International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No. 6 (Nov-2020)
http://dx.doi.org/10.12785/ijcds/0906014

Categorisation of Computer Science Research Papers using Supervised Machine Learning Techniques

Hemrajsingh Gheeseewan[1] and Sameerchand Pudaruth[1]

[1]*Department of Information and Communication Technologies, University of Mauritius, Mauritius*

Received 21 Mar. 2020, Revised 14 May. 2020, Accepted 1 Aug. 2020, Published 1 Nov. 2020

Abstract: In this modern era of bleeding-edge technologies, information creation, sharing and consumption are rising at an exponential rate. In the same vein, there has been a continued increase in the amount of research is are being published worldwide and a large proportion of them are in the computer science field. There is an urgent need to provide some level of order in this huge jungle of data. Thus, in this article, we have used eight supervised machine learning techniques to classify computer science research papers. Machine learning techniques, such as logistic regression, multinomial naive bayes, gaussian naive bayes, support vector machines, k-nearest neighbours, decision tree, random forest and deep learning neural networks were trained to classify research papers into appropriate categories. For this purpose, a labelled dataset of 69776 papers was downloaded from arXiv and these were classified into 35 categories. The best f1-score of 0.60 was obtained by the logistic regression classifier. It was also the fastest machine learning classifier. The best f1-score from the deep learning network was 0.59. Using only the list of references for classification produced an f1-score of 0.57, but the training and testing time was significantly less. This shows that it is possible to use only references to classify computer science research papers. The f1-score for abstracts only was 0.52. Computer science papers often do not fall into neat categories. They are often multi-topical. Thus, in the future, we intend to perform multi-label classification on the same dataset.

Keywords: Document Classification, Computer Science, Machine Learning, Logistic Regression, Deep Learning

1. INTRODUCTION

The continued and relentless digitisation of the society has led to a massive increase in the volume of data that are being produced and this is increasing exponentially year after year. Such data can generally be categorised as either structured or unstructured data. Structured data are data that has a fixed format and are usually stored in electronic databases. Such databases can be easily queried to get relevant information. Structured data requires simple and straightforward search algorithms to be retrieved due to its predictable structure [1]. On the other hand, unstructured data can be generated by humans through text files, emails, social media posts, satellite footage, surveillance footage, but also from sensors [2][3].

White et al. showed that the number of academic publications worldwide almost doubled from 1.3 million to 2.3 million from 2004 to 2014 [4]. The United States of America (USA) and China are at the top of the list with 19% and 17% of the world's total, respectively [4]. Publications in the field of Computer Science are ranked fifth. They account for 8.9% of all research publications. Hänig et al. stipulated that new text mining techniques must be developed to extract intelligence, share information and deliver value from unstructured data as these data cannot be analysed, visualised or sorted in the same way that structured data is processed [5]. Text document classification is the procedure of allocating textual documents to one or more classes or categories by constructing a model through training data. An abundance of supervised machine learning approaches exists, namely logistic regression (LR), k-nearest neighbour (KNN), support vector machines (SVM), decision tree (DT), random forests (RF), naive Bayes (NB), artificial neural network (ANN) and deep learning networks (DNN). Using a dataset of 69776 computer science research papers, which were downloaded from arXiv, we were able to classify them into thirty-five categories with an f1-score of 0.60. Logistic regression was found to be the best classier, followed closely by deep learning networks.

The structure of this paper is organised as follows. Section 2 presents a background on document classification and machine learning classifiers. The literature review is described in Section 3. Section 4 consists of the methodology. Section 5 includes how the classification systems have been implemented, evaluated and tested. The paper comes to its conclusion in Section 6.

E-mails: hemrajsingh.gheeseewan@umail.uom.ac.mu, s.pudaruth@uom.ac.mu

2. BACKGROUND STUDY

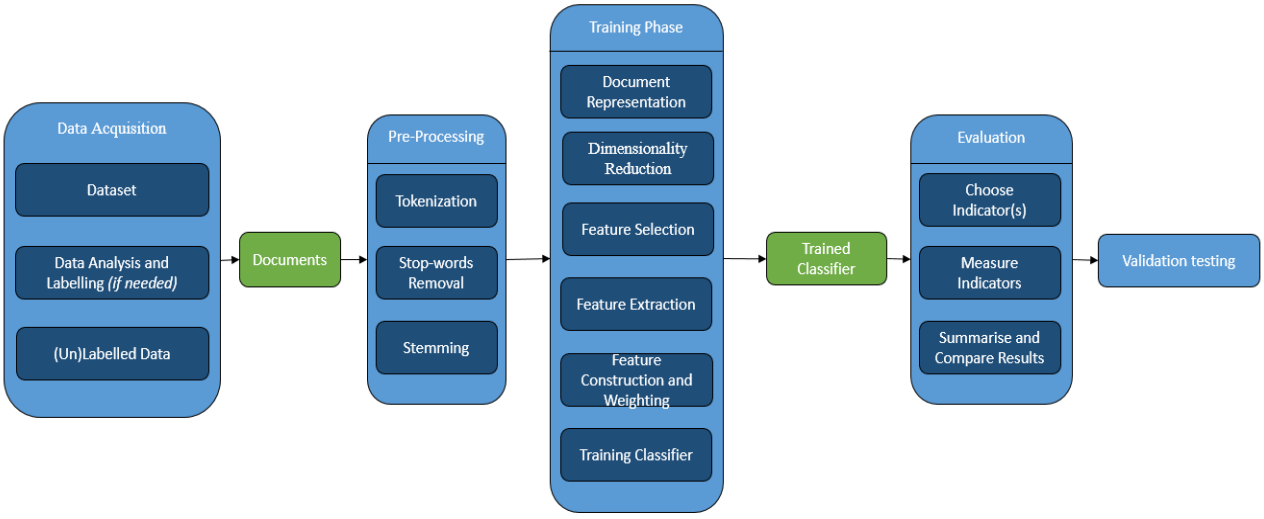

Figure 1. Document classification pipeline

A. The Classification Process

A document classification pipeline is shown in Figure 1. It usually consists of the following steps [6].

- **Data acquisition**

This is the first step of any machine learning problem. The data required to solve the problem needs to be gathered. This can be in the form of raw texts, pictures, videos and may come from any source. This process is not so simple as there are a lot of external factors to take into consideration, namely the size of the dataset, quality of the data, whether the data is labelled or not, accessibility, time required for acquiring the data, storage requirements or even monetary constraints.

- **Data pre-processing**

So as to not hide meaningful patterns, which would lead to redundancy and low performance of the classifiers in the analysis, pre-processing is used to eliminate unrelated strings [7]. In written English, terms (numbers, punctuation marks, words, tags, and other symbols like emoticons) are usually separated by spaces. Irrelevant information to the classification problem like punctuations and numbers are removed, but sometimes these may represent meaningful information and they must be retained like exclamation marks or emoticons [8]. As explained by Kowsari et al., tokenisation is the process in which each and every word of a sentence is separated and considered as a token [9]. The removal of stop-words can make text files lighter and therefore easier and faster to process. Stop-words usually have no real importance in defining what the text is about. Durairaj and Karthikeyan explained that the process of stemming needs to follow two points [10]. Firstly, words with different meanings should be separately stored and secondly, the different forms of the same word should be mapped to the same stem as they are assumed to bear identical meaning [10]. An example is shown in Figure 2. Lemmatisation can be used instead of stemming if the original base form needs to be recovered. Lemmatisation is usually harder than stemming and may add more unwarranted complexity to the problem [11][12].

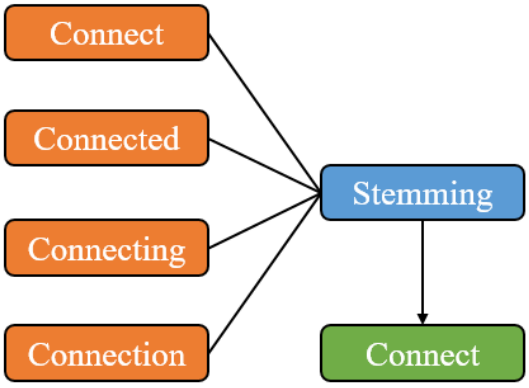

Figure 2. Stemming [10]

- Training phase

N-grams are considered as an important feature for text representation and text classification [13]. For instance, there are 1-grams (unigrams), 2-grams (bigrams) and 3-grams (trigrams), where the numbers indicate the number of successive words that are processed together as a single term (feature) in order to retain more meaning. Bag-of-words approaches typically use 1-grams as features to represent text, while bag-of-terms approaches use a combination of n-grams. According to Silva and Cunha, n-grams do not produce major improvements in the classification performance, but the actual impact on often dependent on the application [14].

Dimensionality reduction can be considered as a compression of the data. It picks the features that are most important, for example, by removing the irrelevant words. Since we have less data to process, this often results in less training time. The feature selection methods remove everything except the most relevant and descriptive features or dimensions. For example, for a classification problem for research papers, the various features can be publication year, author(s), keywords extraction, abstract, and references. After selecting the features, they are extracted from the data. Several combinations of features can be extracted to compare and contrast each of them.

During this process, the ML algorithm chosen will use the set of labelled data from the previous step and put them in a form that can be represented and weighted. Kobayashi et al. refer to this process as text transformation [8]. A vector of feature weights is used to represent a document, where the features consist of the words (terms, phrases) in the document. Weighting schemes, such as Binary Term Frequency and Simple Term Frequency are usually used.

- Model training

Once the above stages are completed, the documents are now in a format which is suitable for further processing. The dataset is split into a training and a testing set and fed to the classifier. The training set is usually much larger than the testing set. It is usually better to provide the classifier with more of the existing data during training so that it has more data to learn from.

- Evaluation

Indicators are the metrics used to evaluate the performance of a classifier, such as accuracy, precision, f1-score, recall, error rate, memory allocated to model, CPU time to create model, etc. All the indicators measured are assessed and contrasted to each other using methods, such as various plots for visualisations. In cross-validation, the dataset is split into k sets [15]. The (k-1) sets are used to train the classifier, while the k^{th} set is used for testing, as shown in Figure 3. The scores of this particular split are recorded. This is repeated until all the k sets have been used as test set. The mean scores of all the splits give the overall performance of the model.

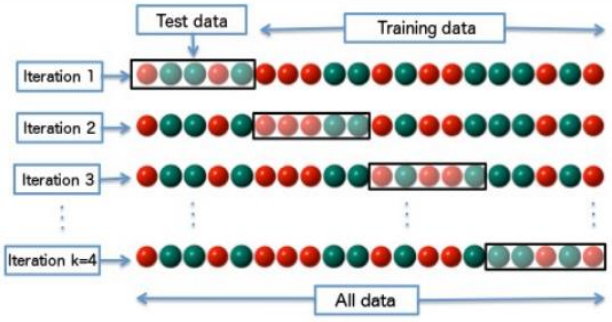

Figure 3. Cross-validation technique

- Validation

The model produced by the classifier is put to a second test on an unseen set of data. It is often seen that models do not have the same performance on testing data and on data that are used later during real-time use. This often occurs because of overfitting (or underfitting) because the model has been over-trained (under-trained) to understand the data. Therefore, to better understand the model and its performance, it is a good practice to validate it on an unseen set of test data. This usually gives a better idea of the model quality.

B. Text Classification and Text Classifiers

Shalev-Shwartz and Ben-David explained that machine learning is a subdivision of artificial intelligence and is one of the quickest growing fields of Computer Science [16]. The focus of ML is to endow programs with the capability to adapt and learn and find meaningful patterns automatically in data. The main goal of text classification is to help users gain information from data by using operations like summarisation, classification and retrieval [16].

- Model training

Using a training set of labelled data, a supervised machine learner constructs a model, which it uses to predict the classification of unlabelled texts. Supervised learning can be further divided into two approaches: parametric and non-parametric models [17]. Parametric models are based on the underlying parameters that can be summarised. The model uses a known form to represent a function. These models are usually simpler, quicker and use less data, but they cannot represent complex functions [17]. For example, linear classifiers have as goal to group similar feature values into groups and when the number of dimensions is large, it works really well [18]. One such example is logistic regression. Kobayashi et al. explained that these types of algorithms are based on the computation of probability between the documents and their classes and the classification for a document attained by finding the category that provides the highest probability, such as in naïve Bayes [8].

On the flip side, non-parametric models learn their functional form from the data provided and do not make strong assumptions. This allows these models to be more flexible and they usually perform better, but they tend to be time-consuming and require more training data [17]. K-nearest neighbour (KNN) and support vector machines (SVM) are examples of non-parametric and geometric classifiers, while decision trees and random forests (RF) are examples of non-parametric logical classifiers. Deep learning (DL), which is a part of ML, is based on brain functions in which neurons are the building blocks [19][20]. Deep learning neural network is also a non-parametric machine learning classifier.

3. LITERATURE REVIEW

Ramesh et al. provided an overview of research in computer science and studied 628 computer science papers from 13 journals [21]. They identified five main categories, namely: problem-solving concepts, computer concepts, systems/software concepts, data/information concepts, and problem-domain-specific concepts. From their findings, they found that most journals normally have one of these categories associated with them. Each of these categories was then divided into various sub-fields of computer science. Moreover, they opined that a lot of research in computer science was focused on mathematical concepts and logic. All the classifications were done manually as the intention was to understand the field and help others to understand it as well by bringing some order to it.

Mirończuk and Protasiewicz provide a general overview of the text classification field for the last ten years [6]. From a dataset of 233 articles, they observed that $2/5^{th}$ dealt with feature selection, construction analysis and projection. One-fifth of articles were dedicated to learning methods, followed by another $1/5^{th}$, which consisted of document representation, resource selection, and evaluation. The remaining $1/5^{th}$ dealt with classification systems and applications areas. The authors also observed that areas, such as multi-lingual classification, cross-lingual classification, text stream analysis are gaining in popularity.

Osisanwo et al. described and compared 7 supervised ML methods, namely decision table (DT), neural networks (NN), naïve Bayes, SVM, JRip, decision trees and random forests [18]. They use the Pima Indians dataset and achieve the best classification accuracy of 77.3 with SVM. Mowafy et al. (2018) used 19997 documents divided into 20 different categories with a training/testing split of 60%/40% to test the model. The authors concluded that Multinomial Naïve Bayes (MNB) is better than KNN and that chi-square can improve the performance of other machine learning classifiers.

Zhou et al. gathered 80,000 Computer Science research papers and another 80,000 non-computer science papers from arXiv after random sampling and removing duplicates [22]. The methods used were Multinomial Naïve Bayes (MNB) on unigrams and bigram models, and MNB, logistic regression (LR) on vector representation generated using sentence2vec. The best f1-score of 0.95 was obtained with MNB on the bigram language model. The removal of stop-words improved the f1-score in all models, but the effects of stemming were insignificant.

In their work, Al-Harbi et al. mainly dealt with the classification of Arabic texts [23]. The steps involved were the collection of text documents and then labelling them. The documents were categorised into seven classes, namely: Saudi Press Agency (SPA), Saudi News Papers (SNP), web sites, writers, discussion forums, Islamic topics, and Arabic poems. The documents from the 7 classes amounted to 17,658 with over 11,500,000 words. The number of documents in the dataset was not balanced. The writers' category had only 821 articles, while SMP had 4842 articles. The Arabic Text Classification (ATC) tool was used for splitting the dataset into training and testing sets and for feature extraction and selection [23]. The chi-square statistics were applied on document frequency to choose the top thirty terms of each class. 70% of the dataset was used for training and 30% for testing. The C5.0 (which is a type of decision tree) outperformed the support vector machines classifiers by a significant margin. Thus, C5.0 had an average accuracy of 78.42%, while the average accuracy for SVM was only 68.65%. C5.0 also had the best performance in all the seven categories. The best accuracy of 92.12% (C5.0) was obtained on the Islamic topics, while the worst accuracy of 49.15% (C5.0) was obtained on poems. As per the authors, this is due to the rich vocabulary of the Arabic language, which makes poems vastly different from each other [23].

Ting et al. aimed at classifying documents into four categories, namely: travel, business, sport and politics using the NB classifier [24]. A total of 4000 documents were gathered where each category had 1000 documents. A training set of 70% (2800 documents) and a testing set of 30% (1200 documents) were used. The data was vectorised using TF-IDF (term frequency-inverse document frequency). The f1-score for the raw dataset was 0.969, while the f1-score for the pre-processed dataset was 0.955. The authors concluded that the effects of pre-processing on classification accuracy are usually insignificant in classification problems, although significant improvements in the time required to build the models can be observed.

From the literature review, we have seen that using larger datasets generally leads to better classification accuracies. Larger feature sets also have a positive impact on performance. However, the effect of pre-processing and cleaning operations are application dependent. We also observed that most researchers had used a small number of categories in their studies. Thus, in this work, we intend to use as many as 35 different categories for the classification of computer science research papers. Most studies have dealt with news articles and general texts, but research on the classification of academic papers and in particular, computer science papers, are limited. We intend to address this gap in this research. Moreover, classification will be performed using different segments of the papers, such as abstracts and the list of references. However, a lot of research has been done on this aspect of text classification using machine learning, research on segments of computer science papers is scarce.

4. METHODOLOGY

A total of 69776 pdf files were downloaded from arXiv for this study. The categories are shown in Table 1. 124 files were found to be corrupted and were removed from the dataset. The remaining 69652 files were separated into a training/testing set of 66152 files and an unseen set of 3500 files. One-hundred documents from each category were selected to be included in this list.

TABLE I. CATEGORIES IN DATASET

Categories	No. of files
Computer Vision and Pattern Recognition	3776
Computation and Language	3027
Computational Complexity	2521
Machine Learning	2462
Information Theory	2401
Programming Languages	2378
Artificial Intelligence	2315
Sound	2303
Human-Computer Interaction	2295
Data Structures and Algorithms	2285
Information Retrieval	2259
Neural and Evolutionary Computing	2258
Distributed, Parallel, and Cluster Computing	2245
Social and Information Networks	2235
Discrete Mathematics	2230
Computational Geometry	2224
Cryptography and Security	2220
Robotics	2214
Networking and Internet Architecture	2210
Logic in Computer Science	2209
Computer Science and Game Theory	2197
Computers and Society	2184
Databases	2166
Software Engineering	2166
Multiagent Systems	1886
Formal Languages and Automata Theory	1566
Computational Engineering, Finance, and Science	1554
Multimedia	1462
Digital Libraries	1408
Graphics	1406
Performance	1208
Hardware Architecture	882
Mathematical Software	691
Symbolic Computation	677
Operating Systems	256

The unseen set is used for validating a trained model. It is crucial to understand how the model behaves on data that it has never seen before. The aim in this research was not only to see how machine learning can be used to classify computer science research papers, but also to assess whether it is possible to use only part of a research paper (segment) for classification. Thus, three different datasets were produced. The first one is the whole paper dataset and it consists of the raw papers with all the content when converting the pdf files into text files. The second dataset consists only of abstracts. The abstract is extracted from each research paper by using a simple heuristic rule, i.e., extract all content that lies between the words 'abstract' and 'keywords'. However, there were some difficulties with this approach, as not all papers are formatted in the same way. Some of these issues had to be resolved manually. To avoid the selection of irrelevant content, a word limit of 400 words was imposed on the abstract. The third dataset consists of references only. Again, a simple heuristic rule was used for the extraction of references, i.e. all content that came after the word 'references' (and before the word appendix, if present) were considered to be part of the references. Similarly, a word limit of 1000 words was imposed on this segment.

For each of the three datasets mentioned above, there are two pre-processed versions (PPP1 & PPP2) of the dataset. PPP1 was produced by tokenising the text and then removing all stop-words, single characters, digits, periods and dashes. PPP2 was produced from PPP1, but with two additional steps which were stemming and lemmatisation. A smaller dataset of 15,000 documents was produced from the whole paper's dataset. Fifteen hundred papers from the ten largest categories were randomly selected to be part of this dataset. The details of all these datasets are summarised in Table 2.

TABLE II. DATASET SUMMARY

Dataset version		No. of Files
Whole paper dataset	Raw	66152
	Pre-processed part 1	
	Pre-processed part 2	
References extraction dataset	Raw	
	Pre-processed part 1	
	Pre-processed part 2	
Abstract extraction dataset	Raw	
	Pre-processed part 1	
	Pre-processed part 2	
Sampled (balanced) dataset	Pre-processed part 2	15000
Unseen text files (whole papers + pre-processed part 2)		3500

The processing of most of the datasets (Table 2) follows the same set of steps, as shown in Figure 4.

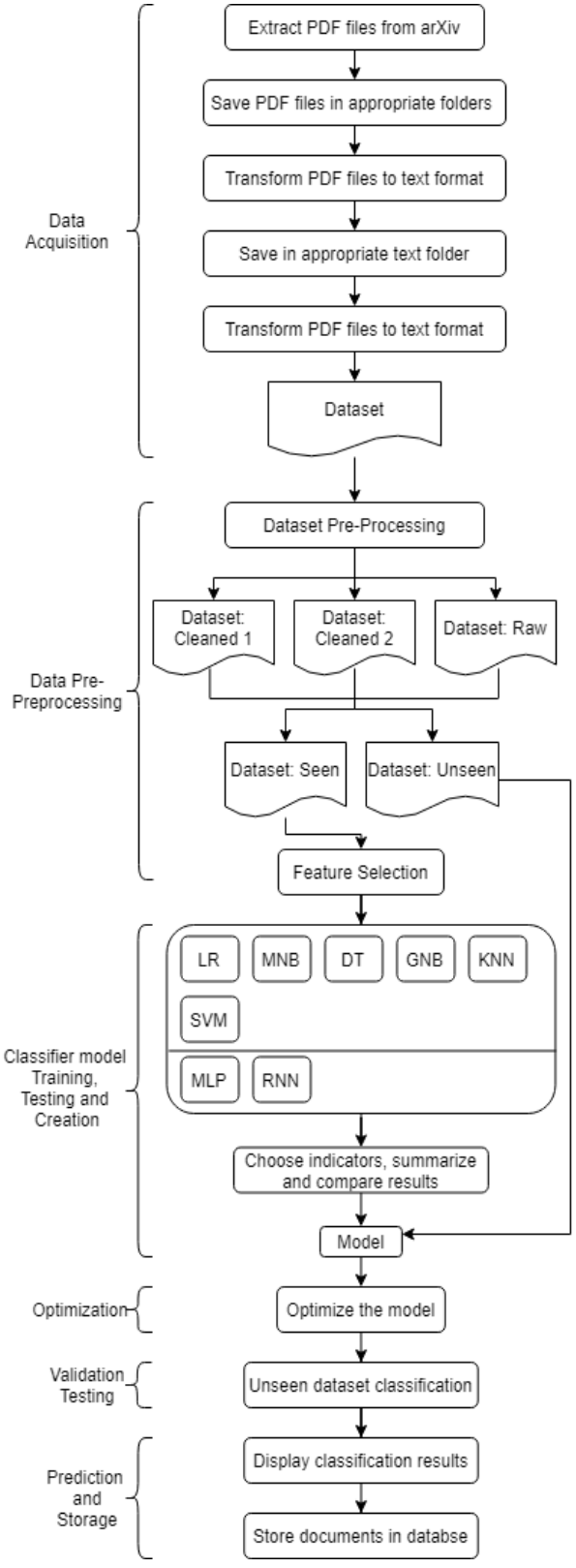

Figure 4. Proposed system model

5. Testing, Evaluation and Discussion

A. Performance of the classifiers on the raw dataset

Table 3 shows the accuracy for each classifier on the raw dataset, raw dataset + PPP1 and raw dataset + PPP2. In general, we can see that logistic regression (LR), support vector machines (SVM) and deep learning neural networks (DLNN) have very similar performances on all these three datasets. The best accuracy was 60% with each of these three classifiers. Decision trees had the worst performances. We can also conclude that cleaning operations, such as removal of stop-words, punctuations and other symbols, have little impact on the accuracy. The effect of stemming and lemmatisation is also negligible.

TABLE III. Accuracies of the Classifiers

Classifiers	Raw	PPP1	PPP2
LR	0.59	0.60	0.60
MNB	0.55	0.57	0.56
GNB	0.45	0.47	0.46
DT	0.31	0.34	0.35
SVM	0.58	0.60	0.60
KNN	0.52	0.52	0.51
MLP	0.51	0.54	0.54
DLNN	0.59	0.60	0.60

Because the datasets were not balanced, the f1-scores were also calculated, as shown in Figure 6. However, the values were found to follow a very similar trend to that of the accuracies. This is probably because there is a significant number of documents in each class, even if they are not balanced. The following parameters were used for the DLNN. The number of layers was set to 3 and the number of output nodes was set to 35. The vocabulary size was set to 1000 and the batch size was 512. Forty epochs were sufficient for the model to be fully trained, as shown in Figure 5. The trends for the three datasets were very similar.

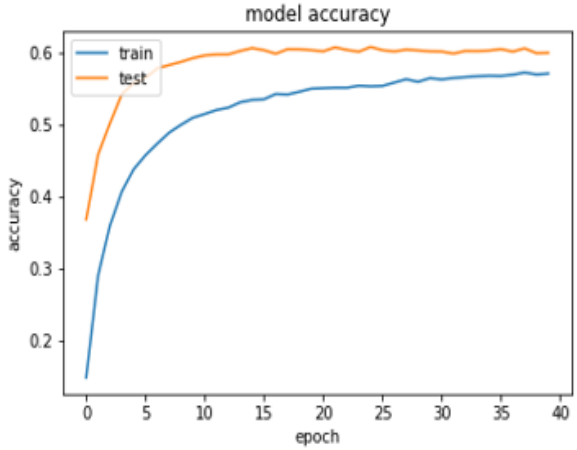

Figure 5. Training of DLNN on raw dataset + PPP2

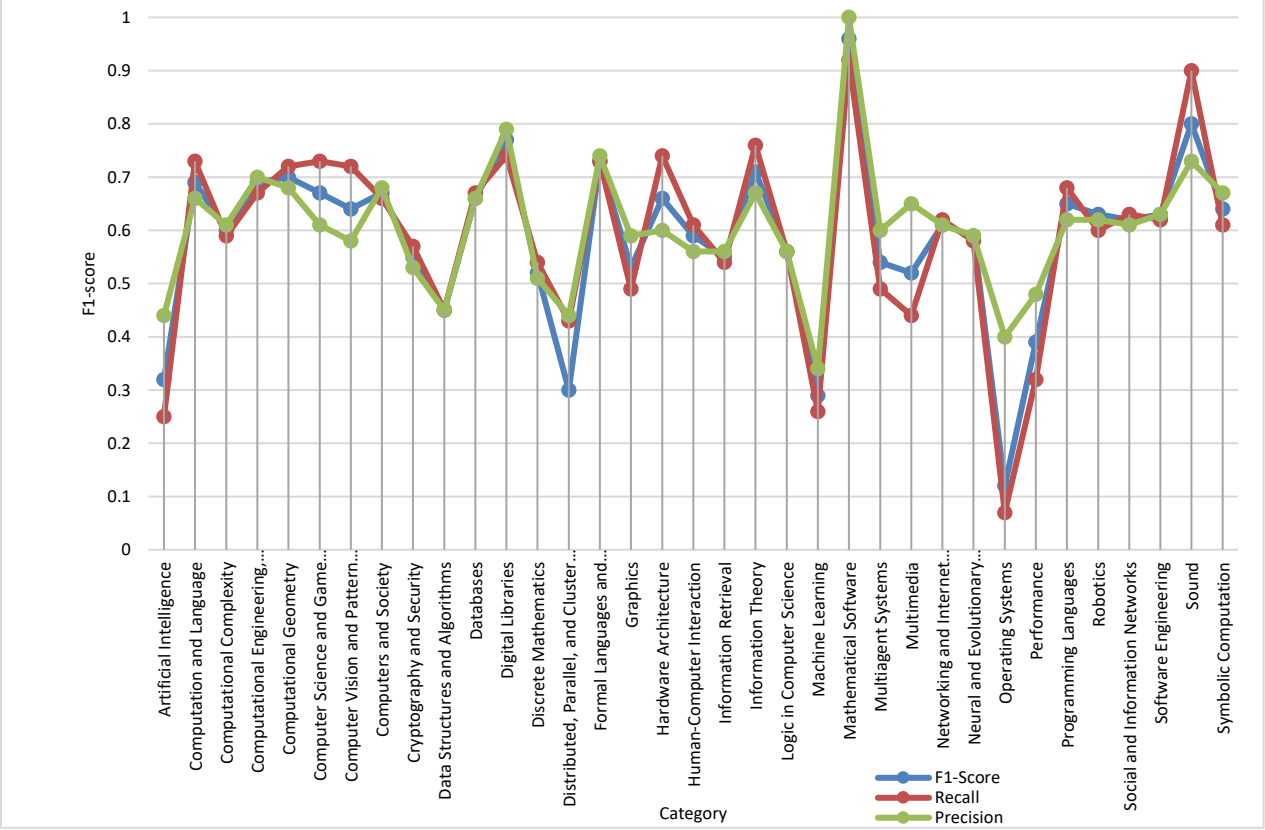

Figure 6. F1-scores for logistic regression for raw dataset + PPP2

B. *Performance of logistic regression*

Logistic regression has shown the highest score of 0.60 using the whole paper dataset and PPP2. Figure 6 shows a comparison of the f1-scores, recall and precision for each of the 35 classes. Mathematical Software (MS), Sound and Information Theory (IT) are the three best-classified categories with f1-scores of 0.96, 0.80, and 0.71, respectively. Operating Systems (OS), Machine Learning (ML) and Artificial Intelligence (AI) are the least well-classified categories with f1-scores of 0.12, 0.29, and 0.32, respectively. The scores for OS are very low because it had the smallest number of documents, while the scores for ML and AI are quite low because there is a lot of overlap between these two categories and others as well, such as Robotics and Neural & Evolutionary Computation. Other categories, such as Computer Science & Game Theory also has elements of ML and AI. Moreover, ML and AI can also be present in many other articles which are classified under categories, such as Computer Vision & Pattern Recognition, Symbolic Computations, Multiagent Systems, etc. There exist many links between the above categories and merging some of them could have been possible. Furthermore, these results open the door for multi-class classification, whereby a single document could be assigned to multiple categories instead of one class only.

C. *Feature selection with n-grams*

For whole papers, logistic regression had better performance in terms of both predictive accuracy and training time. This section focuses on improving the scores of the logistic classifier by performing further feature extraction and fine-tuning of parameters. Thus, besides the application of PPP1 and PPP2, n-grams were also used. The results are shown in Table 4.

TABLE IV. FEATURE SELECTION WITH N-GRAMS

Dataset \ Scores		1-gram		(1-2) grams		(1-3) grams	
		Ac	F1	Ac	F1	Ac	F1
Whole paper dataset	Raw	0.59	0.58	0.58	0.56	0.58	0.56
	PPP1	0.60	0.59	0.58	0.57	0.58	0.57
	PPP2	0.60	0.60	0.59	0.58	0.59	0.58
Abstract dataset	Raw	0.52	0.51	0.52	0.51	0.52	0.51
	PPP1	0.52	0.53	0.53	0.53	0.53	0.52
	PPP2	0.55	0.53	0.54	0.53	0.54	0.53
Reference dataset	Raw	0.58	0.57	0.58	0.57	0.58	0.58
	PPP1	0.59	0.57	0.58	0.57	0.58	0.57
	PPP2	0.58	0.57	0.58	0.58	0.58	0.57
Average for n-grams			0.56		0.55		0.55

From Table 4, we can see that the best results for the whole paper dataset were obtained with 1-grams (i.e. single words only) and PPP1 (or PPP2). The highest accuracy with references only was 59% and this was achieved with 1-grams and PPP1. The highest accuracy with abstracts only was 55%, but this was again achieved with 1-grams and PPP2. Thus, in general, we see that using two or three words in a single term does not bring any benefits in the classification of computer science papers. Stemming had some impact on the abstracts dataset, possibly because of the limited number of words in the abstract and most of the important words would have a very low frequency. Thus, it seems that when frequencies are low, using the roots of words can help to improve the classification accuracy by a small amount.

TABLE V. MODEL CREATION TIME FOR DATASETS

Datasets		**Size (MB)**	**Model Creation Time (Minutes)**
Whole paper dataset	Raw	3492	8.23
	PPP1	1935	4.58
	PPP2	1649	3.75
Reference dataset	Raw	583	1.80
	PPP1	386	1.82
	PPP2	336	1.44
Abstract dataset	Raw	132	1.08
	PPP1	96	0.79
	PPP2	83	0.72

Table 5 shows the times required to create the model for each of the three large datasets using the logistic regression classifier. Feature selection was done using 1-grams since we have seen that they were the most effective in the previous section. There is a very high correlation between the size of the files and the amount of time required to process them. Thus, the raw dataset with no pre-processing took 8 minutes 14 seconds to train, while the abstract dataset with pre-processing (PPP2) took only 43.2 seconds. Only the experiments were performed on a Window 10 machine with an Intel Core i7-6700 @3.40 Ghz, a 64-bit architecture and 16GB of memory.

D. Classification of the sampled (balanced) dataset

This dataset of 15,000 documents was created through the selection of 1500 documents from the top 10 categories with the highest number of articles. It was then trained and tested using an 80/20 split. The PPP2 version of this sampled (and balanced) dataset was tested using the logistic regression classifier using a cross-validation technique with 10 folds. Only the PPP2 version was used because on average it is the one which gave the best accuracies in all our previous experiments. The results for each fold is shown in Figure 7. The accuracy varied between 0.88 and 0.89 during each of these 10 folds and the mean accuracy was 88.4%. The purpose of this experiment was simply to show that it is possible to achieve a very high accuracy using our proposed techniques on computer science research papers provided that a small number of categories are used.

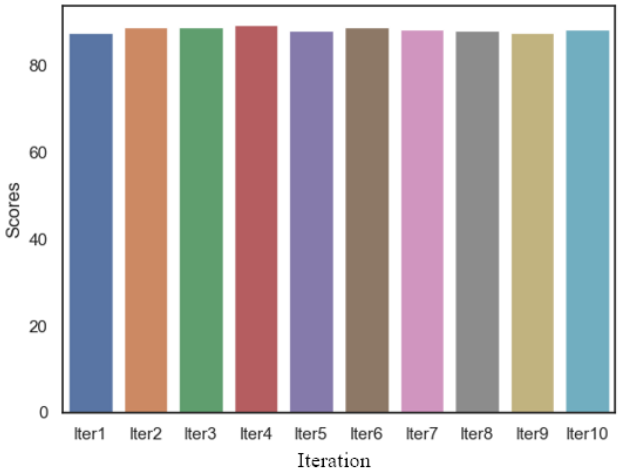

Figure 7. Sampled dataset with cross-validation

E. Validation using the Unseen dataset

The best model can be validated (second level of testing) by applying it on an unseen set of articles. As explained earlier, 100 articles were removed from each of the 35 categories to create a set of 3500 papers. These papers were not used in the training & testing phases. There were separated from the dataset of 69776 papers at the very start. Table 6 shows the accuracy on three dataset versions (those that produced the best accuracy during the training/testing phases). The accuracy on the PPP2 version of the whole dataset has dropped by only 2%. This shows that the model is robust and reliable and is therefore expected to perform equally well on new data if such a system is deployed. The drop is slightly more pronounced for the other two datasets. It is 5% for the reference dataset (+ PPP1) and 7% for the abstract dataset (+ PPP2). Thus, for our study, we can conclude that models with larger feature sets (whole documents) are more reliable than those with smaller feature sets (references and abstracts).

TABLE VI. ACCURACY FOR THE UNSEEN DATASET

Dataset	**Accuracy**
Whole dataset (PPP2 version)	0.58
Reference dataset (PPP1 version)	0.54
Abstract dataset (PPP2 version)	0.48

F. Comparison with related works

Zhou et al. had used only two categories with a total of 160,000 documents, which were equally divided into computer science and non-computer science research articles [22]. The best f1-score of 0.95 was achieved with MNB. However, differentiating between computer science papers and non-computer science papers is much easier than differentiating between the different fields of computer science. It is very difficult to categorise a computer science paper neatly into one category. Many articles are multi-categories and topics, such as machine learning, artificial intelligence, security, databases, etc., can be an important component of papers that are categorised as belonging to other areas. HaCohen-Kerner et al. classified 2082 computer science papers from three different conferences (ACL, SIGIR and AAMAS) belonging to different research domains with an accuracy of 92% using the CART decision tree from the Weka platform [25]. Their accuracy was very high because they had used a very small dataset with three categories only.

6. CONCLUSIONS

In this study, a labelled dataset of 69776 papers was gathered and classified into 35 categories using arXiv labels. A very large number of experiments were performed on different versions of the dataset. The logistic regression classifier proved to be the best classification algorithm in most of these experiments. It was also the fastest. The highest accuracy achieved was 60% after performing some cleaning operations and, followed by stemming. However, these pre-processing techniques had limited impact on accuracy. Our experiments show that using words only gave the best accuracies compared to using combinations of unigrams with bigrams and trigrams. The models were successfully validated on an unseen set of 3500 articles. To our knowledge, this is the first work that has attempted to classify computer science papers using both abstracts and references. This work shows that it is possible to classify computer science papers with a reasonable degree of accuracy using only the list of references. The experiments performed in this work can be used to support current or future studies in the area of text classification, whether in the field of computer science or other fields of research. In our future works, we intend to work on multi-class classification as we noticed from confusion matrices that many documents typically contain elements from several topics and thus, they cannot be conveniently categorised into one class only.

REFERENCES

[1] A.C. Eberendu, "Unstructured data: an overview of the data of big data," *International Journal of Computer Trends and Technology*, vol. 38, no. 1, pp. 46-50, 2016.

[2] A.A. Khandy and R. Miri, "uDCLUST: A novel algorithm for clustering unstructured data," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 5, no. 3, pp. 556-561, 2019.

[3] S. Feldman, J. Hanover, C. Burghard, and D. Schubmehl, "Unlocking the power of unstructured data," *IDC Health Insights*, pp. 1-10, 2012.

[4] K.E. White, C. Robbins, B. Khan and C. Freyman, "Science and engineering publication output trends: 2014 shows rise of developing country output while developed countries dominate highly cited publications," *National Center for Science and Engineering Statistics (NCSES)*, NSF 18-300, 2017.

[5] C. Hänig, M. Schierle, and D. Trabold, "Comparison of structured vs. unstructured data for industrial quality analysis," In: *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, USA, October 20-22, 2010.

[6] M.M. Mirończuk, and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Systems with Applications*, vol. 106, pp. 36-54, 2018. doi:10.1016/j.eswa.2018.03.058

[7] A.K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing and Management*, vol. 50, no. 1, pp. 104-112, 2014. doi:10.1016/j.ipm.2013.08.006

[8] V.B. Kobayashi, S.T. Mol, H.A. Berkers, G. Kismihók and D.N. Den Hartog, "Text classification for organizational researchers: A tutorial," *Organizational Research Methods*, vol. 21, no. 3, pp. 766-799, 2018. doi:10.1177/1094428117719322

[9] K. Kowsari, K.J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, 2019. doi: 10.3390/info10040150

[10] D.M. Durairaj and A. Karthikeyan, "Efficient hybrid machine learning algorithm for text classification," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 6, pp. 680-688, 2017.

[11] M. Toman, R. Tesar, and K. Jezek, "Influence of word normalization on text classification," In: *Proceedings of the International Conference on Information Sciences and Technology, Merida, Spain*, vol. 4, pp. 354-358, 2006.

[12] Y. HaCohen-Kerner, D. Miller and Y. Yigal, "The influence of preprocessing on text classification using a bag-of-words representation," PLoS ONE, vol. 15, no. 5. doi:10.1371/journal.pone.0232525

[13] R. Jindal, R. Malhotra and A. Jain, "Techniques for text classification: Literature review and current trends," *Webology*, vol. 12, no. 2, 2015.

[14] J.F. Silva and J.C. Cunha J.C., "An empirical model for n-gram frequency distribution in large corpora," In: *Lauw H., Wong RW., Ntoulas A., Lim EP., Ng SK., Pan S. (eds) Advances in Knowledge Discovery and Data Mining*. Lecture Notes in Computer Science (Springer), vol 12085. doi.org: 10.1007/978-3-030-47436-2_63

[15] Z. Chen, L.J. Zhou, X.D. Li, J.N. Zhang and W.J. Huo, "The Lao text classification method based on KNN," *Procedia Computer Science*, vol. 166, pp. 523-528, 2020. doi:10.1016/j.procs.2020.02.053

[16] S. Shalev-Shwartz and S. Ben-David, "Understanding machine learning: From theory to algorithms," *Cambridge University Press*, 2014. doi:10.1017/CBO9781107298019.

[17] D. Gueho, P. Singla, R.G. Melton and D. Schwab, "A comparison of parametric and non-parametric machine learning approaches for the uncertain Lambert problem," In: *Proceedings of the AIAA Scitech Forum*, Orlando, Florida, USA, January 5-10, 2020.

[18] F.Y. Osisanwo, J.E.T. Akinsola, O. Awodele, J.O. Hinmikaiye, O. Olakanmi and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48 no. 3, pp. 128-138, 2017. doi:10.14445/22312803/IJCTT-V48P126

[19] A. Elnagar, R. Al-Debsi and O. Einea, "Arabic text classification using deep learning models," *Information Processing & Management*, vol. 57, no. 1, 2019. doi:10.1016/j.ipm.2019.102121

[20] S. Kiranyaz, T. Ince, A. Iosifidis and M. Gabbouj, "Operational neural networks," *Neural Computing and Applications*. doi:10.1007/s00521-020-04780-3

[21] V. Ramesh, R.L. Glass and I. Vessey, "Research in computer science: An empirical study," *Journal of Systems and Software*, vol. 70, pp. 165-176, 2002.

[22] Zhou, T., Zhang, Y. and Lu, J, "Classifying computer science papers," In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York, USA, July 9-15, 2016.

[23] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M.S. Khorsheed, and A. Al-Rajeh, "Automatic Arabic text classification," In: *Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data*, Lyon, France, March 12-14, pp. 77-83, 2008.

[24] S.L. Ting, W.H. Ip and A.H.C. Tsang, "Is naive bayes a good classifier for document classification?" *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 37-46, 2011.

[25] Y. HaCohen-Kerner, A. Rosenfeld, M. Tzidkani and D.N. Cohen, "Classifying papers from different computer science conferences," In: *Motoda H., Wu Z., Cao L., Zaiane O., Yao M., Wang W. (eds) Advanced Data Mining and Applications, Lecture Notes in Computer Science (Springer)*, vol. 8346, pp. 529–541, 2013. doi: 10.1007/978-3-642-53914-5_45

Hemrajsingh Gheeseewan is currently a final year student studying the BSc (Hons) Computer Science programme at the Department of Information and Communication Technologies, Faculty of Information, Communication and Digital Technologies, University of Mauritius. As part of the compulsory Industrial Training module, he had the opportunity to work as an intern at Ceridian Mauritius Ltd for period of 10 weeks from May 2019 – August 2019, where he refined his skills as a programmer as well as his communication & social skills. His research interests include Mobile Application Development, Web Technologies, Database Management Systems, Machine Learning, Machine Translation, Deep Learning, and Artificial Intelligence. He is delighted to see his work being published in the International Journal of Computing and Digital Systems, which is published by the University of Bahrain and is indexed by Scopus.

Sameerchand Pudaruth is a Senior Lecturer and Head of the ICT Department at the University of Mauritius. He is a member of IEEE, founding member of the IEEE Mauritius Subsection and the current Vice-Chair of the IEEE Mauritius Section. He is also a member of the Association for Computing Machinery (ACM). His research interests are Artificial Intelligence, Machine Learning, Data Science, Machine Translation, Computer Vision, Robotics, Mobile Applications, Web Technologies, Multimedia, Blockchain and Information Technology Law. He has written more than 50+ papers for national & international journals and conferences. He has been in the organising committee of many successful international conferences such as Africon 2013, IST Africa 2014, Africon 2015, ICCCS 2015, BigData 2015, DIPECC 2015, Africhi 2016, Emergitech 2016, Nextcomp 2017, ISCMI 2017, Mauritian Academic Conference 2018, icABCD 2018, icABCD2019, icABCD2020, Mauricon ICONIC 2018 and Mauricon ICONIC 2020. He has also written a book entitled, 'Python in One Week'. He is a reviewer for IEEE Access and Environment, Systems and Decisions (Springer) journals, amongst many others.