			1	998 (50%)
FarthestFirst	0.01	2	0	1028 (51%)
			1	972 (49%)
FilteredClusterer	0.01	2	0	910 (46%)
			1	1090 (54%)
K-means	0.02	2	0	910 (46%)
			1	1090 (54%)
MakeDensityBased Clusterer	0.03	2	0	922 (46%)
			1	1078 (54%)
SelfOrganizingMap	10.81	2	0	1679 (84%)
			1	321 (16%)

Out of eight algorithms used for classification, Hoeffding Tree, Random Forest and Naïve Bayes performed well for the NLP data however the performance of LMT and AdaBoost algorithms were moderate. Even though Stacking, Vote and Input Mapped Classifier were faster, their classification accuracy was unreliable.

Random Forest, Hoeffding Tree and Naïve Bayes performed equally but their error rates were different. On reverese mapped datasets unlike standard text mined datasets, classification results showed Hoeffding Tree, Random Forest and Naïve Bayes classifier provided higher classification accuracy than rest of the algorithms.

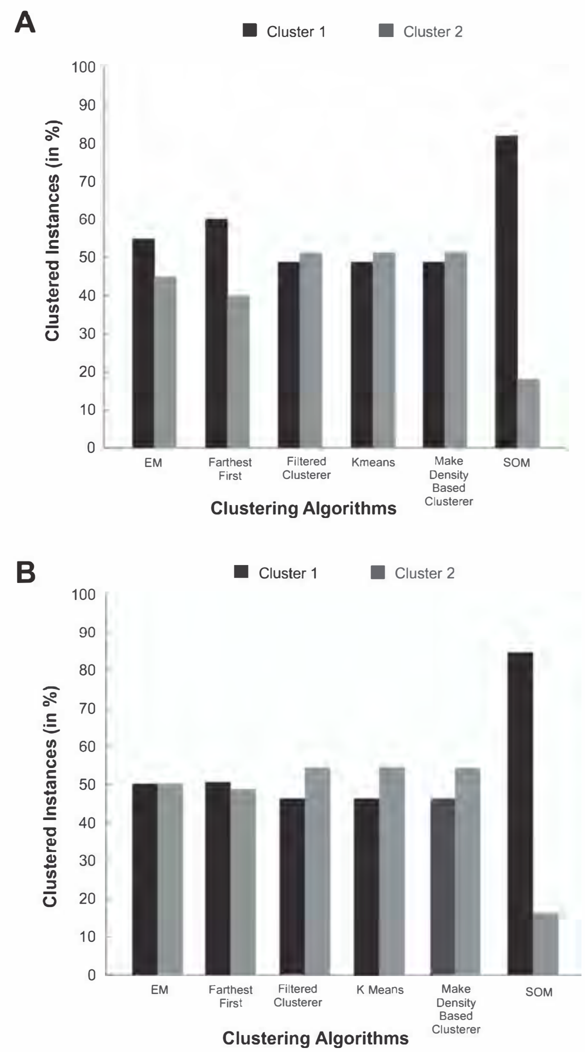

Fig.3. Clustering of NLP datasets. A.Clustering on dataset generated for search term "cerebellum function ataxia". B.Clustering on dataset generated for search terms "cerebellum function ataxia" and "cerebellum physiology ataxia".

With the first dataset generated for single search term "cerebellum function ataxia", EM algorithm provided high accuracy. Filtered Clusterer, K-means and Make Density Based Clusterer performed equally. Filtered Clusterer and K-means took equal time for processing the data. The second dataset was a combination of MeSH words for two search terms such as "cerebellum function ataxia" and "cerebellum physiology ataxia". Clustering results on this dataset showed that all algorithms grouped instances of NLP data into two clusters. EM algorithm clustered the instances with high accuracy. Filtered Clusterer, K-means and Make Density Based Clusterer had performed equally with reasonably less grouping error. For these combined dataset, SOM took more processing time than all other algorithms. It was observed that processing time increased with increase in the size of dataset. This study is significant as it indicates higher accuracy for mapped datasets rather than unmapped NeuroNLP data. Having used both techniques, as a choice of NeuroNLP datasets, we feel clustering is more appropriate.

VI. CONCLUSION

In our study related to improvising the quality of NLP datasets for clustering and classification, it may be suggested that pre-processing the data sets by reverse mapping the set to the UML metathesaurus from MetaMap helps in improving the quality of mined data and their inter-relations. Our studies suggest document clustering was favored over text classification for NeuroNLP datasets. We are extending this study to incorporate clustering into our ANeuroNLP platform for Neuroinformatics.

Acknowledgment

This project derives direction and ideas from the Chancellor of Amrita University, Sri Mata Amritanandamayi Devi. The study is partially funded by DST grant SR/CSRI/61/2014 and by MHRD Sakshat mission Virtual Labs Phase II, Government of India.

References

[1] S. D. Bay, D. Kibler, M. J. Pazzani, and P. Smyth, "The UCI KDD archive of large data sets for data mining research and experimentation," *ACM SIGKDD Explor. Newsl.*, vol. 2, no. 2, pp. 81–85, Dec. 2000.

[2] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: historical perspective and recent advances.," *J. Am. Med. Inform. Assoc.*, vol. 17, no. 3, pp. 229–36, Jan. 2010.

[3] J. W. Lee, J. B. Lee, M. Park, and S. H. Song, "An extensive comparison of recent classification tools applied to microarray data," *Comput. Stat. Data Anal.*, vol. 48, no. 4, pp. 869–885, Apr. 2005.

[4] M. M. S. Shukla and K. R. Rathod, "Stream Data Mining and Comparative Study of Classification Algorithms," vol. 3, no. 1, pp. 163–168, 2013.

[5] S. Chakraborti, "A Comparative Study of Performances of Various Classification Algorithms for Predicting Salary Classes of Employees," vol. 5, no. 2, pp. 1964–1972, 2014.

[6] S. B. Aher and L. M. R. J. Lobo, "Comparative study of classification algorithms," vol. 5, no. 2, 2012.

[7] H. Liu, J. Li, and L. Wong, "A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns.," *Genome Inform.*, vol. 13, pp. 51–60, 2002.

[8] A. Sravani, D. N. D. Harini, and D. L. Bhaskari, "A Comparative Study of the Classification Algorithms Based on Feature Selection," *ICT Crit. Infrastruct. Proc. 48th Annu. Conv. Comput. Soc. India- Vol I*, vol. 249, pp. 97–104, 2014.

[9] D. Pinto, P. Rosso, A. Juan, and H. Jiménez, "A Comparative study of Clustering algorithms on Narrow-Domain abstracts," *Proces. del Leng. …*, no. 2004, 2006.

[10] A. Rangrej, S. Kulkarni, and A. V. Tendulkar, "Comparative study of clustering techniques for short text documents," in *Proceedings of the 20th international conference companion on World wide web - WWW '11*, 2011, p. 111.

[11] B. Chaudhari and M. Parikh, "A Comparative Study of clustering algorithms Using weka tools," vol. 1, no. 2, pp. 154–158, 2012.

[12] T. C. Sharma and M. Jain, "WEKA Approach for Comparative Study of Classification Algorithm," vol. 2, no. 4, pp. 1925–1931, 2013.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, p. 10, Nov. 2009.

[14] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost-teaching AdaBoost to generalize better," *Graphicon*, 2005.

[15] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, "Naive Bayes Classification of Uncertain Data," in *2009 Ninth IEEE International Conference on Data Mining*, 2009, pp. 944–949.

[16] K. P. Murphy, "Naive Bayes classifiers Generative classifiers," pp. 1–8, 2006.

[17] N. Landwehr, M. Hall, and E. Frank, "Logistic Model Trees," *Mach. Learn.*, vol. 59, no. 1–2, pp. 161–205, May 2005.

[18] G. James, "Majority vote classifiers: theory and applications," no. May, 1998.

[19] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos, "Stacking classifiers for anti-spam filtering of e-mail," pp. 44–50, Jun. 2001.

[20] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32.

[21] H. Yang and S. Fong, "Evaluating hoeffding tree algorithm in real-time web applications environment." IMT CASE Journal (http://imtcj.ac.in/).

[22] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. …*, vol. 39, no. 1, pp. 1–38, 1977.

[23] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[24] M. Kumar, "An optimized farthest first clustering algorithm," in *2013 Nirma University International Conference on Engineering (NUiCONE)*, 2013, pp. 1–5.

[25] S. Revathi and T. Nalini, "Performance comparison of various clustering algorithm," *Int. J. Adv. Res. …*, vol. 3, no. 2, pp. 67–72, 2013.

[26] G. Carlsson and F. Mémoli, "Characterization, Stability and Convergence of Hierarchical Clustering Methods," *J. Mach. Learn. Res.*, vol. 11, pp. 1425–1470, Mar. 2010.

[27] C. Borgelt, D. Girimonte, and G. Acciani, "Learning vector quantization: cluster size and cluster number," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, 2004, vol. 5, pp. V–808–V–811.

[28] S. Günter and H. Bunke, "Self-organizing map for clustering in the graph domain," *Pattern Recognit. Lett.*, vol. 23, no. 4, pp. 405–417, Feb. 2002.